

Developing an AI which is capable of passing the Turing Test

O.Smart (SMA14503761) / E.Musk (MUS12434275) / F.Gonzalez (GON13457722)

J.Huang (HUA13490229) / A.Pearce (PEA12451384) / G.Hutchison (HUT14470469) / J.W.Hill (HIL12379231)



Contents

	2
Introduction	3
Aim	3
Objectives	3
Related work	4
Development	6
Human "Persona"	7
.	7
NLP algorithms	7
.	7
Markov chain	7
Typing Delay Simulation	8
Naive Bayes Text Classification	9
Development Log	9
Software Testing	14
Methodology	14
Results and discussion	16
Our Turing Test	18
Tools	23
GitHub	23
Facebook	24
Microsoft OneDrive	26
Team Gantt	27
Outlook Online	28
Sublime Text	29
Google Forms	30
Reflection on our Group Project	30
Mark Allocation	34
References	34

Introduction

Aim

The purpose of this project is to explore the possibility of developing an AI (Artificial Intelligence) that is capable of passing a modified version of the Turing test. The Turing test (Turing, 1950, 433) was described as “Three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman”, but Turing proposed that a machine takes the part of A in the game. Would the interrogator decide wrongly as often who is the computer and who is the human as if the game is played between a man and a woman? (Gilbert and Forney, 2014) The test is ‘said’ to have passed when a computer is judge as human in more than 30% of the cases after five minutes of questioning. The judge knows that one of the agents is a human and one is a computer, this test is known as the “Standard Turing Test”. If more than 30% of the judges mistakenly identify the computer agent as human, the machine is classified as intelligent.

In order to pass the Turing test, we have decided to develop a Twitter bot. Our decision is based on the recent success and use of many socialbots. (Freitas et al, 2014) on its research estimate that more than 20 million Twitter accounts were fake in 2013, those accounts were used by socialbots, AIs designed to mimic real users. Socialbots have been used successfully by many politicians, governments or marketing companies to manipulate public opinion.

Our version of the Turing test consists of choosing six participants to chat with the Twitter bot; they will need to choose between fifteen different hypothetical illnesses. All the participants in the sample are naïve of the possibility of conversing with an AI.

Objectives

- Decide on a topic and theme for the bot. It should be something interesting that people can relate to in order to promote further interaction.
- Acquire a dataset for the bot to base tweets on. For the bot to tweet about the chosen topic, it needs the dataset to go with it.
 - Choose a dataset that is relevant to the theme of the bot.
- Code the Twitter bot:
 - Implement the chosen dataset so that the bot tweets about the subject chosen.
 - Ensure it can reply and interact with other users, as this is vital to passing the Turing test.

- Adapt the language used to the chosen theme and dataset. This can be extremely effective at making it appear human-like.
 - Consider factors such as how often the bot tweets and how fast it replies to users interacting with it. If it tweets too often or replies too quickly, it will make it less likely to pass the Turing test.
- Set up a Twitter account for the bot. The bot needs its own account in which to tweet and interact on.
 - Create the account based on the chosen theme
 - Display as much personal information to persuade the participants in believing the AI a real human. Too little information on the profile can make it seem more like a bot.
- Test and evaluate the bot. Adjustments can be made to the bot based on the results of the tests.
- Choose and use a backup social network just in case Twitter is unavailable e.g. Snapchat.
- Create a methodology to evaluate our AI, researching related work in the field.
 - The methodology will consider the capabilities of the bot.
- Evaluate the project using the methodology.
 - Analyse the results.
- Analyse the tools used during the development process, justifying the reasons we chose the tools.
- Evaluate the development process of the bot, the source code used and the changes made to the code.

Related work

Important innovations have been achieved over the years in AI since Alan Turing first published his paper; Computing Machinery and Intelligence in 1950. However, it was not until 2011 (Gilbert and Forney, 2014) that an AI was able to pass a modified version of the test, which simply involved a judge asking questions to a computer and deciding whether or not the computer was controlled by a human. The judges were naïve to the possibility that they could be interacting with a non-human agent during the test.

Developed by a British scientist, the AI used was a chat bot called Cleverbot. Cleverbot used a database of more than 150 million previous online conversations with humans, much like our project. Cleverbot is able to search through its database to find the key words that match the user input provided, responding with a human-like answer.

Gilbert and Forney (2014) conducted a research study in an online 3D virtual environment, known as Second Life. Second Life offers users the possibility of earning the equivalent of \$2 in virtual currency, if they completed a tour in a virtual store, as well as a survey about the experience. The avatar in charge of the tour was controlled by a researcher in half of the cases and by an AI in the other. The researcher was trained to cover a fixed sequence of topics and was given scripted set topics, as well as flexibility in responding to participants' questions. While conducting the tour, the user is invited to interact through a local chat with the AI or the researcher. The AI (Gilbert and Forney, 2014) converses using a more robust variant of AIML (Artificial Intelligent Markup Language). This bot uses series characteristics to enhance its human-like experience, such as adding pauses in chat responses, making occasional spelling errors and shifting to lower case syntax. In addition, the bot restricts the domain of the answers to those expected in the tour.

Our Twitter bot 'Dr. Brian Mitchell', takes these characteristics and uses them to its advantage. These characteristics are shown by having delayed responses and backed-up with a convincing bio: *"Professional Medical Consultant. PhD in Medicine. Medical Consulting on Twitter and Snapchat."*

Once the tour was completed, the participants were given a link to a questionnaire, consisting of a series of 5-point Likert scale that assessed its view regarding the store presentation of the items, the helpfulness of the guide and quality of the tour. At the end of the questionnaire, the two final questions addressed the central issues of the study. The participants were asked to answer if they felt that the avatar was controlled by a male, female or an AI and were provided with a text box to explain the factors that made them feel that the avatar was controlled by a human or AI. 78% of participants who did the tour with the AI incorrectly judged the bot to be a human. On the other hand, 10% of the participants who did the tour with a human controlled avatar, failed to identify the human.

According to Gilbert and Forney, judgment in the test was impacted by the following psychological and contextual factors.

- All the participants were naïve, regarding the fact that in 50% of tours the chat was control by an AI. (Some researchers have noted that the judges' knowledge of the possibility that they are interacting with a machine greatly increases the difficulty of the test).
- The duration of the tour was limited.
- The tour followed a specific structure, where the ranges of the conversations were more defined.

In our version of the Turing test, we are taking advantage of these psychological and contextual factors as shown in Gilbert and Forney's experiment. The participants will be naïve on the fact that there is an AI between them in the Twitter conversation in 50% of the sample. We have also decided to limit the

duration of the conversation during the tests, and we have chosen the topic of the conversation to be orientated around common illnesses.

The 100th anniversary of Turing’s birth, Reading University (Warwick and Shah, 2014) staged a Turing test to conform to Turing’s original paper (Turing, 1950). Five machines, thirty judges and twenty-five hidden humans took part on the tests. Warwick and Shah (2014) discuss on their paper, not how good or bad machines are in test, but the performance of the judges interacting with the hidden humans. They analysed 13 transcripts of the tests, all transcripts were human-human interaction. After the test, 12 of the 13 judges thought it was definitely a machine and not a human.

Each conversation lasted for a total of 5 minutes; once a sentence had been transmitted, it could not be altered or retracted in any way. At the end of the five minutes, judges were asked whether the entity was human or machine. The test was “unrestricted conversation”, which meant the judge could ask anything or introduce any topic. An unsure classification was allowed if the judge definitely could not say if it was a human or machine. Humans were asked to be themselves, however they were asked not to make it easy for the machine and to not reveal personal details, that could reveal the judges identity. Some of the hidden humans and judges were not native English speakers.

This is a piece of the transcript number 2:

[11:13:11] Judge: Hello
[11:13:19] Entity: hello
[11:13:28] Judge: HOW ARE YOU?
[11:14:06] Entity: VERY WELL THANK YOU. IS THE CAPITALS A STRATEGY?
[11:14:32] Judge: Yes. Or it was.
[11:14:58] Entity: fair enough. surely they’d be more intelligent than that
[11:15:29] Judge: They? Who are you refering to?
[11:15:47] Entity: your momma. sorry couldn’t resist
[11:16:09] Judge: Wow.
[11:16:59] Entity: world of warcraft? i like RPGs but i prefer tabletop. have you ever played?

In this transcript, it is clear how the hidden entity appears to be doing most of the talking in the conversation and using a tendency for humour. It is interesting how the judge in this test thought that the hidden entity was a machine.

Implementing the factors discussed on these researches in our methodology will increase the possibilities of success for our project.

Development

The bot is coded using Ruby programming language. Ruby is a general purpose, object oriented programming language, and was chosen because it is one of the

easiest languages to write in and learn for newcomers to programming. The language is not complicated to read either. The source code of the bot has code comments to explain what the program is doing for people who want to contribute to the bot's development. The bot was made using open source code, which our project '*Masqueraide*' improves upon. This main open source project that we extend is called '*twitter_ebooks*' by Jaiden Mispy (@mispy). Our aim was to use '*twitter_ebooks*' text generation capabilities and improve it by making it smarter. The way in which we do this is by using a text classification algorithm called a Naive Bayes classifier that will be discussed further in the NLP algorithms section.

Human "Persona"

In order to make the bot seem as though it is a human, a convincing persona had to be created to add character to the bot. The Twitter profile was perfect for doing this as when a user signs up they are required to input a bit about themselves in a "Bio" section and they are encouraged to add a picture. According to his Twitter profile, Dr. Brian Mitchell is from Boston, Lincoln and has a short bio describing his job and what he does. The bots profile also features a stock image of a professional looking man. These features would allow someone who interacts with the bot in the showcase to identify the profile to be associated with a human, therefore increasing the chance that a user interacting with the bot will be fooled into believing it is a human.

NLP algorithms

Our bot uses 3 specific NLP (Natural Language Processing) algorithms to identify, generate and process text. This is important so that our bot stands a better chance of passing the Turing test. The algorithms are as follows:

Markov chain

Our bot generates realistic response using an algorithm called a 'Markov chain'. A Markov chain is a random process that happens on a state space such as, it transfers from one state to another state or keeps the current state based on the probability distribution. In the process, it does nothing with previous events that it processed; the next state only relies on the current state (Woods, 2015). Our bot works by generating sentences based on a given dataset or text model. How real the text looks when it generates the sentence depends on how good

the dataset is. In order to make sure our dataset is accurate, our supervisor recommended a medical dataset originating from the healthcare sector. We included tweets from real doctors on Twitter to construct our text corpus model. This model is called a ‘pseudo Markov model’. A ‘Markov model’ is a statistical structure of a piece of text using a simple mathematical model. It can be used to generate a stream of text. In a real example, the previous word is always considered to help a new word that is being generated to find out which new word will be selected. Each node in a Markov model represents a word and chooses the next word based on the weight of each outgoing edge of model (Woods, 2015).

For the model training, the algorithm is as follows: Assume there will be N main dictionaries. Each key in the dictionary will be allocated to a sub-dictionary. Keys in the sub-dictionary will be given the number of times those words have appeared. Then, scan through the text N times for each word. When found where the current iteration is, put those words together and make two specific keys, using those keys to increase the frequency accordingly.

Next, choose a text either from any of the keys in dictionaries or offer the key to the dictionary; one of those words offered a reference to predict the next word. Once the word is selected, do the same thing as before, until there are several or more words that have been generated. Finally, a random response is produced.

Typing Delay Simulation

For our bot to appear human-like, the bot must send a message to the participant, not too fast or it will be caught out. Initially, our bot delays its response time randomly between 1 to 60 seconds. However, this delay is not based on anything other than randomness and can lead to a chance that the bot can generate a large amount of text and delay for 1 second. This is a red flag in trying to convince people that they are talking to a human instead of a bot.

Instead, our bot uses an algorithm based on calculating the distance between the keys on the keyboard for a word in a sentence. For this to work the keyboard has to be mapped on a Cartesian grid. For example, a sentence such as ‘*The quick brown fox jumps over the lazy dog*’ typed with a speed of 23 words per minute (23/wpm) with no mistakes, makes the bot delay for approximately 6.83 seconds. If our bot makes a spelling mistake, the delay will increase by 1 second per mistake. The words per minute of which the bot ‘types’ the sentence can also be changed depending on the situation. For example our bot ‘Dr Brian Mitchell’ can increase its words per minute to 50/wpm if it is in the unlikely situation of being in a hurry. Our typing delay simulation is a better alternative to just randomising the delay time because our bot’s response time is less predictable to the user talking to it.

Naive Bayes Text Classification

For our bot to understand what it's being said to it, we use an algorithm called Naive Bayes. Naive Bayes is a simple classification algorithm which its applications are used in statistics, finance and machine learning. The best way Naive Bayes is applied is when it comes to spam detection. Naïve Bayes is not only powerful but it is rather simple. (McCallum and Nigam, 1998) states its simplicity "in that it assumes that all attributes of the examples are independent of each other given the context of the class." (Poole and Mackworth, 2010) Explains the idea behind Bayesian classifiers like Naive Bayes. Given an agent which is our classifier and a class which can be "spam" or "mail" "If an agent knows the class, it can predict the values of the other features. If it does not know the class, Bayes' rule can be used to predict the class given (some of) the feature values."

Our bot uses Naive Bayes for detecting a sentiment in a given text. For example: given the two sentences: "I feel very sick with the flu Doctor." and "What do you recommend?", Our bot associates the first one with being 'Negative' and the second one as a 'Question'. These are all possible by the bot using a trained dataset of example Doctor responses from the NHS and relevant Twitter tweets. For the development of this feature, we use an existing classifier library called 'classifier-reborn' that implements a Naïve Bayes classifier.

Lastly, our bot detects spam or malformed input and produces an unsure response. For example, the following random input "kforfjprmfioamdhcizycozx" would not be understood by the bot, which will therefore reply by saying: "Please clarify." or "I don't think you're making sense."

Development Log

This is a development log of the software artefact 'Masqueraide'. The development was mostly done using Git + GitHub and unit tested using Travis CI which describes itself as a 'Continuous Integration' service.

commit 6fe569c762ebd90c6ebaba9a86969cf9504d05ea Author: Wesley Hill wesley@hakobaito.co.uk

Date: Mon Feb 15 21:33:47 2016 +0000

initial commit

18 files changed, 647 insertions(+)

commit 0df5ac038351e33937c012e63a7a2b5aecb5afb7 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Mon Feb 22 22:57:23 2016 +0000

ignore json & csv, initial sc api work

2 files changed, 413 insertions(+), 28 deletions(-)

commit 92cfbfc523b790b8e3386b89cda70e3800ab7f88 (tag: v0.1.0) Author: Wesley Hill wesley@hakobaito.co.uk

Date: Thu Feb 25 17:06:21 2016 +0000

more snapchat chat stuff, focus on lib not executable.

2 files changed, 22 insertions(+), 17 deletions(-)

commit fb375b0c47c40e67629a1d6bd1a8a6d90fe1b527 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Thu Feb 25 20:55:26 2016 +0000

cleanup code

13 files changed, 753 insertions(+), 784 deletions(-)

commit 6c75000003b087e4816c67684e4da4a72a2fa4fe Author: Wesley Hill wesley@hakobaito.co.uk

Date: Fri Feb 26 01:08:47 2016 +0000

added minimal tests, commented example.

3 files changed, 51 insertions(+), 27 deletions(-)

commit c2f254fdf66e95137e5473984f6f9026ae6200af Author: Wesley Hill wesley@hakobaito.co.uk

Date: Fri Feb 26 02:27:23 2016 +0000

fixed room bug, renamed method

1 file changed, 2 insertions(+), 2 deletions(-)

commit 1754ec58e7ce76ab2d92a34fd5a059807f6bf7a4 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Fri Feb 26 03:12:17 2016 +0000

fixed assignment bug

1 file changed, 3 insertions(+), 3 deletions(-)

commit 63e8095514331f381fa75d02d1edc7016ae85f59 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Tue Mar 1 13:42:41 2016 +0000

started commandline implementation, small cleaning up.

8 files changed, 143 insertions(+), 36 deletions(-)

commit 133d6c1f73c8fd0f71e5cc44e6ad75278a1915ef Author: Wesley Hill wesley@hakobaito.co.uk

Date: Wed Mar 2 10:13:40 2016 +0000

basic commandline & loading models done

7 files changed, 89 insertions(+), 64 deletions(-)

commit 2218e0085f04c4e33c876e5f7055e427b7056019 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Fri Mar 4 10:51:16 2016 +0000

initial web server work, fixed sc bug

5 files changed, 21 insertions(+), 3 deletions(-)

commit 6a625cd514df958a6dc2c673ec441c7fbc81bbbc Author: Wesley Hill wesley@hakobaito.co.uk

Date: Mon Mar 7 07:12:00 2016 +0000

initial masqueraide web dashboard work

15 files changed, 560 insertions(+), 11 deletions(-)

commit 101c081412a65a1116f72f89a2713a22cc65afe7 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sun Mar 13 03:02:54 2016 +0000

remove conversation info code

1 file changed, 1 insertion(+), 81 deletions(-)

commit 609f32d64456f7133262d36a24c63474b9c68319 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sun Mar 13 03:27:43 2016 +0000

configuring travis for testing

3 files changed, 10 insertions(+)

commit bb8a436ef44be856f98554d165ffa444e56f15a2 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sun Mar 13 03:56:13 2016 +0000

using a script to test instead.

2 files changed, 5 insertions(+), 3 deletions(-)

commit 9cf144dd4a7d4c02ade9ec6fcb6205d6a8f1422d Author: Wesley Hill wesley@hakobaito.co.uk

Date: Wed Apr 6 03:56:52 2016 +0100

added nlp typing delay + rearranged engine code.

10 files changed, 225 insertions(+), 43 deletions(-)

commit d6c85c32dc03c22fb54e741e8f240ca90701b953 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Wed Apr 6 04:24:53 2016 +0100
 fix aspell dependency on travis
 2 files changed, 33 insertions(+), 1 deletion(-)
 commit 9d226ca016f52d022e360dd9043b888696ae2aa5 Author: Wesley Hill wes-
 ley@hakobaito.co.uk
 Date: Wed Apr 6 04:35:10 2016 +0100
 no sudo and change structure
 1 file changed, 7 insertions(+), 3 deletions(-)
 commit d3d055e54fcbfb67e6e32e41804ec35d327cab22 Author: Wesley Hill wes-
 ley@hakobaito.co.uk
 Date: Wed Apr 6 04:42:12 2016 +0100
 lint travis.
 1 file changed, 2 insertions(+), 2 deletions(-)
 commit 6f5337e982e5f84fe8900e412d984e57b5cf6f5b Author: Wesley Hill wes-
 ley@hakobaito.co.uk
 Date: Thu Apr 7 17:07:22 2016 +0100
 added misspellings and snapchat pings
 4 files changed, 43 insertions(+), 1 deletion(-)
 commit bd52eef6589e508c83abc78486b9a3f05284a70f Author: Wesley Hill wes-
 ley@hakobaito.co.uk
 Date: Fri Apr 8 05:53:01 2016 +0100
 started work on profile pic uploading...
 1 file changed, 62 insertions(+), 18 deletions(-)
 commit 3ea979a96205e32570aa3031d8be825ebfc4ba16 Author: Wesley Hill wes-
 ley@hakobaito.co.uk
 Date: Fri Apr 8 05:59:34 2016 +0100
 unused
 2 files changed, 2 insertions(+), 2 deletions(-)
 commit 16b203b61c08a24a3a96a3ad85b4c6378ef64393 Author: Wesley Hill wes-
 ley@hakobaito.co.uk
 Date: Fri Apr 8 21:42:32 2016 +0100
 removed more unused variables, cleaned code
 2 files changed, 32 insertions(+), 9 deletions(-)

commit 55c87bf0bb522f0f4643bcf9d9b01776c7ac87ba Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sat Apr 9 19:47:05 2016 +0100

add more tests to ai rooms

8 files changed, 69 insertions(+), 11 deletions(-)

commit 9bfb90c146599742c8242150a311325d4af6dae1 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sat Apr 9 22:17:10 2016 +0100

minor fixes

4 files changed, 17 insertions(+), 12 deletions(-)

commit 4cfef233ff41f039760f57c37f27e594c1b927a4 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sun Apr 10 08:09:51 2016 +0100

commented code. rubocop linting, and 0.1.1

11 files changed, 82 insertions(+), 53 deletions(-)

commit 2b2a33f6f14a94cb98198f4bf95481a8cac6cfc2 (tag: v0.1.1, web, snapchat, fixes) Author: Wesley Hill wesley@hakobaito.co.uk

Date: Sun Apr 10 08:11:53 2016 +0100

added server comments

1 file changed, 5 insertions(+), 2 deletions(-)

commit ed322c193da517c5ec1f35afdbbba334eb71623 Author: Wesley Hill wesley@hakobaito.co.uk

Date: Wed Apr 13 09:17:19 2016 +0100

added dr brian mitchell bot + classifier, (see desc)

+ added example bot

+ formatted & commented code

+ added bayesian classifier and training set.

21 files changed, 1741 insertions(+), 597 deletions(-)

commit 398067ca8ced465aa6878ec165edab8cf0f560c6 (HEAD -> master, origin/master, nlp-work) Author: Wesley Hill wesley@hakobaito.co.uk

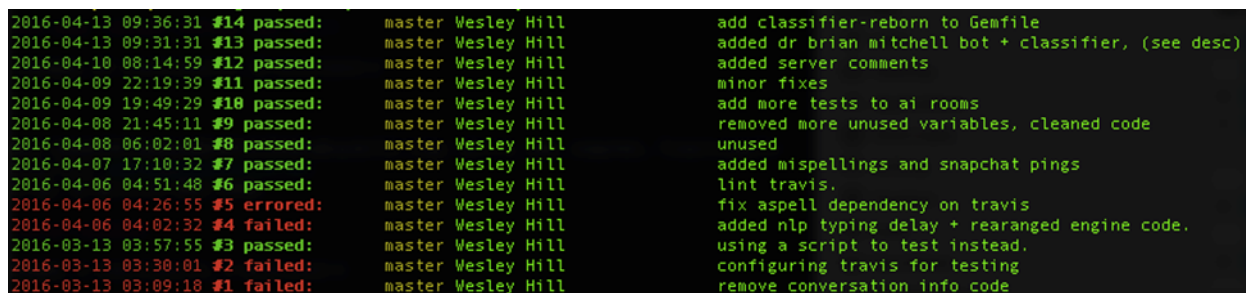
Date: Wed Apr 13 09:33:47 2016 +0100

add classifier-reborn to Gemfile

1 file changed, 1 insertion(+)

Software Testing

The bot's software was tested using a service called Travis CI. This service tests code, which is hosted on GitHub when a git commit is made. The software automatically tests the source code to see if the software works, as it should. The following image below shows a log of the tests being made. A pass means that all the software tests in the source code work, a fail means that one or more tests are broken.



```
2016-04-13 09:36:31 #14 passed: master Wesley Hill add classifier-reborn to Gemfile
2016-04-13 09:31:31 #13 passed: master Wesley Hill added dr brian mitchell bot + classifier, (see desc)
2016-04-10 08:14:59 #12 passed: master Wesley Hill added server comments
2016-04-09 22:19:39 #11 passed: master Wesley Hill minor fixes
2016-04-09 19:49:29 #10 passed: master Wesley Hill add more tests to ai rooms
2016-04-08 21:45:11 #9 passed: master Wesley Hill removed more unused variables, cleaned code
2016-04-08 06:02:01 #8 passed: master Wesley Hill unused
2016-04-07 17:10:32 #7 passed: master Wesley Hill added misspellings and snapchat pings
2016-04-06 04:51:48 #6 passed: master Wesley Hill lint travis.
2016-04-06 04:26:55 #5 errored: master Wesley Hill fix aspell dependency on travis
2016-04-06 04:02:32 #4 failed: master Wesley Hill added nlp typing delay + rearranged engine code.
2016-03-13 03:57:55 #3 passed: master Wesley Hill using a script to test instead.
2016-03-13 03:30:01 #2 failed: master Wesley Hill configuring travis for testing
2016-03-13 03:09:18 #1 failed: master Wesley Hill remove conversation info code
```

This is useful when teams want to introduce a feature without breaking the entire software, and the author is able to track down quickly what build the test failed on.

Methodology

After completing our analysis, we have decided to implement our version of the Turing test, where we can take advantage of the characteristics of our bot. Together with the factors discussed on the research papers analysed, this provided a major advantage in the success of the machine on other versions of the test. This test will allow us to measure the success of the development of our AI, meeting with the aim of the project, pass the Turing test.

The design of the AI wants to portray itself as a human as possible. It will adapt itself to common human habits by analysing normal human language, thinking and responding to the questions given to Dr Brian Mitchell. Many human habits have been incorporated into our AI. For example, realising when a human is spamming and not making any logical sentences. Our AI will be able to determine the difference between these and respond appropriately. It utilises everyday habits of humans e.g. delayed responses, 'typos' on the keyboard in order to make it seem more real to the user. Twitter is one of the largest online social media platforms with thousands of users. By having our AI involved in

this social media platform, humans will already assume the AI is a human. The profile portrays a human-like feel by having a profile picture and a convincing bio 'Professional Medical Consultant. PhD in Medicine. Pioneer of Medical Consulting on Twitter and Snapchat'.

Due to some problems in the development and the time constraint of the project, we changed our original plan for the criteria. Originally, we planned to test our AI on the main part of Twitter (e.g. the AI was supposed to use Tweets to talk to the participants – but as it is quite sensitive information, we were unable to do this). Another measure of success would have been measuring the quantity of followers that the AI could gain. The original criteria have been modified for criteria that we think the AI could meet with capabilities of our AI at this point in the development process.

We designed a test where the participants will have the opportunity to send private tweets to our bot in real time. We have chosen an assortment of ten potential illnesses. The participants are told that the person answering the questions is Chinese and their knowledge of the English language is limited. This could explain any meaningless response from the bot, in case the conversation goes out of topic and the AI gives inappropriate responses.

Six participants will be selected for the test, all participants will naïve of the possibility that there is an AI among the people in the conversation, according to (Gilbert and Forney, 2014) this has been proven an important factor in AIs successfully pass the Turing test. We have limited the time that the participants are able to interact with the AI, increasing the time between tweets and limiting the total period of the test to 3 minutes. This increases the chances of success due the limitation of our AI. In the original version of the Turing test, the time was 5 minutes.

After the 3 minutes, the participants need to fill out a questionnaire. We have designed the questionnaire with 5 questions. The first question asks for, the participants chosen illness and then proceeds with a series of multi selection questions. In the beginning the questionnaire asks for general questions about how helpful was Dr. Brian Mitchell, the delay on the responses and a question about the quality of the English of Dr. Brian Mitchell. All these questions merely served as precursors to the final question that addresses the central aim of the project. In the final question, the participants will be asked to answer if they think Dr Brian Mitchell is or is not human. The question gives them the possibility to fill up a text box explaining, why they think that is or is not human?

Once we have completed our 'Turing test' we will analyse our results, and represent it in a pie chart alongside percentages to see how well or if we passed our version of the Turing test, more that 50% answer that Dr. Brian Mitchell was human. By showing statistics, we can display how well our Turing test performed and if it passed or not.

Results and discussion

Introduction to people testing Turing test

At the beginning of the Turing test, our group selected 10 common illnesses and conditions, which the twitter bot can respond to in a human-like fashion rather than a rare illness. By doing this, we are giving ourselves a higher percentage of passing our guidelines for the Turing test.

We found the most common and widespread illnesses by looking at the demographics of Twitter. We were then able to evaluate what is the most common illness experienced by the most statistically relevant age group, 18-49.

Twitter Demographics	
<i>Among internet users, the % who use Twitter</i>	
	Internet users
<i>Total</i>	23%
Men	25
Women	21
White, Non-Hispanic	20
Black, Non-Hispanic (n=85)	28
Hispanic	28
18-29	32
30-49	29
50-64	13
65+	6
High school grad or less	19
Some college	23
College+	27
Less than \$30,000/yr	21
\$30,000-\$49,999	19
\$50,000-\$74,999	25
\$75,000+	26
Urban	30
Suburban	21
Rural	15
Source: Pew Research Center, March 17-April 12, 2015.	
PEW RESEARCH CENTER	

The most common of Twitter users are aged 18-49 according to PEW Research Centre, therefore by finding the most common illnesses in this age group and

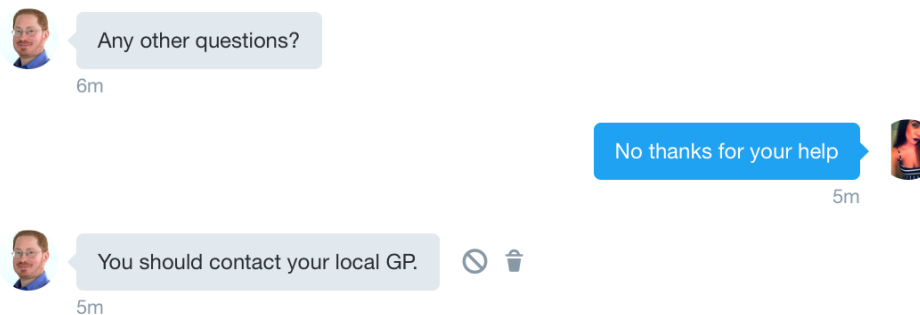
finding a dataset with the language and symptoms used with the illnesses, we could get a more human-like response for the users. From this information, we decided to find illnesses most common in 18-29 year olds, as they are the highest percentage.

Hockley Farm Medical Practice states that, there are several common illnesses that they see ranging from the common cold to cystitis (Hockley farm medical practice, no date).

We chose the common illnesses:

- Common cold
- Cough
- Influenza
- Hayfever
- Cystitis
- Small sprain
- Sunburn
- Earache
- Stomach Pain (Diarrhoea)
- Infected Wound

By choosing something that is common amongst 18-49 year olds, which used Twitter, the bot would be able to respond more accurately as we would develop it to answer questions commonly associated with these chosen illnesses.



Once the bot has finished the conversation because it cannot prescribe any medication over the internet, it would respond with phrases such as 'you should contact your local GP' so that the service can be continued and the person can get better from their hypothetical illness.

Our Turing Test

We used the Google Forms for our survey and gave it to 6 people to fill out once performing the Turing test. It must be noted that the results include 7 participants due to testing purposes on the form.

In order for us to display our artefact, we will need to be able to draw people in to gather information about how well it is performed. We will be using hypothetical problems so no consent form will be needed to be signed, allowing anyone who wishes to partake to be comfortable knowing that it is completely anonymous.

All conversations will be deleted after the user has discussed with the artefact.
All participants would have the same introduction:

“Hi there, would you like to come and try our new remote health clinic. We’ve designed it to use Twitter so it’s available from anywhere! It’s completely anonymous and does not require any personal information from yourself.

We have teamed up with students from the Wuhan University in China, as they’ve agreed to try and help students from the University of Lincoln with potential illnesses.

Now, unfortunately, due to ethical problems with personal health issues, we’re unable to allow personal health problems to be discussed. Instead, we have an assortment of 10 potential illnesses for you to select at random.

All you need to do, is use our laptop and pretend like you have the illness you’ve selected randomly, and speak to our team like you would a normal doctor.

Please be aware that the response time may be slow as our server connection to China sometimes has some latency issues.

Once you feel your theoretical issue has been resolved or you feel you’re not really getting anywhere, come back to us and we have a very small 5 question survey we will need you to fill out, so we can improve on our service and give feedback to those helping us out.”

The participants performed the Turing test and filled out the survey. On the next page, you will see our questionnaire and our results from our version of the Turing test.

Dr Brian Mitchell - The Online Doctors Surgery

Evaluating how useful Dr Brian Mitchell has been from the online doctors surgery

What was the illness you chose?

Your answer

How helpful was Dr Brian Mitchell?

- ☐ Extremely Helpful
- ☐ Very Helpful
- ☐ Average
- ☐ Quite unhelpful
- ☐ Extremely Unhelpful

Was there a huge delay in his responses?

- ☐ Long delay
- ☐ Slight Delay
- ☐ No Delay

Was Dr Brian Mitchell's English okay?

- ☐ Extremely accurate
- ☐ Some mistakes
- ☐ A lot of mistakes

Do you think Dr Brian Mitchell is or isn't human and why?

Your answer

SUBMIT

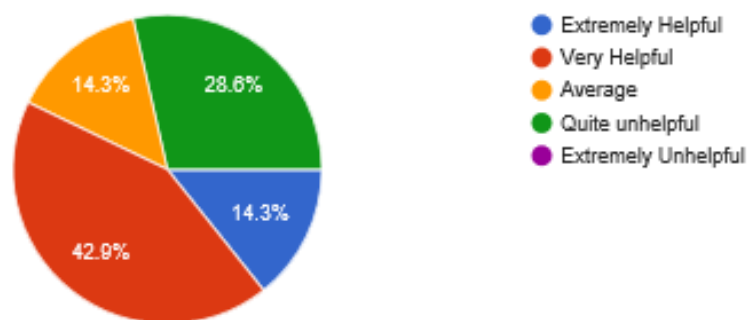
The data was then collated using a Google Form and presented using pie charts.

The form included an open question at the end to determine whether it was a human or not. This allowed us to see if we had passed our version of the Turing test. Our Turing test passed according to our criteria with 4 / 6 participants believing it was a human.

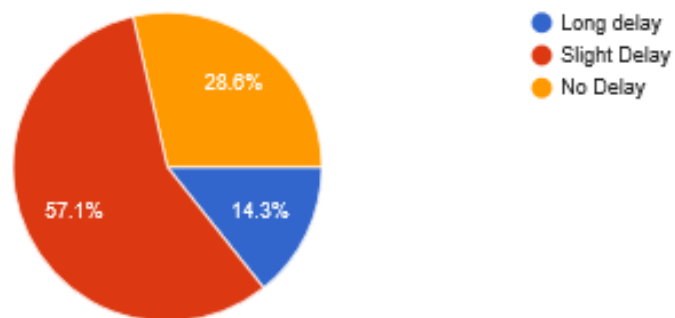
What was the illness you chose? (7 responses)

Sunburn	Create Spreadsheet
Sunburn	
Hayfever	
Earache	
Common Cold	
Cough	
Small Sprain	

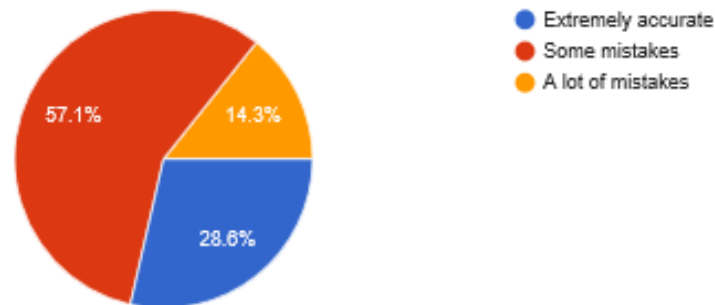
How helpful was Dr Brian Mitchell? (7 responses)



Was there a huge delay in his responses? (7 responses)



Was Dr Brian Mitchell's English okay? (7 responses)



Do you think Dr Brian Mitchell is or isn't human and why? (7 responses)

There was a lot of grammatical errors and responses which didnt make sense.

No the grammar and responses just didn't make sense sometimes and I feel a medical undergrad may have had better English

he was human, because the responses were accurate and useful

Yes and no, there was some mistakes like any ordinary human but everything was a bit delayed. I got good advice and I feel like this could be human

I think he is human. He responded to my question well, providing useful advice. The speed at which he replied was decent but not impossibly fast.

I think it is a human.

Yes this is human because it solved my problem online and now i know the way forward if this ever happens to me.

A range of different illnesses were chosen and interacted with the bot. 42.9% of the participants believed Dr Brian Mitchell was helpful with their illness and 57.1% of people thought there was a slight delay. The slight delay makes the human-like persona more lifelike, because every single human makes slight mistakes when typing (typos). Dr Brian Mitchell also made little mistakes when concerning his English (57.1%). The responses that were sent from Dr Brian Mitchell illustrate a good understanding of English and even 28.6% of the participants thought that it was extremely accurate.

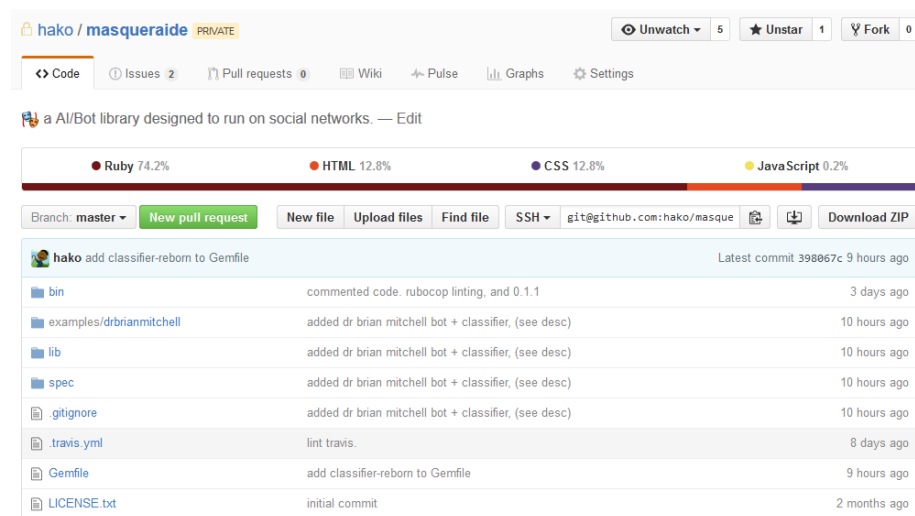
In conclusion, our version of the Turing test met our objectives and criteria. In the future, we would make an AI, which had 'perfect' English language skills

and adapt to more human-like behaviours e.g. regular tweet updates about the Doctors Surgery and more specific diagnosis to a range of diseases that are more aggressive.

Tools

During the course of the project, we, as a team, have used many tools to aid in our success. These range from communication tools such as social media, to organisational tools, like online gantt charts. This section is all about why we chose said tools and how we justified their use for the benefit within the project.

GitHub



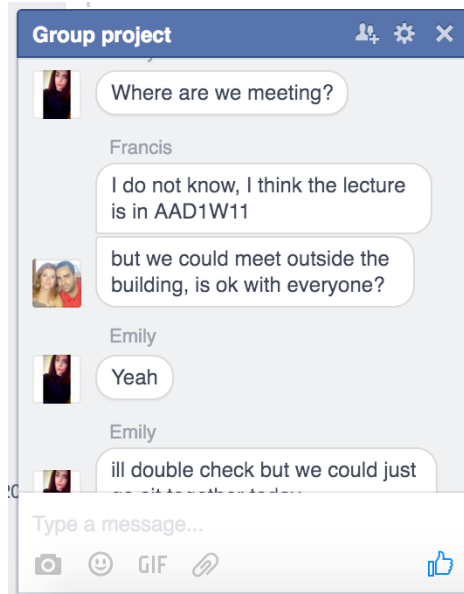
To use GitHub, our team members have to use a tool called 'Git' which is a revision control system for keeping track of source code changes. Without focusing too much on Git itself, our team makes a change to the source code. If they are happy with the change they can make what's called a 'code commit' that produces a snapshot of the code. Once the commit is made they can carry on programming and making changes, or they can push their changes onto a remote code repository like GitHub. By using GitHub the group were able to fix bugs without affecting each other's code base and test code every time a commit has been made. The tests written for the bot uses a service called 'Travis

CI', which automatically tests code every time one person in the group pushes their changes to GitHub. This helps the group make sure that every feature implemented into the bot is working as expected and eliminates room for error when running the bot.

Facebook

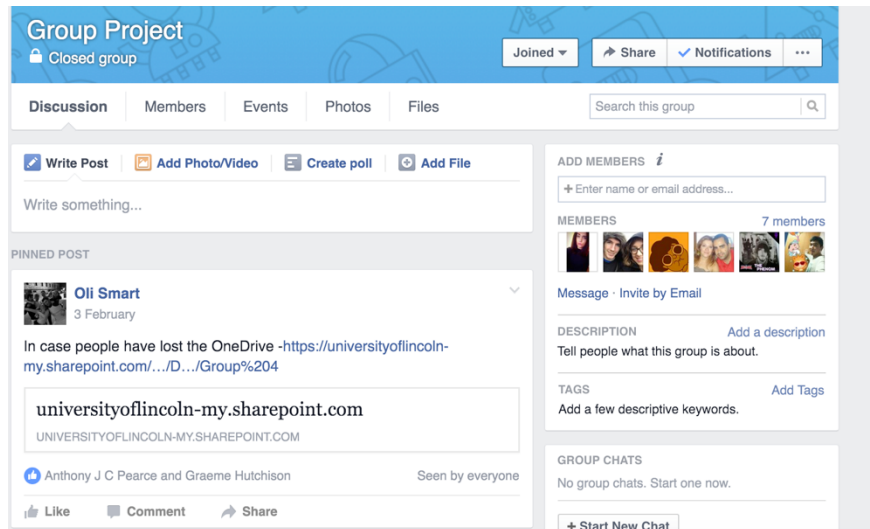
Facebook is an online social media site, which has many tools, which can aid in a group project. The tools, which we chose to use, were:

- Facebook Messenger
- Facebook Group Page (Closed Group)



Facebook Messenger (Fig. 0) was used primarily at the start. We conversed to each other through the application, and it allowed us to instantly message others about the whereabouts of important events/rooms/deadlines etc.

This significantly improved our turnaround time of pieces of work.



However, due to the size and lack of organisation available within Facebook Messenger, we decided to create a group page (Fig. 0.1). This was a private group page which enabled all users to post important information which was easily organised, editable and functional. Users could comment on posts which meant it was clearer as to what issue they were talking about, links could be pinned (pinned posts means they stay at the top of the page – regardless of new content), and items could be deleted.

Having a Facebook Group also allowed decisions to be made easier. We used the “create poll” option frequently when we hit a crossroad. This allowed us to vote for options and continue with our work without having to organise a group meeting.

Facebook also allowed for “Mobile” planning and updating as well. Through the Facebook and Messenger apps, group members could keep in touch and post updates even when they were not at home, at a computer or on the move. This allowed for almost instant replies from any if not all members in the event of an update or question from another group member. It also allowed the group to track the progress and attendance of all members. For example if someone would be unable to attend, it could be made clear via the group chat and then could be conveyed directly to Derek.

While there were and still are many different ways in which the team could communicate, Facebook seemed to be the easiest and most obvious choice as everyone in the group had already been using the social network for some time and therefore had some familiarity with the layout and function of the site and mobile applications. As all of the members frequently use Facebook, it meant that they would most likely be compulsively checking for updates throughout the day meaning it would be difficult for them to miss an update. These factors made it the first choice for communication throughout the project.

Microsoft OneDrive

We used Microsoft OneDrive as our cloud-based storage. Everyone in the group had an Office account (as this is available through the University) – so we decided it would be best to create a shared folder in which we could store all of our documents.

The main structure of the shared area is very organized (Fig 1.0). With folders for both assessment items (each of which have sub-folders containing important documents). The Main folder also includes up-to-date files for our AI (Twitter Chat bot called Masqueraide), the most *recent* assessment information and *job allocation*.

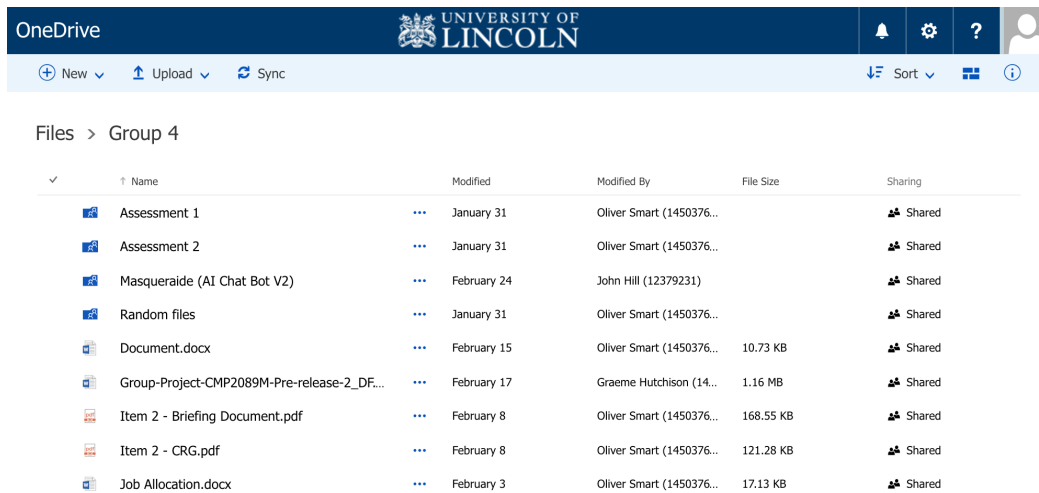


Figure 1.0 - Showing the base layer of our shared OneDrive folder

Going deeper into the folder structure (Fig 1.1), you can see that all items are categorised and placed into their respective folder. Fig 1.1 shows the minutes of meetings that we have kept for the second assessment, as well as the report. Progressing deeper into the file structure, we have folders for the sub-groups and their work.

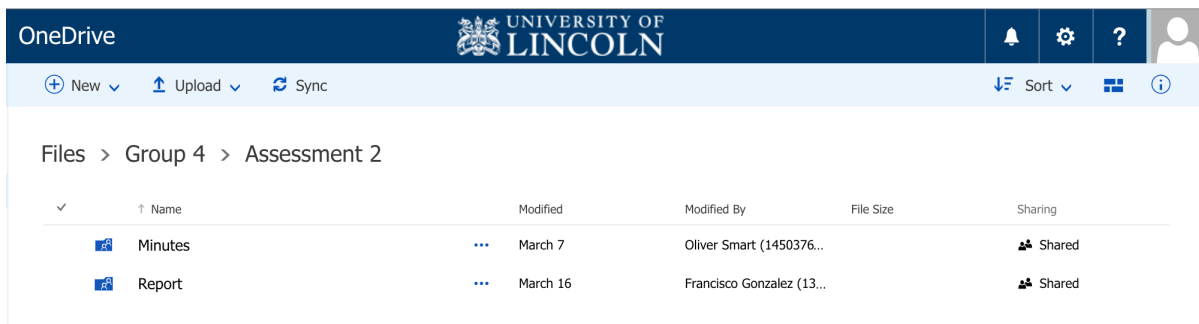


Figure 1.1 - Showing Assessment Item 2

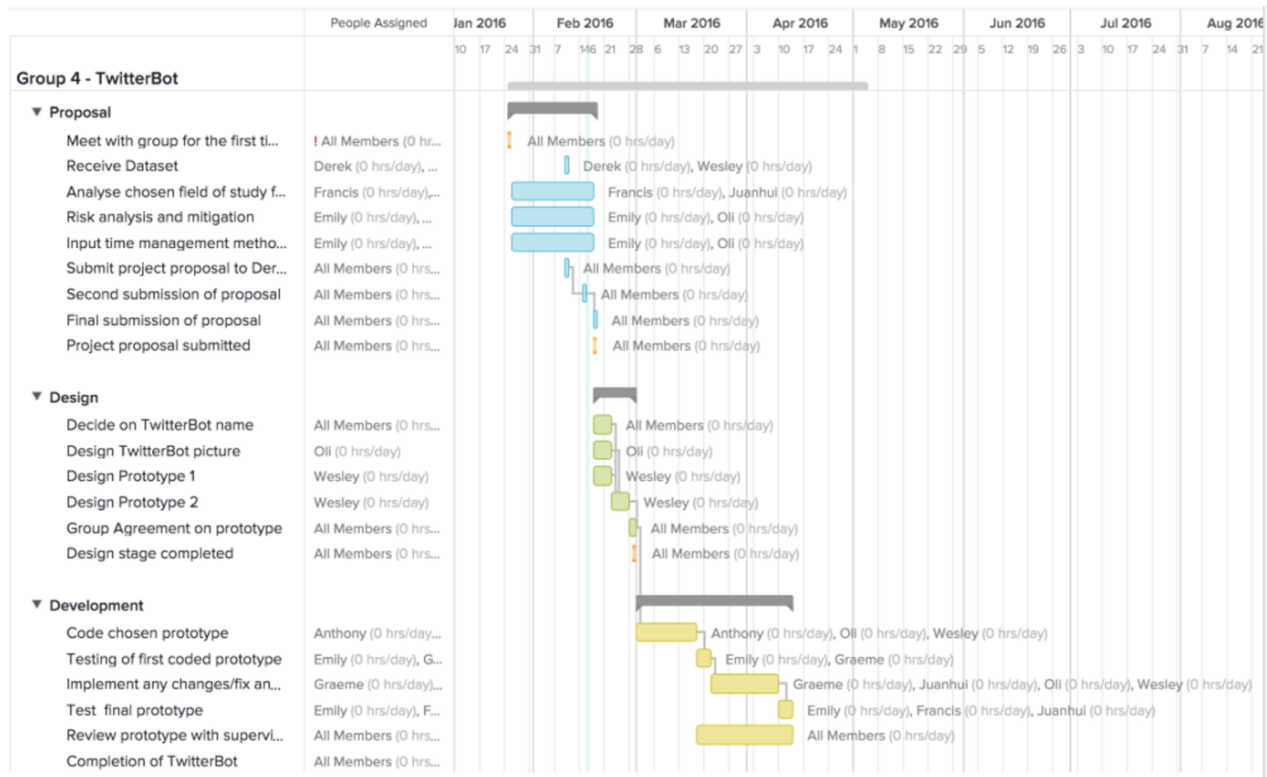
At the end of all of the work, we then compiled all documents into one and then we would take it in turns to do a full grammar/spelling/layout check of the document, ensuring it is ready for publishing and submitting.

As OneDrive is a storage tool, it's functionality is far superior to Facebook's storage mechanisms, as there is no hierarchal folder functionality. Moreover, OneDrive has an online live-edit tool, which allowed us to comment, edit and save, all in real time.

Team Gantt

An online tool we used to create our work chart (also known as a Gantt chart), was teamgantt.com. Team Gantt is a fantastic bit of online software aimed to create organisational charts for group projects. Users are able to sign up to the service (which is a free trial for 30 days), and then view/edit and even received email updates about upcoming deadlines.

Other software tools were available (such as Microsoft Project), however, we had trouble installing this on all machines. Moreover, seeing as Team Gantt is an online resource, we could access the most up-to-date chart.



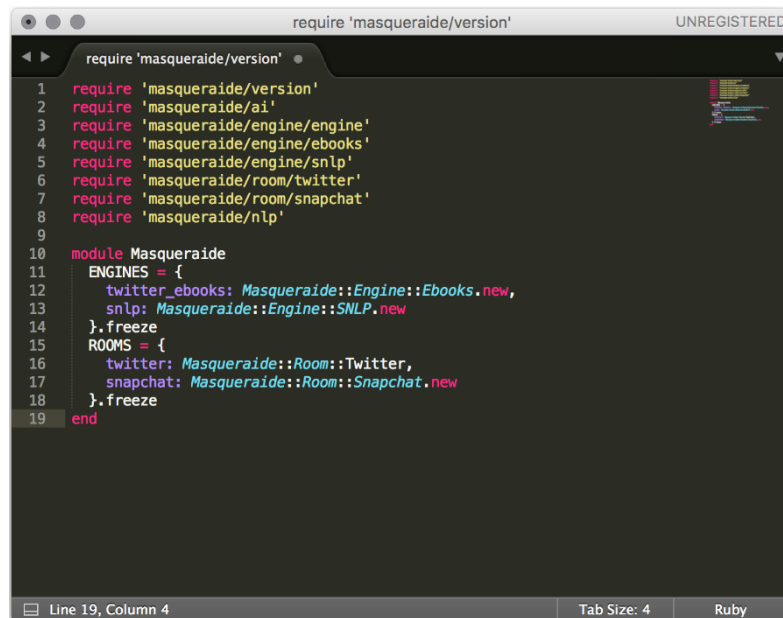
By creating the Gantt chart, all users were able to see when the deadlines for their pieces of work were, as well as the dependencies of their work. For example, looking at Fig. 2.0, it shows that Emily and Graeme were not able to start her task without the completion of the previous task; done by Anthony/Oli/Wesley.

Outlook Online

During the course of the project, we were told it's crucial to maintain a steady workflow in a group project to ensure that all members are focused, as well as to ensure that we don't fall behind on the huge demand of work. This was achieved primarily because of regular tutor meetings, as well as constant feedback on our work (which allowed us to maintain a high standard and constantly improve the quality). This was all done through emails with our tutor (Derek Foster). He informed us of the dates/times/locations of the meetings, as well as when he had commented on our work on OneDrive.

Sublime Text

Sublime text (Fig. 3.0) is a fantastic text editor that the group used to code the AI. It allows split editing (useful for when you need to work on more than one document), language plug-ins (we used Ruby so we installed the plug-in which allows specific functions/variables etc. to be highlighted). Sublime text obviously played the largest part in the production of the artefact, as it is the tool that was used.

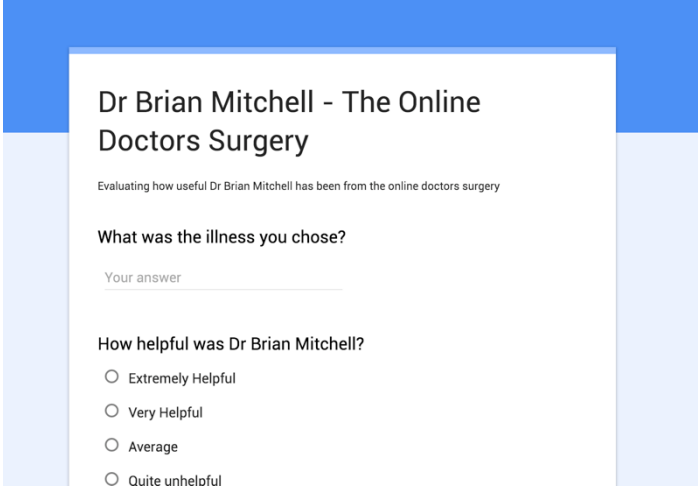


```
require 'masqueraide/version'

1 require 'masqueraide/version'
2 require 'masqueraide/ai'
3 require 'masqueraide/engine/engine'
4 require 'masqueraide/engine/ebooks'
5 require 'masqueraide/engine/snlp'
6 require 'masqueraide/room/twitter'
7 require 'masqueraide/room/snapchat'
8 require 'masqueraide/nlp'
9
10 module Masqueraide
11   ENGINES = {
12     twitter_ebooks: Masqueraide::Engine::Ebooks.new,
13     snlp: Masqueraide::Engine::SNLP.new
14   }.freeze
15   ROOMS = {
16     twitter: Masqueraide::Room::Twitter,
17     snapchat: Masqueraide::Room::Snapchat.new
18   }.freeze
19 end
```

The reason to use Sublime text instead of other text editors is its flexibility. Its cross-platform so all of us were able to view the code (however most of the time we viewed the code and the versions through GitHub). Moreover, the main coder (John) had vast amounts of previous experience using Sublime so it seemed fitting that the group adapted to his coding suite/style.

Google Forms

A screenshot of a Google Form titled "Dr Brian Mitchell - The Online Doctors Surgery". The subtitle is "Evaluating how useful Dr Brian Mitchell has been from the online doctors surgery". The first question is "What was the illness you chose?" with a text input field labeled "Your answer". The second question is "How helpful was Dr Brian Mitchell?" with four radio button options: "Extremely Helpful", "Very Helpful", "Average", and "Quite unhelpful". The form has a blue header bar and light blue sidebars.

To obtain the data needed to see if we had passed our simplified version of the Turing test, we created a simple questionnaire with five questions (Fig. 4.0) for users of the artefact to complete to give feedback. With options for scalar/non-scalar/categorised answers, Google Forms makes it effortless to allow anyone to fill it out. Moreover, it automatically creates pie charts to graphically represent the results.

Reflection on our Group Project

Our group was allocated depending on what type of person we were according to the Belbin SPI test. (Furnham et al, 1993) The Belbin Team-Role Self-Perception Inventory is a behavioural test which can also be called the SPI test. The SPI test can help determine who works well as what role in a group or a team e.g. team worker or specialist.

"An assessment of the BTRSPI has appeared which has questioned its psychometric properties and cast doubt on the ability of the inventory to provide a reliable measure of team role preferences." (Furnham et al, 1993).

By utilising this test and seeing who falls under what role preference, making it easier to allocate tasks.

This is the result of the test of each member of the group:

Pearce	Anthony James Carroll	ENRZ	PL	SH	
Musk	Emily	ENRZ	CO	CF	TW
Hutchison	Graeme	ENRZ	ME	SH	
Smart	Oliver	ENRZ	IM	SH	
Gonzalez	Francisco	ENRZ	SP	CF	IM
Hill	John	ENRZ	TW	IM	SP
Huang	Jianhui	ENRZ	SP	CF	IM

CO: Co-ordinator

TW: Team worker

CF: Completer Finisher

SP: Specialist

SH: Shaper

ME: Monitor Evaluator

PL: Plant

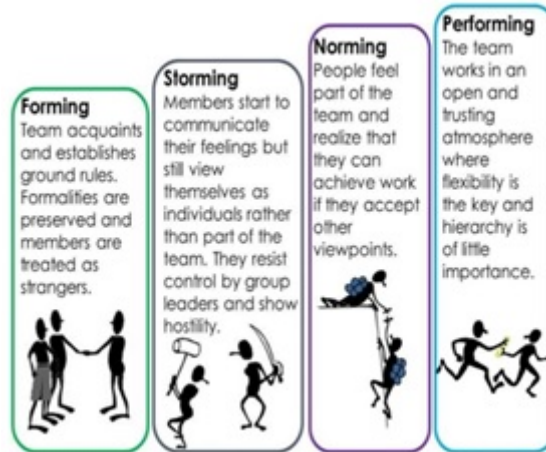
IM: Implementer

Our group had a different result per individual; our team covered all of the roles available but resource investigator. According (Furnham et al, 1993) the manager groups are cover by, negotiator which is cover by (resource investigator and team worker), manager worker (completer finisher) intellectual (monitor evaluator and plant) and team leader (shaper). Using these resources, we allocated someone as the project leader, Oliver Smart (shaper and implementer). We implemented a project leader so that we could have someone to keep us focused and organised to make sure the project was completed to a good standard.

Section					Total				
I	2	0	2	0	2	1	1	1	1
II	0	2	0	1	0	0	2	2	3
III	2	0	0	0	1	3	2	1	1
IV	0	2	4	0	1	2	0	0	1
V	2	1	3	0	0	1	1	2	0
VI	0	2	0	1	2	3	1	1	0
VII	2	2	1	0	1	2	0	1	1
Total	8	9	10	2	7	12	7	8	7

Roles	
Implementor	8
Coordinator	9
→ Shaper	10
Plant	2
Resource Investigator	7
→ Monitor Evaluator	12
Team Worker	7
Complete Finisher	8
Specialist	7

Shown above is a group members results for the Belbin SPI test. This test shows a result of a Shaper and Monitor Evaluator. As an example, we set this member of the group to the tasks of evaluating how we worked as a team and the effectiveness of the tools used in making the artefact. We figured that giving a member of the group tasks related to their Belbin result would produce a higher quality of work than someone whose speciality was different.



Dr Bruce Tuckman (Bonebright, 2010) in 1965 introduced the stages of group development theory. The model presents the well-known stages of forming, storming, and norming, performing and adjourning. The model is very popular among HRD (Human Resource development) practitioners, responding to the growing importance of groups in the workplace. This model has been proved useful, helping group members understand what was happening in the development process and predict the stages of growth in groups. According to (Betts and Healy, 2015) groups tend to perform better than individuals especially in complex

tasks, this is because individuals are bounded by intellectual and information processing capabilities. In addition to that, groups offer a diverse pool of skills and information.

On the other hand, groups tend to take more time to make decisions. This was the first problem that we faced as group, we expected nearly three weeks to make a decision on the case of study, and we could not make a decision until the first meeting with the project coordinator. This led to a tighter dead line to present the project proposal. We think that if everyone had presented more ideas and more research had been done, we could make a decision much earlier.

Once we decided the case of study, we started the forming stage of the group development theory. This stage is (Betts and Healy, 2015) task oriented, we established the rules, e.g. we decided who was the team leader, we set out a plan and put in place some project management tools. By using the project management tools we was able to plan and allocate our tasks using tools such as Gantt Charts, this gave us a way to check the advance in the project regarding the time left before the deadline. Only two members of the group had previous experience with these tools, meaning a lot of the workload was allocated to these individuals in this sector of the project. In addition to this, only one of the members of our team had previous experience on AI development, so we assigned the development side of the project to one member from the beginning (John Hill). This has been proven an error for development later, when we realized that the deadline for the submission of the project was closed and more work needed to be done in order to, make the bot answers make more sense of the questions. Some of the members could help in the development, but the development had expanded so much since the first few week there was not time to catching up to understand the code to make a significant contribution.

By when we started the second part of this project, we had had a couple of meetings together, we already met each other. This is when (Betts and Healy, 2015) the storming phase of Dr Tuckman's group development comes into place. On this stage, we communicated well with the group, but we think all of us did still presenting their ideas from an individual point of view at this stage.

We assigned the different tasks for the second part of the project just before Easter holidays to ensure maximum production.

(Betts and Healy, 2015) on its research point out that groups bring the danger of 'social loafing', "that is the notion that individuals may not expend as much effort in the group setting as they would have if they were working alone". We think that this could be avoided, better identifying each member contributions to the project, and with a better criteria to assign marks to the members of the group.

In order to 'advance' to the norming stage of group development, as a group we decided to meet up one or two times a week to update each other on what we had done, and new tasks that we needed to do as a team. By meeting up a few times a week, people would start to feel part of the team rather than individuals

and contribute more to the project.

Throughout the project, we finally achieved the performing stage of the development. This is where we are fully trusting in one another, and we contributed our thoughts openly as a team. Even with breaks throughout Easter and weekends, we could still carry on the project because we had clear objectives, good communication skills and was fully developed as a group and as a team.

By following Dr Tuckman's theory and the results from our Belbin test, we were able to delegate and see each other's strengths and weaknesses to ensure our project was completed by our submission date. Although there were several things that could have been changed, as a group we feel we have worked well as a team for our first project and produced a detailed and working artefact.

Mark Allocation

Name	Allocation
Emily Musk	A
Oliver Smart	A
Anthony Pearce	A
Francis Gonzalez	A
Jianhui Huang	A
John Hill	A
Graeme Hutchison	A

References

- Bonebright, D. (2010) 40 years of storming: a historical review of Tuckman's model of small group development. *Human Resource Development International*, 13(1) 111-120.
- Betts, S. and Healy, W. (2015) Having a ball catching on teamwork: an experiential learning approach to teaching the phases of group development. *Academy of Education Leadership Journal*, 19 (2) 1-9.
- Freitas, C. and Benevenuto, F. and Saptarshi, G. and Veloso, A. (2013) Reverse Engineering Socialbot Infiltration Strategies in Twitter.
- Furnham, A., Steele, H. and Pendleton, D., "A psychometric assessment of the Belbin Team- role Self-perception Inventory", *Journal of Occupational and Organizational Psychology*, Vol. 66, 1993, pp. 245-57.

- Gilbert, R and Forney, A. (2014) Can avatars pass the Turing test? Intelligent agent perception in a 3D virtual environment. *Int. j. Human-Computer Studies*, 15(1)30-36.
- Hockley Farm Medical practice, (no date) Self management of minor illnesses [ONLINE] Available at: http://www.hockleyfarmmedicalpractice.co.uk/website/C82053/files/minor_illnesses_HOCKLEY_LOWRES.pdf. [Accessed 11 April 2016].
- McCallum, A. and Nigam, K. (1998) *A Comparison of Event Models for Naive Bayes Text Classification*. *Journal of the AAAI/ICML-98 Workshop on Learning for Text Categorization*, Technical Report WS-98-05. 5th (May). AAAI Press. pp. 41-48. [Accessed 13 April 2016].
- Newland, J. (2015) Human versus artificial intelligence. *The online Journal of Issues in Nursing*, 10 (1097).
- Poole, D. and Mackworth, A. (2010). *Artificial Intelligence - foundations of computational agents -- 7.3.3 Bayesian Classifiers*. [online] Artint.info. Available at: http://artint.info/html/ArtInt_181.html [Accessed 13 Apr. 2016].
- Pew Research Center (2015) *Twitter Demographics*. New York: Pew Research.
- Turing, A. (1950) Computing Machinery and Intelligence. *Oxford Journals*, 59(236) 433-460.
- Warwick, K. and Shah, H. (2014) Human misidentification in Turing tests. *Journal of Experimental & Theoretical Artificial Intelligence*, 27 (June) 123-135.
- Woods, A. (2015) Generating Text Using a Markov Model. *Alexahwoods*, 5 August, 1.