# **Literature Review**

## 

## Case Studies

### 

### Notable Chatbots

The first chatbot ever implemented was ELIZA which was created by Joseph Weizenbaum in 1966. The program’s purpose was to emulate a psychotherapist; however, its conversational abilities were limited to simple pattern matching and it lacked the ability to keep the conversation going (Bradeško and Mladenić, 2012; Radziwill and Benton, 2017). Despite this, some people were still able to be fooled into believing they were speaking to a person (Dale, 2016).

Another famous chatbot is ALICE (Artificial Linguistic Internet Computer Entity) created by Richard Wallace in 1995. It is best known for being effective enough to win the Loebner Prize in three separate years. The Loebner Prize is a competition based on the Turing Test where artificial intelligence programs are judged by how human-like they are (McNeal et al., 2013a). The Turing Test refers to a test developed by Alan Turing in which a human speaks to two partners, one human and one machine, and attempts to distinguish which is which. If they are unable to guess correctly, then the machine passes the Turing Test (Abdul-Kader and Woods, 2015). ALICE’s success popularised the use of Artificial Intelligence Mark-up Language (AIML) to create chatbots (Shawar and Atwell, 2015).

A more recently created notable chatbot is IBM Watson, a chatbot created to answer general knowledge questions which won the game show Jeopardy against two human competitors in 2011. It was not connected to the internet during the game - all answers were taken from its knowledge base.

Although IBM Watson was able to answer most questions correctly, there were still a few it was unable to find the information for. For example when given the clue: “Garry Kasparov wrote the foreword for The Complete Hedgehog, about a defence in this game” which needed the answer “What is chess?”, IBM Watson was unable to give a response. In other cases, it knew the correct information but failed to word the answer correctly. For one question it answered: “What is leg?” instead of the correct reply of “What is: he’s missing a leg?” (Shah, 2013).

### Other Case Studies

Quarteroni and Manandhar (2007) used AIML to create a chatbot to answer questions using Google search results. They investigated how people phrase questions when speaking to chatbots using a ‘Wizard of Oz’ experiment – participants were asked to speak to what they believed was a chatbot system over chat while they were in fact speaking to a researcher.

They found several potential issues and attempted to modify the design of the system to counteract them. Firstly, they found that users sometimes asked two questions within one message. In response, the chatbot would look for the word “and” within the request and only answer one of the questions. Secondly, they found that users would ask follow-up questions that would require the chatbot to recognise they were referring to a previous request. The chatbot was designed to recognise follow up questions by recognising common patterns the questions would have – for example if the question repeats the nouns of the previous request or if it doesn’t contain any verbs.

Dowd’s (2011) case study details the development and evaluation of a chatbot created for the University of Wolverhampton to help students with queries they have about the library service.

The chatbot was tested against a group of students. The case study includes the result of a questionnaire they were given, and well as observations about how they tended to communicate with it. Most students typed short sentences, or one keyword and it was rare for them to ask lengthy questions. Several students engaged in small talk and were impressed by the chatbots ability to respond to it.

The paper states that the most common reason for failure was the use of unrecognised synonyms, for example, users asking for the expiry date for their book rather than the renewal date. They also had minor problems with misspellings and homophones – words that have the same pronunciation but have different spellings, such as ‘new’ and ‘knew’.

Yan et al. (2016) discuss the development process of a prototype chatbot service that uses a serverless platform. They used IBM’s OpenWhisk service and several API services to create a chatbot with six different functions: show the top news stories, tell the date, tell the weather forecast, answer music related questions and tell jokes.

They also detail the challenges that were encountered during development. For example, they found it difficult to debug the chatbot, especially when composing a series of functions that depend on each other. This was because there was a lack of tooling that allowed them to discern why the application failed such as detailed stack traces.

As the price plan for OpenWhisk depends on the applications total active time, it needed to be monitored carefully to minimise costs. However, the monitoring infrastructure is dependent on the serverless platform that is used. The application’s performance also relies on the platform used as well as any other outside services. Therefore, when deciding which hosting platform to use the monitoring infrastructure and Quality of Service (QoS) guarantees of each option should be considered.

There are also some potential security issues when using a hosting service and when connecting different services together they may have security flaws that expose data. It also means developers need to manage several tokens that are used to access each service.

Jain et al. (2018) designed a shoe shopping chatbot with the aim of overcoming the issue of a mismatch between user’s perception of the current ‘state’ of the chatbot and its actual state. They added an experimental feature called Convey (derived from “Context View”) which would display the conversational context and context values that the chatbot was currently assuming. For example, if a male user asks: “show me brown shoes”, the current context would be ‘search for shoes’ and the context values might be ‘male’ and ‘brown’ shoes, where ‘male’ was assumed from the user’s profile and ‘brown’ was inferred from the conversation. The Convey window would then show these values at the top of the page which could also then be edited or deleted by the user.

A similar chatbot without Convey was then created using IBM Watson. Sixteen participants were asked to try and complete the same two tasks using both chatbots. The ratings and feedback from participants showed that they found the version with Convey easier to use, faster and less frustrating. However, a small sample size was used, and most participants had a technical background which may have affected the validity of the study results. All participants also had previous experience using a messaging platform and considered themselves fluent in English. In addition, few of the participants had previous experience using a chatbot before.

## Advantages of Chatbots

Researching the advantages of chatbots and why they are used is useful because it helps build a clearer idea of what people expect from a chatbot service.

To explore the reasons why chatbots are sometimes preferred to the use of a search engine, Shawar and Atwell (2007a) created ‘FAQChat’, a chatbot aimed at university staff and students. After testing they asked participants if they preferred to use FAQChat or Google to find answers and why. Two-thirds of the participants preferred FAQChat, but even those who preferred Google responded favourably to the chatbot.

Those who preferred the chatbot gave several reasons for their choice. Firstly, the chatbot could usually give direct answers to questions while Google gave a list of links, meaning that an answer could be found more quickly. However, this may not still be as relevant as Google has since introduced ‘featured snippets’ which show a summary of a relevant website at the top of the page, so the answer can sometimes be found without having to click on a link.

Additionally, the chatbot could repeat the same information as many times as needed, whereas a person might get bored or irritated repeating themselves. Finally, many participants reported that they found it interesting to be able to access the FAQ using natural language as they had never used anything similar before.

McNeal, Michele and Newyear (2013a) also discuss the advantages of chatbots that they discovered while investigating the feasibility of introducing chatbots into libraries.

They state that chatbots are useful because they can remain unaffected by impolite customers or by being repeatedly asked the same questions. In addition, they note that chatbots can provide faster responses and can handle a higher volume of requests at a time than staff can. If a chatbot is connected to a data source, it can also ease the process of finding information by a “natural language interface”. Customers might find talking to a chatbot a more intuitive way of searching a database than learning the structures and language that would normally be required to query it. Finally, they also suggested that if a chatbot is anonymous then it could encourage users who were too shy to ask their question to a person because they were worried their question might be ‘stupid’.

Although chatbots were only investigated in terms of how they could specifically be beneficial for a library, many of the advantages listed could also apply to a chatbot used within a university.

Brandtzaeg and Følstad (2017) conducted a study where 146 people were asked about their motivations for using chatbots. The most frequent reason given (42% of participants) was that chatbots are quicker and more convenient to speak to than speaking to a person – chatbots can usually reply almost instantly whereas a person would need to look up information and take time to type a response. Chatbots are also typically always available whereas a person might not be available at some hours of the day or might be busy with another customer.

The next most frequent reason (41%) was the ease of obtaining help from a chatbot. As McNeal, Michele and Newyear (2013a) predicted participants reported that they were more comfortable asking a chatbot questions they thought might be seen as stupid. Some people also reported they felt more comfortable asking the same question multiple times within the same conversation.

Some participants (20%) also mentioned they used chatbots for entertainment or social purposes, even when entertainment is not the chatbot’s main purpose.

Finally, a few of the participants (10%) reported that they had used the chatbot because it was a novelty and they were curious to understand the chatbot’s limitations. This is similar to Shawar and Atwell’s (2007a) findings that some people preferred using the chatbot because it was new and interesting.

The paper also discusses the implications the findings have for chatbot design. As the most common reason for using a chatbot is response time and convenience, it is important for it to be available as constantly as possible and for it to respond quickly. In addition, because some participants use chatbots for entertainment purposes, Brandtzaeg and Følstad argue that chatbots should be designed to support “enjoyable social interactions” such as small talk. The implications of some participants using the chatbot to ask questions they would be uncomfortable asking a person is not discussed in paper but might suggest that users would find it very important to know that the chatbot is secure and confidential.

Overall this study provides a detailed report into the reasons why chatbots are used, however it is important to consider the limitations of the study; the participants were self-selected and consisted only of people who had previously used chatbots, and therefore may not be a representative demographic of the average user.

## Natural Language Processing Technologies

Natural language refers to any language that has evolved naturally in humans. Natural Language Processing (NLP) is a subset of artificial intelligence concerned with the techniques and methods that allow communication with computers using natural language (Lehnert and Ringle, 2014). NLP is an important component of chatbots as users interact with them primarily using natural language, which the chatbot needs to be able to understand and respond to.

### AIML and Related Technologies

AIML is a mark-up language derived from Extensible Mark-up Language (XML) and is commonly used to implement chatbots (Bradeško and Mladenić, 2012). It is used to define the chatbot’s input rules and its responses. AIML also uses pattern matching with \* or \_ as a wildcard which can be used to match any words (McNeal, et al., 2013b).

The AIML below shows an example of wildcard usage. If the user responds with “My name is John”, then the chatbot would reply “Hello John”.

<category>

<pattern> My name is \* </pattern>

<template> Hello <star/> </template>

</category>

Bradeško and Mladenić (2012) describe the features of different chatbot technologies including AIML and ChatScript. A main advantage of AIML is that it can be called recursively using the <srai> tag, meaning the output can be built from multiple templates.

ChatScript is the successor of AIML and has additional functionalities like logical and/or, variables, concepts. For example, the concept of fruit can be defined as:

~fruit (apple orange banana strawberry pear)

And can then be used within a template:

s: (What is a ~fruit?) It’s a type of fruit

ChatScript also contains over 2000 predefined concepts.

McNeal, Michele and Newyear (2013b) outline AIML’s advantages and disadvantages in their book “Library Technology Reports”. Firstly, AIML is simple as it follows only a few basic rules. This also makes it easy to learn and implement. As AIML is a popular option for chatbots there are a large number of pre-set AIML sets publicly which can reduce development time.

However, the pattern matching technique it uses is relatively weak compared to other options available. It can also be difficult to maintain and create the categories needed to create an intelligent and convincing chatbot, especially if there are no fitting pre-set rules already available. Several permutations need to be created for each question to ensure that any way a user phrases a question the chatbot will recognise it.

The advantages and disadvantages of ChatScript are also discussed. Like AIML, ChatScript uses a mark-up language and utilises pattern matching to define the chatbot’s responses. ChatScript has stronger and more flexible pattern matching as it has more features than AIML. The increase in number of features also has the drawback that it makes it more complicated and difficult to learn.

### Chatbot Engines

Chatbot engines are Natural Language Understanding (NLU) engines responsible for analysing input to a chatbot and processing into a “machine understandable” format (Kar and Haldar, 2016). There are currently several chatbot engines available to use as a service, the majority of which have been released within the last 5 years, such as DialogFlow, IBM Watson Conversation and Microsoft LUIS (Savenkov, 2017).

A common set of key concepts that are used by chatbot engines are intents, context and entities. Intents are used to represent the purpose of the user’s interaction with the chatbot (Canonico, and Russis, 2018).

Each time a user responds the intent of the message needs to be determined so that the chatbot can decide what action it needs to take. Entities are specific values within the user’s request that are needed to understand the intent. Context is used to keep track of the conversation to understand the current request using the conversation’s history (Kar and Haldar, 2016; Jain et al., 2018).

For example, the phrase “What’s the weather forecast for London tomorrow?” would have the intent ‘get weather forecast’ and the entities would be ‘tomorrow’ and ‘London’. If the chatbot was then asked: “What about for Edinburgh?” it could potentially use context to determine that the user still wanted to use the ‘get weather forecast’ intent and the ‘tomorrow’ entity.

Savenkov's (2017) "NLU / Intent Detection Benchmark" compares the intent detection performance, rate of false positives, response time, language coverage, and pricing of seven different NLU service providers. The benchmark compared DialogFlow (formerly Api.ai), wit.ai, Snips.ai, LUIS, IBM Watson Conversation and Amazon Lex and used the same dataset with the same 7 intents on all engines.

It was found that IBM Watson performed the best in terms of intent detection performance (calculated from precision and recall time) at 99.7% with DialogFlow a close second at 99.6%. Savenkov also noted that Snips.ai and DialogFlow are good at detecting when the user has asked something it hasn’t been given an intent for; other chatbot engines will try to select the closest matching intent. Another factor to consider is that wit.ai has the biggest support for different languages – this could be useful for a university that teaches many international students and may want to release the chatbot in multiple languages.

It should be taken into consideration that although the dataset used is publicly available it was provided by Snips.ai - one of the solutions that was benchmarked – so the results may be biased. However, Snips.ai performed the fourth best in terms of intent performance and worst in terms of recall time.

A similar comparison of different chatbot engines was created by Debasatwa Dutta (2017) while investigating the options available to create a chatbot for high school students. Chatbots were created using DialogFlow, Wit.ai, Luis.ai and Pandorabots.com and were then trained using the same knowledge base, then tested with the same questions. The number of correctly matched intents was then recorded for each chatbot, as well as the confidence score if it was available.

It was found that DialogFlow, Wit.ai and Luis.ai had correctly matched a very similar number of intents, while PandoraBots.com matched a much lower number. However, DialogFlow returned a lower confidence score for false matches than all the other platforms. Dutta noted that this is useful because DialogFlow has a “threshold” option which changes how high the confidence level needs to be before an intent is matched. Unfortunately, the comparison did not also cover IBM Watson Conversation, Snips.ai or Amazon Lex.

Canonico, M. and Russis, D.L. (2018) also compare DialogFlow, wit.ai, Microsoft LUIS, IBM Watson Conversation, Amazon Lex and Recast.ai. Each chatbot engine is judged on its usability, language and programming language support, number of pre-set intents and entities, availability of features such as a default fall-back intent, and price. IBM Watson, DialogFlow and LUIS were also rated on their intent detection accuracy.

They found that IBM Watson had the highest intent detection accuracy, followed by DialogFlow. DialogFlow and IBM Watson were also rated high usability, and only Amazon Lex was given a low rating. DialogFlow had the highest number of: pre-built intents and entities, number of integration options and the number of supported programming languages. Finally, they also reported that Wit.ai supported the highest number of languages.

## Cloud Platforms

Integration with a messaging platform or chatbot engine requires the chatbot to be hosted online so that it can send and receive data over the internet. If the chatbot cannot be self-hosted another option is to use a cloud hosting platform service instead.

A study by Patil, Marimuthu and Niranchana (2017) compares the different features of cloud hosting platforms for chatbot development. It compares Microsoft Azure, Heroku and IBM Watson’s built in hosting service for IBM Watson Conversation.

Microsoft Azure is the most well documented but has the fewest supported programming languages and requires an Azure membership which is a paid service. Heroku is integrated with Git and has simple automated deployment. However, it requires programming skills and has a steep learning curve, seeing as it is complex. IBM Watson Bot Service on the other hand is easier to use, and minimal programming skills are needed. Little setup is required in comparison to the other options but only for Watson services.

The study concludes that all platforms have different features and functionalities should be chosen based on requirements of individual projects. For example, when developing a chatbot using IBM Watson Conversation, the in-built service may be the most obvious choice. Otherwise, if Git integration is important or if a free service is needed then Heroku could be the best choice instead.

## Appearance

An important aspect that needs to be taken into consideration when designing a chatbot is its appearance. In addition to the text input and output component, some chatbots also include an avatar. In some cases, especially when the chatbot uses text-to-speech, the avatar is animated and can gesture as though it was talking.

Kuligowska (2015) argues that because people often judge things by their appearance, the chatbot's avatar is a very important factor in its success. When evaluating chatbots he rates those that more closely resemble living people higher than those with more cartoon-like avatars or without any visual representation at all. He also gives higher ratings to chatbots that use text-to-speech and that have an animated avatar which can move in accordance to its responses so that it appears that it is talking. However, the paper focus on commercial chatbots that are used by customers as opposed to information retrieval chatbots, so might not be relevant to a chatbot created for a university.

Figure 1: Ikea’s retired chatbot, Anna (BBC, 2016)

On the other hand, Ben Mimoun, Poncin and Garnier's (2012) state that a more realistic avatar can be a cause of failure for a chatbot as it can create realistic expectations. They found that users accept a chatbot with an avatar that appears human to have the same range of knowledge and intelligence as a real human – one of the experts that was interviewed reported that customers were frustrated because “the agent has a very sophisticated visual display, but it is not able to answer all users’ questions”. One example they gave was Anna, a chatbot created by Ikea. Although it initially increased user activity it was eventually removed due to its poor performance relative to its appearance.

In addition, Ciechanowskia, Przegalinskab, Magnuskia and Gloorc (2018) published details of the results of an experiment where participants were asked to speak to either a simple text chatbot with no avatar, or a chatbot with a human avatar that delayed its responses slightly to give the effect that it was typing them out. Their primary goal was to investigate the effects of these differences on the 'uncanny valley' effect – an uncomfortable feeling that is experienced when interacting with something that has human-like features but is not completely human.

They found that the participants preferred interacting with the simpler chatbot because they were less stressed as talking to the more human-like chatbot gave the uncanny valley effect. This study was also conducted very recently so it may provide a more up to date idea of what people prefer.

## Evaluation Methods

Shawar and Atwell (2007b) used three evaluation metrics to evaluate their AIML based chatbot:

* Dialogue efficiency in terms of matching type - the ability to recognise intent and which approach was used to match the sentence to the intent. For example, an ‘atomic match’ occurs when the user’s sentence matches one of the intent templates exactly.
* Dialogue quality metrics based on response. Three different categories were used -reasonable, related but unusual or nonsensical.
* User satisfaction. This was assessed based on feedback from testers.

They also suggest that the evaluation method used should be adapted to suit the chatbot’s specific purpose and functions. For example, chatbots built for the Loebner Prize should be evaluated differently to a chatbot used in customer service. The former chatbot would aim to appear human and would have an open domain, whereas the latter would have a more closed domain and may not need to seem human.

[Chakrabarti](https://www.sciencedirect.com/science/article/pii/S0957417415003097) and Luger (2015) assess the effectiveness of a chatbot’s conversational abilities using four different attributes:

* The quality of the chatbot’s speech, namely the accuracy of the information it provides
* The quantity of the chatbot’s speech is appropriate, and isn’t too long or short
* The relativeness of the information provided to the current context
* The manner the chatbot uses is direct and clear, and isn’t ambiguous

The first quality is the easiest to assess as it is objective and can be distinguished to be either true or false. The other three are subjective and are therefore harder to assess reliably. To evaluate the overall effectiveness of a chatbot they calculate the ratio of successful conversations out of the total number of conversations.

A widely used evaluation method is the PARADISE Framework (Cahn, 2017). It separates quality factors into two categories, subjective factors and objective factors. Subjective factors include ease of usage, clarity, naturalness, friendliness, robustness regarding misunderstandings, and user willingness to use the system again. As these qualities are subjective they are usually measured using user ratings through the distribution of questionnaires.

Objective qualities include maximising task success (e.g. an information retrieval chatbot giving correct information) and minimising ‘dialogue costs’. Dialogue costs include the time and number of responses needed to complete a task.

Radziwill and Benton (2017) researched and categorised quality attributes from 32 papers and 10 articles. The main category areas were efficiency, effectiveness and satisfaction.

The attributes for efficiency include robustness to unexpected input and manipulation and providing “appropriate escalation channels” such as switching contact to a human.

The attributes for effectiveness include speech synthesis, the ability to maintain themed discussion, accuracy of responses and general ease of use. They also found conflicting opinions between sources for whether a chatbot should pass the Turing Test as well as whether it should reveal that it is a chatbot or not. It was concluded that as mainly older sources prioritised developing a human-like chatbot and newer sources generally found users could usually tell that a chatbot was not human and were indifferent that it shouldn’t be considered a quality attribute.

Finally, the attributes for satisfaction included a pleasant personality by providing greetings and conveying emotion, protecting and respecting privacy and ability to detect meaning and intent. The paper also suggests prioritising attributes that relate to performance and accessibility over other categories.