Can rule-based chatbots outperform Neural models without pre-training in Small Data Situations?: A Preliminary Comparison of AIML and Seq2Seq

Md Mabrur Husan Dihyat, Julian Hough[1]
[1] Cognitive Science Group
School of Electronic Engineering and Computer Science
Queen Mary University of London, UK
{m.dihyat@se17.qmul.ac.uk, j.hough@qmul.ac.uk}

Abstract

This paper addresses the limitations of rule-based and end-to-end neural chatbots with little training data. We compare an AIML-based chatbot and a Seq2Seq chatbot developed on a small, high quality dataset of 300 turns of IT Service queries and responses. The chatbots were then evaluated using ROUGE automated evaluation metrics as well as task completion rate by human judges. While the Seq2Seq model could generalize quite well to new inputs, the rule-based AIML chatbot was found to ensure better task completion rates as well as higher ROUGE scores. The findings suggest that rule-based chatbots are still a useful tool with little resource available, though more needs to be done to confirm their limitations.

1 Introduction

While much previous research has concentrated on developing and testing advanced chatbot systems within rule-based and deep learning based paradigms, there have been few studies that objectively compare these two types of system using the same data. Moreover, with the advance of neural models with large parameter spaces, it is not clear how these models scale downwards when big data is not available.

The purpose of this paper is a preliminary investigation into building, evaluating and comparing two common examples of these two chatbot paradigms in a realistic real-word application and resource scenario: an IT services chatbot at Queen Mary University of London which responds to queries relating to computing issues people may have at the institution, where only a small amount of example dialogue data is available to develop on, without the use of pre-training for a neural system.

2 Methodology

Data A corpus of 400 turns, or 200 query-response pairs, was gathered from the Queen Mary University of London IT Services chat transcripts collected between 2018-2020 between students and human support staff. While creating and cleaning the responses, it was found that 49% of the user queries had detailed relevant answers in the FAQ pages of the Queen Mary IT Services webpages. Thus, where possible, the FAQ answer was used as the response, replacing the original human agent response. 150 query-response pairs were used for the development of both the rule-based and neural chatbots, with the remaining 50 heldout for testing.

Rule-based chatbot We use Artificial Intelligence Markup Language (**AIML**) (Wallace, 2003), an XML-based markup language, to create our rule-based chatbot. AIML uses pattern matching techniques to formulate answers from queries. Each AIML file consists of <category> tags which are the basic unit of knowledge in AIML, containing an input question, an output answer and an optional text. Inside each category, the question is stored in the <pattern> tag while the corresponding answer to the question is stored in the <template> tag, which is the text the chatbot will reply with. The pattern language consists of words, spaces and wildcard symbols such as: "_" and "*". Wildcards are used to replace strings in AIML (words or sentences). The wildcard "_" is given the highest priority, which means that categories containing this wildcard are analysed first (Mikic et al., 2009).

Based on the user intents found while analysing the dataset, a total of ten AIML files were created where each file addresses a specific issue. Out of these ten files, eight of them were created from scratch for the domain (e.g. *Login Issues* and others in Fig. 2) while two were imported from the A.L.I.C.E chatbot system (Wallace, 2009), sourced

Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue, September 20–22, 2021, Potsdam / The Internet.

Task	ROUGE-1			ROUGE-L		
	Average precision score	Average recall score	Average f-measure	Average precision score	Average recall score	Average f-measure
Greetings	0.62	0.69	0.68	0.62	0.69	0.68
Login Issues	0.50	0.48	0.48	0.49	0.48	0.48
MYHR Issues	0.49	0.53	0.47	0.46	0.49	0.45
Password Issues	0.61	0.54	0.55	0.59	0.53	0.54
Password Requirement	0.60	0.50	0.54	0.57	0.50	0.51
Agresso Issues	0.67	0.64	0.65	0.66	0.62	0.64
Address	0.54	0.54	0.548	0.54	0.54	0.548

Task	ROUGE-1			ROUGE-L		
	Average precision score	Average recall score	Average f-measure	Average precision score	Average recall score	Average f-measure
Greetings	0.79	0.73	0.75	0.79	0.73	0.75
Login Issues	0.33	0.23	0.26	0.32	0.22	0.25
MYHR Issues	0.47	0.35	0.39	0.30	0.19	0.23
Password Issues	0.21	0.27	0.23	0.15	0.18	0.17
Password Requirement	0.64	0.52	0.47	0.60	0.54	0.53
Agresso Issues	0.89	0.87	0.88	0.87	0.86	0.86
Address	0.554	0.45	0.49	0.554	0.45	0.49

Figure 1: ROUGE scores of the AIML chatbot (left) and Seq2Seq chatbot (right)

from the website *Kaggle* (Bhatia, 2020). There were on average 12 categories per file, each one designed to closely match and cover all the relevant training set queries, and the response templates for IT issues were the strings from the corresponding responses in the cleaned dataset.

Due to its popularity and relative simplicity, the `Python-aiml` (Stratton, 2003) library was used to build the AIML engine for the chatbot.

Seq2Seq neural chatbot For our neural model, we train an LSTM (long short-term memory) Sequence-to-Sequence (**Seq2Seq**) (Sutskever et al., 2014) model on the 150 query-response pairs in the training data, with the responses identical to the AIML templates. The chatbot was developed in Python using the Tensorflow and Keras libraries (Panchal, 2020).

The Seq2Seq model has one input layer x, a vector of length 40 (the maximum query length) and a decoder input layer, y, a vector of length 141 (the maximum response length). The encoder model has three more layers after the input layers: An Embedding layer (of size 200), an LSTM layer, and the Dense layer of dimension (141, 535), where 535 is the vocabulary size. The Seq2Seq model has a total of 963,135 trainable parameters.

Evaluation We evaluate the success of the responses to the 50 test set queries automatically using a **ROUGE-1** and **ROUGE-L** (Lin, 2004) comparison to the ground truth response (precision, recall and F-1 measure) and also measure **human-judged task completion success**. For the task completion evaluation, both authors judged separately whether the outputs for the models constituted successful outputs or not, based on their knowledge of the IT problem in question, and the authors agreed on the judgement of success on 97 of the 100 responses from the two systems.

TASKS	AIML CHATBOT	SEQ2SEQ CHATBOT
1. GREETINGS	5/7 = 71.4%	5/7 = 71.4%
2. LOGIN ISSUES	3/7 = 42.8%	1/7 = 14.3%
3. MYHR ISSUES	5/11 = 45.5%	3/11 = 27.3%
4. PASSWORD ISSUES	5/8 = 62.5%	1/8 = 12.5%
5. PASSWORD REQUIREMENT	4/6 = 66.7%	4/6 = 66.7%
6. AGRESSO ISSUES	4/6 = 66.7%	5/6 = 83.3%
7. ADDRESS	3/5 = 60%	2/5 = 40%
OVERALL	29/50 = 58%	21/50 = 42%

Figure 2: Task completion rates.

3 Results and Discussion

Using the first author's judgements as the ground truth, as can be seen in Fig. 2, overall the AIML chatbot was found to be approximately 16% more proficient in handling user queries than the Seq2Seq model (58% vs 42% task completion). The AIML chatbot particularly outperformed the Seq2Seq model on *Login Issues*, *Password Issues*, *MyHR Issues*, and *Address* queries.

In terms of automatic metrics, the mean ROUGE-1 and ROUGE-L scores for both chatbots' responses is shown in Fig. 1. As can be seen, in most problem types the AIML bot outperforms the Seq2Seq model across the metrics in each category, with two exceptions (*Greetings* and *Agresso Issues*). The Seq2Seq model shows some generalization with novel input sequences which are similar, but not identical, to those in its training data in less flexible categories: in more varied input categories, AIML is more robust.

This preliminary investigation suggest that with a small amount of data, both in terms of task success and output quality, it is still safer to use a rule-based chatbot with AIML than relying on generalization from an end-to-end neural model. No pre-training of the Seq2Seq model was employed, so future work will involve testing its effect on performance systematically from this baseline.

References

Ruchi Bhatia. 2020. AIML files from Kaggle.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Fernando A Mikic, Juan C Burguillo, Martín Llamas, Daniel A Rodríguez, and Eduardo Rodríguez. 2009. Charlie: An aiml-based chatterbot which works as an interface among ines and humans. In *2009 EAEEIE Annual Conference*, pages 1–6. IEEE.

Shubham Panchal. 2020. Creating a chatbot from scratch using keras and tensorflow.

Cort Stratton. 2003. PyAIML – The Python AIML Interpreter.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NIPS*, Montreal, CA.

Richard Wallace. 2003. The elements of AIML style. *Alice AI Foundation*, 139.

Richard Wallace. 2009. The anatomy of ALICE. In *Parsing the Turing Test*, pages 181–210. Springer.