

## Cheshire: A New Believable Chat Bot Using AIML, LSA, Emotions from Personalities, and Voice Recognition and Synthesizer

Sang Suh  
Texas A&M University-Commerce, USA  
sang.suh@tamuc.edu

Stephen Baxter  
Texas A&M University-Commerce, USA  
stebaxter22@gmail.com

John Carbone  
Forcepoint, LLC, USA  
John.Carbone@forcepointgov.com

Patrick Then  
Swinburne University of Technology  
pthen@swinburne.edu.my

### Abstract

Since the 1950s, people have been trying to create a more believable chatbot. The Standard Turing Test (STT) has generally been used to test them. Development of chatbot initiated with pattern recognition with Eliza in 1966 and PARRY in 1972, further with AI by Jabberwacky, and AIML with ALICE in 1995. Since then, people have tried adding nonverbal features, personalities, and audio input and output features. The goal of this research is to use these advancements to create a chatbot believable enough to pass the STT. To do this in a different way than most other chatbots, this new chatbot will use AIML with LSA to generate a response for every query without the need of a default response, derive and use the emotional ton of the user input along with a selected personality to apply an emotional ton to the response, and provide a means for the user to talk to the chatbot and for the chatbot to talk back.

**Keywords:** Chatbot, AIML, LSA, Personality

### 1. Introduction

For over half a century, programmers have been trying to create a believable chatbot. By advancing interior functionality, adding visuals for nonverbal communication, adding pseudo personality, and adding audio and voice recognition, chatbots have become nearly indistinguishable from real people but it's not perfect yet.

In 1950, Alan Turing asked the question, "Can machines think?" (Adamopoulou and Moussiades, 2020). To answer this question, the Standard Turing Test (STT) was created. When a person is presented with two choices, one being a human and the other being a machine and cannot tell the difference between

the two, then the machine is believed to pass the STT (Adam, 2019). The first accepted chatbot, Eliza, was created by Joseph Weizenbaum in 1966 to simulate a psychotherapist. It won the Loebner Prize in 1991 (Shin et al., 2022). Eliza used pattern matching and a template-based response mechanism. Despite Eliza being capable of passing the STT, the chatbot was not always capable of passing the test and its success may have been partially due to everyday people not fully understanding how computers worked yet (Adamopoulou and Moussiades, 2020). In 1972, PARRY was created as a schizophrenia (Dirani, 2021) by Kenneth Colby and also used the same pattern-response as Eliza but with better language understanding and other features. PARRY was the first chatbot recognized to pass the STT (Adam, 2019). The first chatbot to use Artificial Intelligence (AI) was developed by Rollo Carpenter in 1982 and was called Jabberwacky (Dirani, 2021). The first chatbot to earn the title "most human computer" was 1995 ALICE (the Artificial Linguistic Internet Computer Entity) (Adamopoulou and Moussiades, 2020, Dirani, 2021). ALICE also won the Loebner Prize, the annual STT in 2000, 2001, and 2004. ALICE's main power was on using the Artificial Intelligence Markup Language (AIML) (Adamopoulou and Moussiades, 2020).

AIML was developed between 1995 and 2000. Its structure was drawn from a structure based on XML markup language as shown below (Adamopoulou and Moussiades, 2020).

```
<aiml version="1.0.1" encoding="UTF-8">
  <category>
    <pattern> My name is * </pattern>
    <template>Hello <star/></template>
  </category>
</aiml>
```

As given in the above code, the language is tag based, where <category> is the basic unit of dialog, <pattern>

is something the user could say, and <template> is the chatbot's response. Based on pattern matching technique, AIML uses conversation between users and chatbots, following the stimulus-response approach with natural language modeling to emulate a conversation. AIML enables chatbots to generate general responses to user queries as long as the patterns are recognized. However, when it is combined with Latent Semantic Analysis (LSA), chatbots can generate responses to queries not recognized by AIML alone. LSA uses vector representation to find the similarities of words (Adamopoulou and Moussiades, 2020). If the user input isn't recognized by the AIML alone, then the LSA combined with cosine similarity (Günther et al., 2016, Simmons and Estes, 2006) may be used to find the best pattern the user inputs may match with and generate matching output of the chatbots.

Many chatbots may use the above methods to create a chatbot but to make sure this chatbot is believable enough to pass the STT, additional features need to be considered and adopted, in addition to AIML and LSA to generate a response to the user via text and from text. Incorporating personality types, specifically Myers-Briggs Type Indicator (MBTI) and Enneagram, into the Cheshire chatbot serves to enhance its interactions with users. These personality frameworks provide a deeper understanding of individual differences, enabling Cheshire to tailor its responses to align with the user's communication style and emotional needs. This could lead to more personalized and engaging interactions.

By using both MBTI and Enneagram, Cheshire can generate a wider variety of responses, making the chatbot more interesting and engaging for users. Furthermore, it allows Cheshire to adapt its responses based on the user's emotion and the selected personality type, leading to more satisfying interactions. The use of these personality types also aids in better understanding its users, which can lead to improved performance over time as it learns to better match its responses to the user's personality type and emotional state (Fernau et al., 2022 and Makwana and Dave, 2020). Thus, the integration of these personality types provides Cheshire with a unique advantage in delivering more personalized, emotionally aware, and diverse responses. To make chatbots behave like humans in interaction, MBTI (Verywellmind) and Enneagram (Truity) may be considered to handle personalities and emotional tones in interaction and generation of responses. User styles and personalities

may be used to apply emotional tones to the output text generated by chatbots. AIML responses, their emotional variants, the emotional tone of the user's input, the ability to take input from the user's voice via speech recognition, and a voice synthesizer to output the response as speech are used to generate the appropriate emotions for the output. The key feature of this chat bot is LSA with cosine similarity eliminating the need for default responses. The LSA will be used with cosine similarity (Günther et al., 2016, Simmons and Estes, 2006) to find which AIML pattern is the most similar to the user input and feed that pattern into the AIML chatbot instead of the user input. This way the AIML chatbot will always have a response that matches with one of the templates and does not need a default response as in the chatbots introduced in (Kumar et al., 2019) and (Lakshmi et al., 2019) when a response cannot be given from the algorithms they used. The paper is organized as follows. Section 2 describes related work on the chatbots and related technologies, followed by Section 3 describing an integrated approach toward the proposed believable chatbot. Section 4 contains detailed description of implementation of Cheshire followed by conclusion and future work in Sections 5 and 6.

## 2. Related Work

The chatbots described in (Nißen et al., 2022), (Adamopoulou and Moussiades, 2020), and (Cahn, 2017) cover the underlining structure of chatbots. These papers were used to help understand the history of chatbots and how they are developed. Furthermore, there have been various efforts in developing chatbots using AIML technique at the core to implement chatbot that are human like (WALLACE, 2023, Arunkumar et al., 2020, Roos, 2018, Chandan et al., 2019, Yamaguchi et al., 2018, and Arsovski et al., 2017). Some chatbots have used latent semantic analysis in their implementation to handle wide range of questions (Kumar et al., 2019 and Lakshmi et al., 2019). More details on how the latent semantic analysis works is covered in a book (Landauer et al., 2011). These papers were used to help understand how a AIML could be used in a chatbot, along with LSA.

The latent semantic analysis combined with cosines similarity is used other chatbots to find similarities between words and sentences (Günther et al., 2016 and Simmons and Estes, 2006). With LSA and cosines similarity, the user input is added to the patterns and are put in a list, a 2d chart is created with

the elements of list making up the x and y axis. The chart shows how similar each element is to every one of the elements. As such the last comparison is between the user input and the user input, so to find what pattern is the most similar to the user input, we look for the index of highest value that isn't the last comparison on the final row or column.

In other cases, the large language models like ChatGPT from OpenAI were used to paraphrase sentences (Wahle et al., 2022 and Witteveen and Andrews, 2019). Likewise, to generate the data for this paper, the ChatGPT was used to create the categories for the AIML file and the emotional tone variants for the templates.

Furthermore, various efforts have been made to explore the possibility of adding a personality to make a Chatbot more human-like (Sutoyao et al., 2019, Higashinaka et al., 2018, Galvao et al., 2004, Morales-Rodríguez et al., 2010, Shin et al., 2022, Farah et al., 2021, and Adam, 2019). These helped with the creation of the method described in this paper to apply emotions to Cheshire, but were not able to explain how the personalities used in this paper would react to the emotional tones used in this paper.

The website described in (Verywellmind) covers information about MBTI personalities and the website in (Truity) covers information about Enneagram personalities. The works by (Fernau et al., 2022 and Makwana and Dave, 2020) go into detail on how MBTI could be used and it benefits. No research papers could be found that describe in detail on what emotional tone MBTI or Enneagram would react to when exposed to the emotional tones used in this paper. The algorithms used in this paper are derived from the interpretation of the information on the websites.

The possibility of using Emojis and other nonverbal forms of communication have been explored in order to allow a chatbot to be more approachable (Kang, 2021, Fadhi et al., 2018, Dirani, 2021, and Li et al., 2017). However, they found that these are not suitable for more serious subject matters. Although no such features were added to Cheshire as they are related to 3D design, the research in that direction remains as future works.

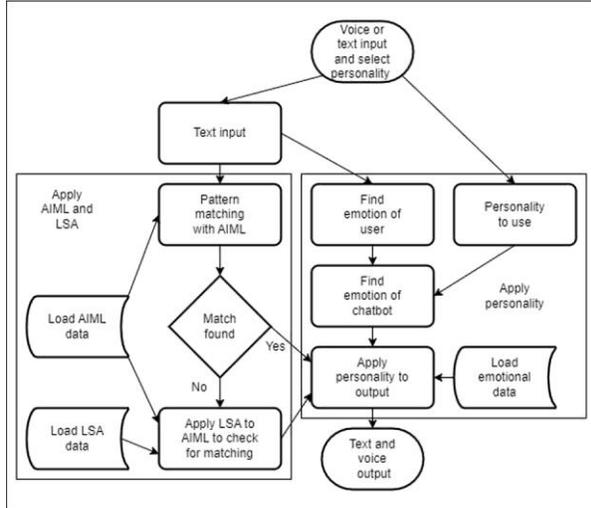
A few chatbots cover topics on software to deal with people's response to chatbots (Kuligowska, 2015).

One of the things covered is the ability to receive input from a user via voice and the ability to respond via speech synthesizer. Some chatbots used Microsoft voice synthesizer to give speech recognition to a AIML chatbot (Ahmed and Singh, 2015).

### **3. An Integrated Approach Toward Believable Chatbot**

This chatbot presented in this paper and called Cheshire will collect the user inputs and send them to AIML to start, then switch to LSA if a predictable response can't be found. Then a personality is applied to the output depending on the emotional tone of the user's input.

Figure 1 shows the flow of how the proposed chatbot operates and responds to the input. It starts with obtaining "Voice or Text input and select personality" which gets "Text input" and "Personality to use" from the user then splits into two paths; "Apply AIML and LSA" and "Apply personality". "Apply AIML and LSA" module takes in "Text input" and outputs a base response. "Pattern matching with AIML" module takes the "Text input" and "Load AIML data" then performs pattern matching. If a match is found then it sends the base response to Apply Personality module. Otherwise, Apply LSA to AIML to check if matching module takes text input. "Load AIML data", and "Load LSA data" then apply LSA and cosine similarity to find the best input for matching to send the base response to "Apply personality". "Apply personality" takes in "Text input", "Personality to use", and base response then outputs "Text and voice output". "Find emotion of user" module takes in "Text input" and obtains the emotion from it and send to "Find emotion of chatbot". "Find emotion of chatbot" takes it along with "Personality to use" and output the emotion of the chatbot to send it to "Apply personality to output". "Apply personality to output" takes the "Find emotion of chatbot" output and "Load emotion data", and then applies a personality to the base response and generates output to "Text and voice output". "Text and voice output" outputs the personality response to the user via text and voice synthesizer.



**Figure 1. A Consolidated Approach Toward Believable Chatbot**

## 4. Experiment

The data for Cheshire was mostly created with ChatGPT by OpenAI. The “cheshireAIML3.aiml” holds the categories for the aiml. The “aiml\_options.csv” holds the LSA and emotional variants for the templates. The structures of “aiml\_options.csv” is nine columns; “pattern”, “that”, “topic”, “template”, “Happy”, “Angry”, “Surprise”, “Sad”, and “Fear”. The “Happy”, “Angry”, “Surprise”, “Sad”, and “Fear” columns hold the variants for the template. Each row contains every possible combination of patterns and templates along the emotional variants for templates. The columns “that” and “topic” were not used due to how simple the AIML files had to be. The LSA data uses the pattern column with the duplicates taken out and the chart holding the aiml responses and their emotional variants use the “template”, “Happy”, “Angry”, “Surprise”, “Sad”, and “Fear” columns after duplicates have been removed. The prompt that was used to get the variants was “rephrase the “<template>” five times; one with the emotion of happy, one with the emotion of angry, one with the emotion of surprise, one with the emotion of sad, and one with the emotion of fear.”

For Cheshire, Python 3.10 with Microsoft’s Visual Studio Code was used as the program language and code IDE respectfully. The modules aiml (0.9.2) and python-aiml (0.9.3) were used for the AIML functionality. The modules pandas (2.0.2) and numpy (1.24.3) were used for loading and handling data. The

module sklearn (0.0.post5) was used for LSA and cosine similarity. The module text2emotion (0.0.5) was used for deriving emotion from the user input text. The module text2emotion can only detect “Happy”, “Angry”, “Surprise”, “Sad”, and “Fear” hence the reason these 5 emotions were used in this paper. Finally, the module Flask (2.3.2) was used to create a HTML, CSS, and JavaScript front end so the JavaScript voice recognition and synthesizer functions could be used and so that a website could be created for having a large amount of people to test the chatbot.

The NLTK module used by text2emotion uses external files necessitating these files to be checked for updates. If updates are needed or any of the files are missing, they can be downloaded to replace older files. This is done by default by text2emotion.

Before any functionality is implemented, the errors in aiml (0.9.2) and pyttsx3 (2.90) needed to be fixed. For aiml (0.9.2), time.clock needs to be changed to time.time (code: time.clock = time.time) and for pyttsx3 (2.90), emoji.UNICODE\_EMOJI needs to be changed to emoji.EMOJI\_DATA (code: emoji.UNICODE\_EMOJI = emoji.EMOJI\_DATA). The pyttsx3 (2.90) error is from the use of emoji (2.2.0). These are done with “FixAIMLTimeError” and “FixText2EmotionEmojiError” functions.

Next, classes and data need to be set up. First the aiml kernel is created using the “CreateAimlKernel” along with the “std-startup.xml” file that points to the aiml file “cheshireAIML3.aiml” to train the aiml with. Second, the chatbot can use the LSA data holding all the patterns and the chart holding the aiml responses and their emotional variants are loaded in with the “GetAIMLOptionData” function using the “aiml\_options.csv” file.

Once the classes and functions are setup, the main loop begins.

First, the user input and the personality selection are retrieved. For user input it is done through text input unless the user speaks into the computer microphone and the result is turned into text. For personality selection, a drop-down list containing both MBTI and Enneagram types is used. After the text input and personality are set the user hits the submit and the information is posted to the back end for

processing. If no personality is selected then the MBTI “ISTJ” type is used.

Second, the chatbot base response is gathered. It first passes the user input through the aiml kernel using the “GetAIMLOutput”; failing that, the user input is then passed through the LSA and cosine similarity to find what aiml pattern best matches the user input and send the pattern through the aiml kernel with the “GetAIMLOutputWithLSA” function. The “GetAIMLOutput” function simply feeds the user input into the aiml kernel and returns the results. The “GetAIMLOutputWithLSA” function works by combining the user input with the LSA data. Then a LSA is created using the combined data, after which a similarity matrix is created by applying cosine similarity to the LSA. The index of the second most similar item to the user input in the matrix can be located. This can be done since the most similar item is the user input itself. Then, the index is fed back into the combined data and the corresponding pattern is fed into the aiml kernel. Finally, the aiml kernel response is returned.

Third, a personality is applied to the base response. It first finds the emotion of the user by using the “GetUserEmotion” function and the user input. The emotion of the chatbot determined by using the “predict\_emotion” function, personality type, and the emotion of the user. The “GetPersonalityOutput” function, the chart holding the aiml responses and their emotional variants, the emotion of the chatbot, and the base output of the chatbot are used to apply personality to the chatbot base response. The “GetUserEmotion” function uses text2emotion to take text and turn it into a python object with five floating numbers from 0.0 to 1.0 labeled "Happy", "Angry", "Surprise", "Sad", and "Fear". The function returns the label of the highest one. The “predict\_emotion” function sets up an object containing the MBTI personalities and what emotions they react to, sets up an object containing the Enneagram personalities and what emotions they react to, selects the object the personality type is in and puts its emotions into a list. If the emotion of the user matches with one in the list then it returns that emotion, else if the emotion of the user is not one of the five emotions, then it returns “template” (this is done because “GetUserEmotion” function could return “None” instead of one of the five emotions), else the zeroth emotion in the list is returned. The “GetPersonalityOutput” takes the chart holding the aiml templates and their emotional variants, finds the

index of the template column that matches the aiml output, uses the index on the column corresponding to the catboat's emotion and returns its variant. If there is no variant or if the chatbot’s emotion is a “template”, then it keeps the aiml kernel output as is.

Finally, the website front end is re-render with the chatbot output. The website will show the user input and the chatbot’s output. The output is then passed through a JavaScript voice synthesizer to output it through the speakers.

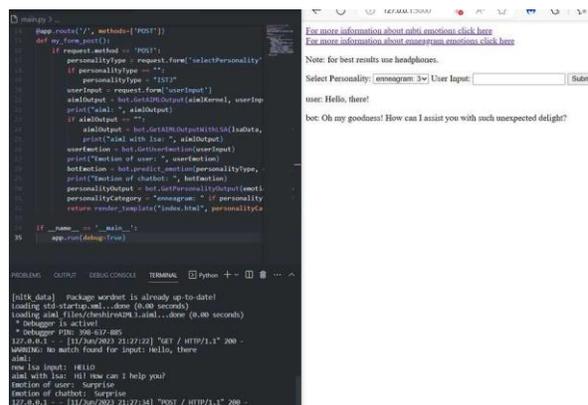


Figure 2. A Snapshot of Cheshire Running

Figure 2 shows a snapshot of Cheshire running in the Visual Studio Code command line and Microsoft Edge. “Hello, there!” does not match with any pattern. However, the LSA with cosine similarity found “HELLO” which is in patterns and the aiml returns “Hi! How can I help you?”. The “GetUserEmotion” function returns “Surprise” that is one of the emotions that Enneagram 3 will react to, and Cheshire returns “Oh my goodness! How can I assist you with such unexpected delight?” which is the “Surprise” variant to “Hi! How can I help you?”

## 5. Conclusion

To get a better view of how well Cheshire works compared to others, a table of comparison among the chatbots that were found online is shown in Table 1 in terms of various features of chatbots.

	Detects emotions	Uses emojis to show emotions	Uses default response	Uses 3D model to show emotions	Uses voice recognition	Text input	Uses different personalities	Uses AIML	Uses LSA to compensate for AIML	Text output	Uses voice synthesizer	General purpose
ALICE	No	No	N/a	No	No	Yes	No	Yes	No	Yes	No	Yes
Android assistance chatbot	N/a	Yes	N/a	No	Yes	Yes	No	N/a	N/a	Yes	Yes	Yes
Drift chatbots	No	Yes	N/a	No	No	Yes	No	N/a	N/a	Yes	No	No
Reve Chatbot	No	Yes	N/a	No	No	Yes	No	N/a	N/a	Yes	No	No
Chatbot in Python	No	No	Yes	No	No	Yes	No	Yes	No	Yes	No	Yes
DISCAL	No	No	Yes	No	No	Yes	No	Yes	Yes	Yes	No	Yes
Persona-AIML	No	No	N/a	No	No	Yes	Yes	Yes	No	Yes	No	Yes
ABVEAIC	No	No	N/a	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes
Cheshire	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

**Table 1. Comparative Analysis of Various Chatbots**

Table 1 shows a comparative analysis of various chatbots. This way a better view of how Cheshire compares with other chatbots can be obtained. Cheshire has the ability to detect the emotional state which makes it unique. Chatbots used in this comparison are:

- ALICE. This is an AIML chatbot that used an extensive and interconnected list of castigators to handle almost any situation. (WALLACE, 2023)
- Android assistance chatbot. This is the chatbot built into Google's Android 13.
- Drift's website chatbot. This is the chatbot provided by Drift on their website, whose purpose is to provide information about the Drift company. (<https://www.drift.com/>)
- Rave Chat's website chatbot. This is the chatbot provided by Rave Chat on their website, whose purpose is to provide information about the Rave Chat company and to schedule demos on Rave Chat's more capable chatbots. (<https://www.revechat.com/>)

- Chatbot in Python. This is an aiml powered chatbot created with python. (Kumar et al., 2019)
- DISCAL. This is a chatbot that uses AIML and LSA to return a response to the user. If AIML can't find a response then LSA is used and if the LSA cannot find a response then a message to admin. (Lakshmi et al., 2019)
- Persona-AIML. This paper proposes a chatbot that uses AIML with a personality component to add actions and beliefs to the AIML categories. (Galvao et al., 2004)
- ABVEAIC. This chatbot uses AIML and natural language processing to handle speech to text and text to speech. (Ahmed and Singh, 2015)

The main features described were implemented as proposed with some limitations but further work and research needs to be done to create the chatbot fully functional as proposed and achieve the goal of developing truly human-like believable chatbot. Although AIML with LSA was implemented the number of categories used in our experiments was limited in size, thus what it can talk about is somewhat limited. The function of the website was deployed with core tasks implemented, limiting the user to follow

links provided to learn more about MBTI and Enneagram personalities.

Using LSA with cosine similarity responses can be given without the need of a default response to act as a stop gap or with an extensive AIML file.

## 6. Future Work

As shown in the “Experiment” section of this paper, the AIML with LSA can handle patterns not defined by the AIML data, so having categories with robust pattern options is not necessary, but the current categories are limited. A wider variety of categories would provide the user and Cheshire with more subjects to talk about. Section 3 describes “Apply AIML and LSA” and “Apply Personality” which require specialized data from simple AIML file in which the categories can only contain a pattern and template with the template only being text. Specialized AIML files would have to be created to allow for more generic AIML files. A tool could be created that uses a large language model (Wahle et al., 2022) to automatically generate emotional variants for the responses.

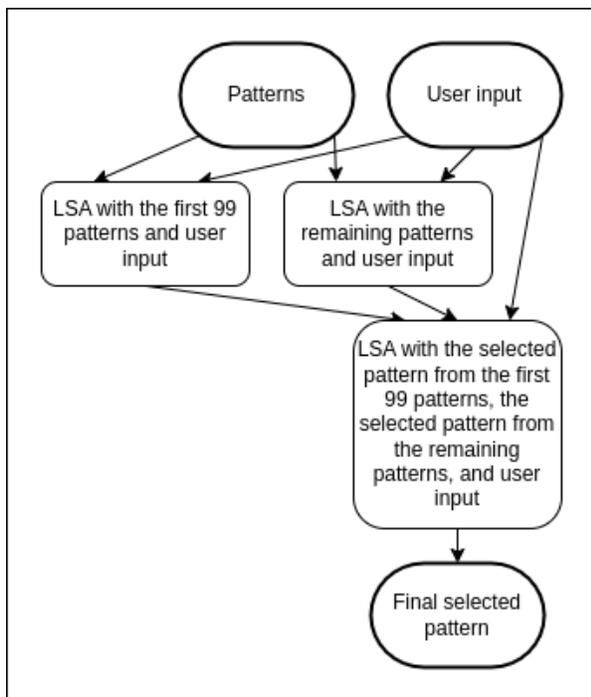


Figure 3. Possible Solution for Using More Than 99 Patterns

The function used in LSA with cosine similarity can only handle about 100 patterns plus the user input. To handle more inquiries, a new algorithm needs to be created. One method could be to split the patterns into two or more groups and apply the LSA with cosine similarity to each groups comparing them to the user input, then comparing the resulting patterns to the user input one more time to find final pattern to use. Figure 3 shows how this would work with a list of patterns that is greater than 99 but less than 198. Using multiple layers of LSA with cosine similarity, Cheshire could use more than 99 patterns, however this could insincere the processing power needed and the time of each user input. This could limit what servers it could run on and if the time requirement is too high, then the user’s experience may be harmed. Testing will have to be done to find the right balance between quantity of stable inputs and system limitations.

Nonverbal communication could be added to make it more believable. However, it was found that it is not always suitable to use nonverbal communication such as Emojis (Kang, 2021) and (Fadhi et al., 2018). So, the chatbot should determine the emotional state of the user’s inputs first and only use emojis or some other system if the user’s input isn’t serious. A 3D face could be rendered on the website to change its expression depending on the user emotion and the personality.

Long-term memory could be implemented. AIML allows for a bot to remember but since the AIML files have to be simple, this isn’t facilitated. There could also be some insistence from each user but this would require the website to be fully implemented to allow for a user to log in and be remembered by Cheshire.

Finally, the question of whether or not such a system described in this paper could pass the STT was not answered. If the previous changes could be made and people could test the chatbot for themselves, then a better understanding of how a chatbot described in this paper could be made to pass the STT.

## 7. Acknowledgment

This research has been supported by National Science Foundation under S-STEM computing grant #: 2221189. Further, special thanks are given to Swetha Pathuri for her research on MBTI and Enneagram personality types and experiments on various chatbots for the comparisons and her input on different features

that could be implemented in future work including generation of greeting statements and premade statements to support more smooth interaction with the users.

## 8. References

- Nißen, M., Selimi, D., Janssen, A., Cardona, D., Breitner, M., Kowatsch, T., Wangenheim, F., 2022, "See you soon again, chatbot? A design taxonomy to characterize user-chatbot relationships with different time horizons", *Computers in Human Behavior* Volume 127, 107043.
- WALLACE, R., 2003, "The Elements of AIML Style", ALICE A. I. Foundation, Inc.
- Arunkumar, B., Balaji, A., Prasanth, V., Kumar, S., 2020, "Development of Chatbot Automation Application – DGCT CSE ALEXA", *International Research Journal of Engineering and Technology (IRJET)*, Volume: 07, Issue: 03, pp. 4498-4501.
- Roos, S., 2018, "Chatbots in Education: A Passing Trend or A Valuable Pedagogical Tool?", Department of Informatics and Media Uppsala University.
- Chandan, J., Chattopadhyay, M., Sahoo, S., 2019, "Implementing Chatbot in Educational Institutes", *International Journal of Research and Analytical Reviews (IJRAR)*, Vol. 6, Issue 2, pp. 44-47.
- Yamaguchi, H., Mozgovoy, M., Danielewicz-Betz, A., 2018, "A Chatbot Based On AIML Rules Extracted from Twitter Dialogues", *Communication Papers of the Federated Conference on Computer Science and Information Systems* pp. 37–42.
- Arsovski, S., Cheok, A., Idris, M., Raffur, M., 2017, "Analysis of The Chatbot Open-Source Languages AIML and Chatscript: A Review", Preprint.
- Adamopoulou, E., Moussiades, L., 2020, "An Overview of Chatbot Technology", IFIP (International Federation for Information Processing) International Conference on AI Applications and Innovations (AIAI20), pp. 373-383.
- Kumar, A., Meena, P., Panda, D., Sangeetha, 2019, "Chatbot in Python", *International Research Journal of Engineering and Technology (IRJET)*, Vol. 6, Issue 11, pp. 391-395.
- Lakshmi, K., Reddy, Y., Kireeti, M., Swathi, T., Ismail, M., 2019, "Design and Implementation of Student Chat Bot using AIML and LSA", *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* ISSN: 2278-3075, Volume-8 Issue-6, pp. 1742-1746.
- Landauer, T., McNamara, D., Dennis, S., Kintsch, W., 2011, "Handbook of Latent Semantic Analysis", Routledge, NY.
- Sutoyoo, R., Chowandaa, A., Kurniatia, A., Wongsoa, R., 2019, "Designing an Emotionally Realistic Chatbot Framework to Enhance Its Believability with AIML and Information States", 4th International Conference on Computer Science and Computational Intelligence 2019 (ICCS CI), pp. 621-628.
- Higashinaka, R., Mizukami, M., Kawabata, H., Yamaguchi, E., Adachi, N., Tomita, J., 2018, "Role play-based question-answering by real users for building chatbots with consistent personalities", In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 264–272, Melbourne, Australia, Association for Computational Linguistics.
- Galvao, A., Barros, F., Neves, A., Ramalho, G., 2004, "Persona-AIML: an architecture for developing chatterbots with personality", *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems 2004, AAMAS 2004*, pp. 1266-1267.
- Morales-Rodríguez, M., González B., Juárez, R., Huacuja, H., Flores, J., 2010, "Emotional Conversational Agents in Clinical Psychology and Psychiatry", *MICAI 2010, Part I, LNAI 6437*, pp. 458–466.
- Ahmed, I., Singh, S., 2015, "AIML Based Voice Enabled Artificial Intelligent Chatterbot", *International Journal of u- and e- Service, Science and Technology* Vol.8, No.2, pp. 375-384.
- Cahn, J., 2017, "Chat Bot: Architecture, Design, & Development", Senior Thesis (EAS499) University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science.
- Shin, H., Bunosso, I., Levine, L., 2022, "The influence of chatbot humor on consumer evaluations of services", *International Journal of Consumer Studies*.
- Farah, J., Sharma, V., Ingram, S., Gillet, D., 2021, "Conveying the Perception of Humor Arising from

- Ambiguous Grammatical Constructs in Human-Chatbot Interaction", HAI '21: Proceedings of the 9th International Conference on Human-Agent Interaction, pp. 257-262.
- Fadhi, A., Schiavo, G., Wang, Y., Yilma, B., 2018, "The Effect of Emojis when interacting with Conversational Interface Assisted Health Coaching System", Pervasive Health '18: Proceedings of the 12th EAI International Conference on Pervasive Computing Technologies for Healthcare, pp. 378-383.
- Kang, M, 2021, "Non-verbal Emotional Expressions for Social Presence of Chatbot Interface 챗봇의 사회적 현존감을 위한 비언어적 감정 표현 방식", The Journal of the Korea Contents Association (한국콘텐츠학회논문지), Volume 21, Issue 1, pp. 1-11, 1598-4877(pISSN) / 2508-6723(eISSN).
- Dirani, C., 2021, "Chatbots and Emojis for an Improved Human Experience", CCTP 607: Leading Ideas: AI to the Cloud.
- Adam, E., 2019, "Bachelor's Thesis Improving the User Experience of Chatbots with Tone Analysis", Chairman of the Examination Board.
- Kuligowska, K., 2015, "Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents", Professionals Center for Business Research, Vol. 2.
- Li, J., Lee, Y., 2017, "Multimodal Interaction and Believability: How can we design and evaluate the next generation of IPA", Proceedings of the 31st International BCS Human Computer Interaction Conference (HCI 2017)(HCI).
- Günther, F., Dudschig, C., Kaup, B., 2016, "Latent Semantic Analysis Cosines as a Cognitive Similarity Measure: Evidence from Priming Studies", Quarterly Journal of Experimental Psychology, 69(4), pp. 626-653.
- Simmons, S., Estes, Z., 2006, "Using Latent Semantic Analysis to Estimate Similarity", In Proceedings of the annual meeting of the cognitive science society, Vol. 28, No. 28.
- Wahle, J., Ruas, T., Kirstein, F., Gipp, B., 2022, "How Large Language Models are Transforming Machine-Paraphrased Plagiarism", 10.48550/arXiv. 2210.03568.
- Witteveen, S., Andrews, M., 2019, "Paraphrasing with Large Language Models", ArXiv, abs/1911.09661.
- Verywellmind, Myers-Briggs Type Indicator: The 16 Personality Types - <https://www.verywellmind.com/the-myers-briggs-type-indicator-2795583>
- Truity, What Are the Nine Enneagram Types? - <https://www.truity.com/enneagram/9-types-enneagram>
- Fernau, D., Hillmann, S., Feldhus, N., Polzehl, T., Möller, S., 2022, "Towards Personality-Aware Chatbots", Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue.
- Makwana, K., Dave, G., 2020, "A Study of Identification of Personality Profiles of Undergraduate Management Students Using Myers Briggs Type Indicator (MBTI) Test", Pacific Business Review International Volume 12 issue 8