AIML and Sequence-to-Sequence Models to Build Artificial Intelligence Chatbots: Insights from a Comparative Analysis

Nishant Teckchandani*[1], Aditya Santokhee[1], Girish Bekaroo[1]

[1]School of Science and Technology, Middlesex University Mauritius
Flic-en-Flac, Mauritius
{nishant.teckchandani@gmail.com; a.santokhee@mdx.ac.mu; g.bekaroo@mdx.ac.mu}

Abstract - A chatbot is a software that is able to autonomously communicate with a human being through text and due to its usefulness, an increasing number of businesses are implementing such tools in order to provide timely communication to their clients. In the past, whilst literature has focused on implementing innovative chatbots and the evaluation of such tools, limited studies have been done to critically comparing such conversational systems. In order to address this gap, this study critically compares the Artificial Intelligence Mark-up Language (AIML), and Sequence-to-Sequence models for building chatbots. In this endeavor, two chatbots were developed to implement each model and were evaluated using a mixture of glass box and black box evaluation, based on 3 metrics, namely, user's satisfaction, the information retrieval rate, and the task completion rate of each chatbot. Results showed that the AIML chatbot ensured better user satisfaction, and task completion rate, while the Sequence-to-Sequence model had better information retrieval rate.

Keywords: Sequence-to-Sequence Model, AIML, Chatbot, Conversational Agents, Artificial Intelligence, Recurrent Neural Network.

1 Introduction

As businesses need to ensure effective and timely communication to customers, smarter solutions in the form of chatbots have emerged. Chatbots are computer programs that interact, through conversations, with a user via a textual medium [1]. These programs receive information, process it, and send out a reply in a coherent manner. Instead of using mobile applications to find information, a chatbot can get these tasks done through a text query. Although these tools exist since the 1960s, many chatbots have been developed with different functions and methods over the years [2]. Due to their usefulness, more than $140 million has been invested in chatbots since 2010 and an increasing investment is expected to follow in the next few years [3].

Two main types of chatbots exist, namely, close-domain system and open-domain system [3]. The close-domain system chatbot functions based on a set of rules and responds to specific commands that were hard coded into the software. This category

of chatbot is also as smart as it is programmed to be and the scope of query answers is considered as limited. The first few chatbots like the Artificial Linguistic Internet Computer Entity (ALICE), ELIZA and ELIZABETH are key examples of close-domain systems [4]. On the other hand, open-domain system chatbot utilizes artificial intelligence, machine learning algorithms, and natural language processing methods to understand and process what a user types in. As such, it continuously gets smarter over time to respond to different questions and formats. Examples of chatbots implementing this approach include AliMe Chat [5] and MovieTriples [6]. In order to implement such conversational systems, different models are available. A popular close-domain based one includes the Artificial Intelligence Mark-up Language (AIML) and one open-domain based model is the Sequence-to-Sequence model. The AIML is an XML derivative and is used for developing natural language software agents. The AIML is used by ALICE, in order to form responses to questions and inputs and this approach was also found to be effective for the Mitsuku chatbot, which has won numerous awards in recent years [7]. On the other hand, in the Sequence-to-Sequence model, dialogs between end-users and agents can be regarded as a mapping of one sequence of words representing the request to another sequence of words that represents the response [8]. In this process, different learning techniques, including deep learning could be used to learn the mapping from sequences to sequences [9].

Although high amount of money is being invested on such technology as mentioned earlier, limited work has been undertaken so as to critically compare different methods used for constructing chatbots. As related work, a recent study provided a brief review of AIML based chatbots [10]. Even though this study provides some insightful information on different chatbots, it focuses only on AIML-based ones. Likewise, another study discussed the technologies and special features during development of conversational systems and chatbots [11]. In this study, limited comparison has been made between popular models, especially, sequence-to sequence based chatbots. As such, in order to address this gap, this study critically compares the AIML and Sequence-to-Sequence models for building chatbots.

2 Chatbot Implementation Models

As briefly explained earlier, two popular models to implement chatbots include the AIML and Sequence-to-Sequence models. Literature and theoretical background on both models are given as follows:

2.1 Artificial Intelligence Mark-up Language

The AIML model takes advantage of keywords in sentences, and generates a predetermined output. By verifying the keywords, and the ways it could be used, the chatbot would be able to generate an output that the user needs. AIML contains data objects consisting of two types of units called topics and categories [12]. The topic contains the name, attribute and a set of categories related to the topic. On the other hand, category is used to represent the knowledge gained through the input and

contains a template, which represents the chatbot response [13]. The AIML files consist of the components/tags described Table 1:

Table 1. Description of attributes.

Tag	Description
<aiml>	Defines the beginning and the end of the AIML file
<category>	Contains the unit of knowledge of the AIML file. Each category tag must contain a <pattern> and <template> tag
<pattern>	Represents the pattern that the user's input could potentially match to. The user's input will check each <pattern> tag of all the <category> tags, and choose the <pattern> it matches to. Also, this tag can contain wild card characters, such as _ or *.
<template>	It contains the response to the user. If the user's input matches with the <pattern> of a certain <category> tag, the <template> content will be sent back as a response.
<srai>	It is a multipurpose tag, which can redirect to other <category> tags and output the corresponding <template>. This tag works better when synonyms need to be defined, and for keyword detection.
<random>	This tag is able to generate a variety of responses at random for the same input.

AIML is completely case insensitive. Therefore, if the user asks "how are you" or "HOW ARE YOU", it will get directed to the same <category tag>. Additionally, punctuations like question marks, periods and commas do not need to be defined in the <pattern> tags. The "_" and "*" are wild card characters and are incredibly helpful in the AIML scripts. These wildcard characters represent any combination of words that may not have been defined in the AIML files.

2.2 Sequence-to-Sequence

The Sequence-to-sequence model makes use of Recurrent Neural Networks (RNN) to take in a user's input sentence, and generates an output sentence [9]. The RNNs are trained through a dataset of conversations, and its weight as well as bias values are adjusted to generate the necessary words for the output. After training of the model, each word gets processed through an encoder, and the next word is determined with the RNN according to the node that has the highest weight value through the decoder. As seen in Fig. 1, the input message runs through the encoder, and creates a thought vector. In the decoder, the first word is then calculated based on the input message. Each word is finally determined by the previous word in order to create a complete sentence for an output message.

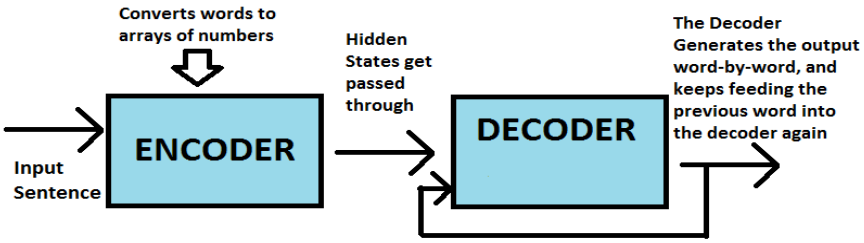

Fig. 1 - Example of the encoder-decoder architecture used by the chatbot

3 Methodology

In order to achieve the purpose of this paper, work was conducted in three main phases, starting with implementing two chatbot prototypes followed by the evaluation phase. In the first two phases, two prototypes were built, due to limited availability of existing chatbots that implement the AIML model and Sequence-to-sequence models, with same functional requirements. The two chatbots were developed using the Python programming language for the context of Middlesex University Mauritius in order to answer queries of prospective students on admissions, financial aspects, programme/courses, accommodation, and general other queries. This information was gathered from the university's website, in addition to documents provided by the same institution. Conversation datasets were created for the Sequence-to-sequence model through the FAQs given by the university. AIML and Tensorflow libraries were installed to implement the AIML and Recurrent Neural Networks within the chatbot software. Also, the chatbots ran on the windows command line, and user queries, along with the answer generated were saved into a SQLite database table. The three phases are further described as follows:

3.1 Implementation of the AIML Chatbot

The AIML chatbot contains an AIML file for each functional requirement (admissions, finance, etc.) and each AIML file consists of at least one category tag. This category tag would give the response to a main query that users could potentially ask. For example, the finance AIML file contains a category that detects an instance when the user asks about "tuition". The possible answers are then included in the template tags. Finally other category tags were created, for possible combination of words being used by users for the same "tuition" related queries. The srai tags were used in the template tags to recursively refer to the original tuition related category tag in the AIML file. New topics would be implemented into the chatbot with this process. In total, over 30+ main category tags were used and between 10 – 20 recursive category tags were used for each main category tag. A sample code snippet containing the tags is given in Fig. 2. In this code snippet, choices of output answers are given in the initial <li> tags. The latter category tags will be activated, if the user's

input query consists of the word “admissions”. If the program does not detect any keywords, it will give an output message containing the last category tag.

```
<category>
	<pattern>PROGRAMS</pattern>
	<template>
		<random>
			<li>Middlesex University, Mauritius offers many
programs. The Bachelor programs consist of BSc, and BA in Information
Technology, Computer Science, Psychology, Business Management, Accounting and
Finance, Public Relations, Media, Advertising, LLB Bachelor of Law, and LLB
with International Relations.</li>
			<li>The Middlesex University, Mauritius offers numerous
programs ranging from Psychology to Information Technology. Please refer to
http://www.middlesex.mu/courses for more information</li>
		</random>
	</template>
</category>
<category>
	<pattern>_ COURSE _ </pattern>
	<template>
		<srai>PROGRAMS</srai>
	</template>
</category>
<category>
	<pattern>COURSE _ </pattern>
	<template>
		<srai>PROGRAMS</srai>
	</template>
</category>
```

Fig. 2 – Example of AIML tags in the programmes/courses AIML file

3.2 Implementation of Sequence-to-Sequence Chatbot

In the second phase, the second chatbot was designed, developed and tested. This chatbot uses a Sequence-to-Sequence model consisting of Recurrent Neural Networks to help the chatbot learn [9]. The Tensorflow library was used to implement the RNNs in Python and the development phase was broken down into three steps, described as follows:

- *Step 1: Creating and transforming the dataset*

Firstly, a dataset consisting of questions, and the corresponding answers were saved into a text file. In the same file, question sentences started with a “Q:” notation whilst answer sentences started with the “A:” notation. The different topics of discussions were separated with a “===”. It was important for punctuation marks such as commas, periods and exclamation marks be separated by spaces for proper processing. The questions and answers were taken from the Middlesex University FAQ pages, or were self-generated depending on the information available on the

Middlesex University websites. As such, a dataset containing 1688 lines of conversation was used. An illustration of the dataset is illustrated in Fig. 3.

```
Q: Where do I go for University events ?
A: Check your MDX email for constant reminders of University
Events .
===
Q: What happens if I lose my ID card ?
A: You will have to contact the student office and request to
create a new one .
===
Q: What do I need to bring for my first few days at University ?
  A: The only things that you need to bring for your first few
days at the University are your documents for your ID check ,
notebooks , and stationary .
```

Fig. 3 - The Dataset

- *Step 2: Creating the Sequence-to-Sequence model with Encoder-Decoder Architecture*

The RNN encoder-decoder architecture consists of two recurrent neural networks. The encoder takes in a variable-length compilation of words, and maps it to a fixed length vector. This thought vector is then mapped to a variable length compilation of words that produces the output [14]. Since RNNs are only able to process numbers, each word in the training data has to be tokenized to an integer value so it can be processed correctly in the RNN. Once the integer values are determined through the RNN, these integers get de-tokenized and a final output message is generated.

- *Step 3: Training the model*

Once the model was created, the training script was run to the embedding weight values, encoder weight values, decoder weight values, and decoder embedding weight values. Amongst, the mean loss value represents how well the Sequence-to-Sequence is performing at a given time [14]. The higher the mean loss, the more mistakes the chatbot is likely to make, and vice versa. As the training went on, the mean loss value kept dropping according to the learning rate. Since the learning rate determines how much change occurs in the RNN weight and bias values, the learning rate had to be decreased after the mean loss reached certain values. When the mean loss is at a value of 8.1062, the learning rate started at 0.0008. After 60 epochs of training, the mean loss had decreased to 0.1744, with a learning rate of 0.000095. It is important to decrease the learning rate as the training goes on, as small changes to the RNN is needed to find the final gradient values of each node (tokenized word).

3.3 Evaluation

Once the chatbots were developed, the final stage was about conducting the evaluation to compare the effectiveness of the two models for implementing chatbots.

For this, the evaluation method had to be determined first where in the past, numerous methods have been used to evaluate the effectiveness of chatbots [15]. These include:

- *Turing Test*

The Turing test originated in the 1950s and is considered as one of the first methods for evaluating a chatbot [4]. It involves a human observer conversing with a chatbot and determining if the conversation was completed by a machine or a human [15]. This approach was however regarded as biased since it is based on human observation, where it was claimed that as long as a developer is able to create a system that will convince the human observer, the system will pass the Turing test [16].

- *Glass Box and Black Box*

Other methods used to evaluate conversation models include glass box and black box evaluation [17]. The glass box evaluation involves analyzing individual components of the chatbot, like each response to a user input sequence. Black box evaluation involves evaluating the system as a whole, such as giving a certain conversation, or the chatbot a certain score of effectiveness [4]. Black box evaluation is based on user satisfaction, acceptance of the system, time taken to achieve a task, and the accuracy of the information that the user received [18]. Utilization of both glass box and black box evaluation can be more effective due to the increased number of components graded [4]. An issue with this approach is that could it take time to get the right participants for the evaluation.

- *Algorithmic Evaluation Methods*

The final and most cost effective way to evaluate chatbots, is to use algorithmic methods. This includes using an algorithm to test the chatbot's performance based on a certain numeric value. Amongst, the Bilingual Evaluation Understudy (BLEU) was proposed to evaluate conversation models [19]. BLEU uses the dataset, for the purpose of training, as the reference of what a good quality response is [20]. It then calculates a score on each response by comparing the sequence given by the user to the good quality response discussed earlier [19]. An average score is calculated based on all the responses from the conversation, outputting from 0 to 1, where 1 is the most accurate, while 0 represents a wrong response. Other algorithmic evaluation methods include the ROUGE (takes into account groups of words when making the comparison with the dataset), and METEOR (works on word-to-word comparisons and includes synonyms) methods [21]. While such models are cheaper and quicker options over human evaluation like the glass box or black box, it was not considered to be fully effective [20].

Among the evaluation methods, a mixture of glass box and black box was chosen due to its effectiveness, as discussed above. As such, to proceed with the evaluation, a questionnaire was designed containing effectiveness, acceptance and user satisfaction related questions. Once the questionnaire was ready, a pilot study was conducted with 5 participants and feedback from the respondents helped to finalize the questionnaire and evaluation process. Then, participants were recruited from Middlesex University Mauritius and during the recruitment process, a brief on the study was given before seeking informed consent of the participants using relevant forms. In total, 45

participants agreed to participate in the study. Every participant was then asked to complete a set of tasks with each chatbot by communicating with the chatbots, through text to seek details on admissions, financial aspects, courses, etc. After communicating with the chatbots, the participants were asked to fill-in the previously designed questionnaire. The filled-in questionnaire was then collected and checked to ensure reliability and correctness. The whole process took around 1 hour per participant and all the conversations were saved into a database for individual response analysis. Each response was also evaluated with a precision and recall method [21]. Finally, data collected were statistically analyzed while focusing on key metrics, namely, user satisfaction, the information retrieval rate, and the task completion rate of each chatbot were computed.

4 Results and Discussions

In terms of task completion rate, the AIML Chabot was better as compared to the Sequence-to-Sequence model by a small margin of approximately 4%. The AIML chatbot was able to score better results when a user had inquired about admission, finance, programme, accommodation, and location related questions. However, with greetings and queries that were outside the scope of the use cases, the Sequence-to-Sequence chatbot had scored better. The results are displayed in Table 2.

Table 2. Task completion rate.

Task completion rate	AIML Chatbot	Sequence-to-Sequence Chatbot
Task 1 (Basic Greetings)	39/45 = 86.7 %	45/45 = 100 %
Task 2 (Admission Queries)	32/45 = 71.1 %	27/45 = 60 %
Task 3 (Tuition Queries)	34/45 = 75.6 %	25/45 = 55.6 %
Task 4 (Programmes/Courses Queries)	30/45 = 66.7 %	19/45 = 42.2 %
Task 5 (Accommodation Queries)	34/45 = 75.6 %	23/45 = 51.1 %
Task 6 (Location Queries)	30/45 = 66.7 %	34/45 = 75.6 %
Task 7 (User's Choice Queries)	15/45 = 33.3 %	28/45 = 62.2 %
Overall task completion rate	**214/315= 67.9 %**	**201/315 = 63.8 %**

The second part of the evaluation included analyzing individual responses of the chatbots. Responses are categorized into 4 different types as shown in Table 3.

Table 3. Response types

Response type	Description
True Positive	Chatbot responded correctly to user's correct input message
False Positive	Chatbot responded incorrectly to user's correct input message
False Negative	Chatbot responds incorrectly to user's incorrect input message
True Negative	Chatbot responds correctly to user's incorrect input message

Based on the responses given in Table 3, Precision, Recall, and F-Measure were calculated using the following formulae:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

$$F\text{-}measure\ (\beta) = \frac{1}{\frac{\beta}{precision} + \frac{1-\beta}{Recall}} \quad (3)$$

Results showed that the Sequence-to-Sequence chatbot had slightly outperformed the AIML chatbot based on individual response of questions (information retrieval). The higher precision, recall, simple accuracy, and F-measure in Table 4 show that the Sequence-to-Sequence model was able to answer questions more effectively than the AIML model.

Table 4. Table of Precision, Recall, Simple Accuracy, and F-Measure Values

Information Retrieval (Per Response)	AIML Chatbot	Sequence-to-Sequence Chatbot
TP (True Positive)	292	304
FP (False Positive)	217	196
FN (False Negative)	41	38
TN (True Negative)	36	11
Precision	0.574	0.608
Recall	0.877	0.889
F-Measure (β = 0.8 to favor precision over recall)	0.617	0.649

Overall, the AIML chatbot had scored significantly better than the Sequence-to-Sequence model, in terms of user satisfaction. This was principally because the AIML chatbot giving clear cut answers that always made sense according to the users. The Sequence-to-Sequence model had a problem with grammar, and repetition of words, which made it harder for participants to enjoy the experience with this model. The user satisfaction rating results are given in Table 5.

Table 5. Table of User satisfaction values.

Task completion rate	AIML Chatbot (out of 10)	Sequence-to-Sequence Chatbot (out of 10)
How satisfied are you with the chatbot?	7.29	5.24
How human-like was the chatbot?	5.67	5.64
How easy was it to converse with the chatbot?	7.24	5.69
Overall score	6.73	5.52

5 Concluding Remarks

This study aimed to compare two chatbot implementation methods, namely, AIML and Sequence-to-Sequence, through the implementation of two prototypes for the context of Middlesex University Mauritius. The two chatbots were evaluated using a mixture of glass box and black box evaluation, based on 3 metrics, namely, user's satisfaction, the information retrieval rate, and the task completion rate of each chatbot. Results showed that the AIML chatbot outperformed the Sequence-to-Sequence one on overall, especially in terms of user satisfaction, and task completion rate. On the other hand, the information retrieval rate of the chatbot implementing the Sequence-to-Sequence model was found to be better. Although requirements of chatbots vary from business to business, for the context of Middlesex University Mauritius, the AIML model was found as a better option. This is because there is no absolute need for human-like conversation, and fixed responses would be able to answer queries effectively. However, if a more intelligent system would be needed, then the Sequence-to-Sequence model would fit better as this model showed to better cope with queries that were outside the scope of use cases.

As future works, contextual meaning to sentences would be implemented such that if a user is chatting with the chatbot, the chatbot would consider demographic details of participants to answering queries. Furthermore, the experiment could be conducted with a larger number of users and in different organizational contexts.

Ethics Compliance

All procedures involving the participants in this study were in accordance with the Ethics Framework of Middlesex University, which has been set out by the University Ethics Committee. Also, during the data collection phase, informed consent was obtained from all individual participants who participated in the study.

References

[1] M. Levy, "Technologies in use for second language learning," *The Modern Language Journal,* vol. 93, no. 1, pp. 769-782, 2009.

[2] D. Madhu, C. Jain, E. Sebastain, S. Shaji and A. Ajayakumar, "A novel approach for medical assistance using trained chatbot," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2017.

[3] Spartans AI Bots & Virtual Reality Development, "Spartans AI Bots & Virtual Reality Development," 2017. [Online]. Available: https://www.spartans.tech/single-post/Chat-Bots-Explained. [Accessed 25 Oct 2017].

[4] B. Shawar and E. Atwell, "Different measurements metrics to evaluate a chatbot system," in *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies*, 2007.

[5] M. Qiu, F. Li, S. Wang, X. Gao, Y. Chen, W. Zhao, H. Chen, J. Huang and W. Chu, "Alime chat: A sequence to sequence and rerank based chatbot engine," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[6] I. Serban, A. Sordoni, Y. Bengio, A. Courville and J. Pineau, "Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models," in *AAAI*, 2016.

[7] Mitsuku, "Mitsuku Chatbot," 2018. [Online]. Available: http://www.mitsuku.com/. [Accessed 11 Nov 2017].

[8] A. Xu, Z. Liu, Y. Guo, V. Sinha and R. Akkiraju, "A new chatbot for customer service on social media," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017.

[9] I. Sutskever, O. Vinyals and Q. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014.

[10] M. Satu and M. Parvez, "Review of integrated applications with aiml based chatbot," in *2015 1st International Conference on Computer and Information Engineering (ICCIE)*, 2015.

[11] J. Masche and N. Le, "A Review of Technologies for Conversational Systems," in *International Conference on Computer Science, Applied Mathematics and Applications* , 2017.

[12] Tutorialspoint, "AIML Tutorial," 2018. [Online]. Available: https://www.tutorialspoint.com/aiml/index.htm. [Accessed 21 Jan 2018].

[13] B. Shawar and E. Atwell, "A comparison between ALICE and Elizabeth chatbot systems," University of Leeds, School of Computing research report 2002, Leeds, 2002.

[14] M. Luong and E. Zhao, "Neural Machine Translation (seq2seq) Tutorial," 2017, 2018. [Online]. Available: https://github.com/tensorflow/nmt. [Accessed 22 Jan 2018].

[15] N. Radziwill and M. Benton, "Evaluating Quality of Chatbots and Intelligent Conversational Agents," *Software Quality Professional,* vol. 19, no. 3, pp. 25-36, 2017.

[16] A. Turing, "Computing machinery and intelligence," *Mind,* vol. 59, no. 236, pp. 433-460, 1950.

[17] S. Shieber, "Lessons from a restricted Turing test," *Communication,* vol. 37, no. 6, pp. 70-78, 1994.

[18] P. Price, "Evaluation of spoken language systems: The ATIS domain," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania*, 1990.

[19] E. Maier, M. Mast and S. LuperFoy, "Dialogue Processing in Spoken Language Systems," in *DPSLS: Workshop on Dialogue Processing in Spoken Language Systems*, 1996 .

[20] K. Papineni, S. Roukos, T. Ward and W. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002.

[21] C. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin and J. Pineau, "How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[22] X. Zhang, S. He, X. Song, P. Wei, S. Jiang, Q. Ye, J. Jiao and R. Lau, "Keyword-driven image captioning via Context-dependent Bilateral LSTM," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017.