ECHO: A Tool for Empirical Evaluation Cloud Chatbots

Abdur Rahim Mohammad Forkan*, Prem Prakash Jayaraman*, Yong-Bin Kang*, Ahsan Morshed†
*Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, Australia
†School of Engineering and Technology, Central Queensland University, Melbourne, Australia
{fforkan, pjayaraman, ykang}@swin.edu.au, a.morshed@cqu.edu

***Abstract*—A chatbot is a software that interacts with humans by conducting conversations via textual or auditory methods. Chatbots have recently been used for plethora of applications including travel, medical, education, retail etc. Several cloud-based platforms (e.g. IBM, Amazon, Google, Microsoft) are available for developing and deploying chatbots. However, there is a lack of an evaluation methodology and a tool for evaluating chatbots comprehensively. Current approaches for comparing cloud-based chatbots are manual and rely on expert's judgement. In this short paper, we propose, devise, implement and demonstrate a tool namely ECHO for empirical evaluation of cloud-based chatbots. ECHO is capable of automatically evaluating multiple cloud-based chatbots and report the outcomes of the comparative evaluation. We validate the efficacy of ECHO by conducting comparative evaluation of 3 popular cloud-based chatbots in 2 different question-answering application scenarios with 3 levels of complexities.**

***Index Terms*—Chatbots, NLP, Cloud platform, Conversation, Performance Evaluation**

I. Introduction

The interest for Chatbots is growing exponentially with Gartner forecasting that by 2020, over 85% of customer interactions will be handled by chatbots [1]. Chatbots have proved in boosting operational efficiency and bring cost savings to businesses. Chatbots are machine agent (virtual assistants) that interact with humans by conducting conversations via textual or auditory methods. Chatbots underpinned by Artificial Intelligence (AI) learn interests, preferences and the context of the human by interacting with them and provide meaningful responses to their request and in most cases even making recommendations. The experience and fluidity of chatbots increase with more training data, as evident by the recent advancements in AI and natural language processing (NLP) [2]. Several providers such as IBM, Amazon, Google, Microsoft offer cloud-based solutions (software-as-a-service) for developing and deploying chatbots [3]. They offer APIs for conversational interfaces that has made the development and use of a chatbot to serve specific business needs easier.

With the emergence of cloud-based chatbot platforms and exponential increase in demand for AI-driven chatbots to boost businesses, it is a significant challenge to empirically evaluate, compare and select the most suitable cloud-based chatbots platform that can serve business needs in terms of user satisfactions, effectiveness, achievable goals and efficiency [4]. Such a tool will allow chatbot-based application developers to easily compare and evaluate the performance of various cloud-based chatbot platforms before developing their solution. Chatbots are specifically trained for specialised topics depending on the application domain under consideration, such as healthcare, travel booking and education. Existing approaches to evaluate chatbots rely on human experts in the target domain [5] manually evaluating the outcomes produced by the cloud-based chatbot platform. Such evaluations can be in-accurate, time-consuming and labour-intensive.

In this paper we propose, devise, implement and demonstrate a tool namely ECHO for empirical evaluation of cloud-based chatbots. ECHO itself is a cloud-based application and can readily integrate with any cloud-based chatbot platform (e.g. Amazon, Google, IBM). ECHO incorporates a novel evaluation methodology that provides the foundations for quantitative evaluation of cloud-based chatbots. In particular, this paper makes the following contributions:

- Propose a systematic architecture for integrating and evaluating multiple cloud-based chatbots under various conversational domains.
- Propose an evaluation framework that provides a common ground to compare the outcomes produced by multiple cloud chatbots under various conversational domains.
- Implementation and demonstration of ECHO by integrating 3 popular chatbots (Lex, DialogFlow, Watson).
- Experimental evaluation to validate the efficacy of ECHO in 2 conversation domains (medical recommendation and flight booking), under 3 level complexities (basic, medium and complex).

II. Background and Related Works

Using AI platforms offered by cloud providers and open-source technologies, one can build interactive chatbots. Some popular examples include Google Dialogflow [6], Amazon Lex [7], IBM Watson [8], Microsoft Bot framework [9], open-source ChatterBot [10] and Rasa [11]. A standard for empirical evaluation of cloud chatbots has not been well-established and some cases can be outdated as the technology evolves. However, a proper evaluation is vital for a business organisation when a decision needs to be made to pick from a range of available chatbots that will suit the business objectives [12]. There is not much previous work to compare multiple cloud-based chatbots. Research on chatbot evaluation mainly being adapted from NLP such as content evaluation or based on user satisfaction and goal [4].

978-1-7281-6095-5/20/$31.00 ©2020 IEEE
DOI 10.1109/CCGrid49817.2020.00-26

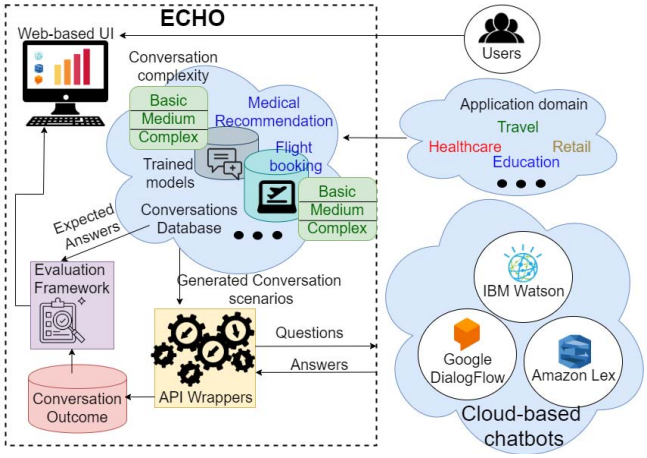

Fig. 1: The systematic architecture of ECHO

ChatEval, a web-based hub for researchers is proposed in [13], where people can share and compare their dialog systems with baselines and prior works. ChatEval used evaluation metrics from NLP such as Lexical diversity, Average cosine-similarity, Sentence average and Response perplexity. Different metrics to evaluate chatbots are proposed in [14] that used the efficiency of 4 sample dialogues in terms of atomic match, first word match, most significant match, and no match. The aim was to measure the efficiency of the adopted learning mechanisms to see if they increase the ability to find answers to general user input. These works are useful to compare a newly developed chatbot with some baselines, however, do not provide a simple tooling support to evaluate and compare performance of multiple cloud-based chatbots.

III. ECHO: A Tool for Evaluating Cloud Chatbots

Figure 1 shows a systematic architecture of ECHO. ECHO includes a web-based UI component to enable interactions with users wanting to compare and evaluate the cloud-based chatbots. The conversation database components contains conversations scenario stored as a sequence of questions to be asked and expected answers. Depending on the type of the conversation, the evaluation of the chatbot can be done in different complexity levels (e.g. basic, medium, complex). The API wrapper component is responsible for providing an interface to different cloud-based chatbots via API call. The responses provided by all chatbots are stored in a response database. Upon execution of a conversation scenario by a cloud-based chatbot, the responses are compared against expected answers to compute the evaluation results. The results are displayed using the web-based UI.

A. Conversation Database

The conversation database stores conversation scenarios pertaining to a target application domain (e.g. medical, travel, education). In chatbot conversation an *Intent* is defined as a user's intention to know about something when a conversation begins. This is the point of mapping between the question and user's expectation from the system to respond. Intents are defined and developed as part of chatbot integration so it can recognise the domain and the type of questions asked by the user.

ECHO allows to define 3 complexity levels (which is extensible in future) ranked as basic, medium, complex. The higher the level is the more comprehensible the chatbots need to be. In the basic level, the conversation is usually straightforward and short. At a higher level, the conversation is longer with more follow up questions. In a basic conversation the user provides a detailed information so the chatbot does not need to ask too many follow up questions. Close-ended questions that require "yes" or "no" answer are examples of such basic conversation. For example "I would like to book one flight ticket from Melbourne to Sydney at 9pm in economy class". This detailed statement from the user lets the chatbot to provide some flight ticket alternatives along with their prices and dates so the user can pick a suitable one. A complex conversation is more like a mesh flow instead of a simple hierarchical flow, hence can jump to multiple directions. When a user is unsure about the need, it requires more back and forth questions during conversation with the chatbot. Open-ended questions and statements that requires more follow up also make a complex conversation.

B. Evaluation Framework

ECHO implements a novel evaluations framework that incorporates the following 7 metrics:

- **Average Response time** (τ): The response time (rt) is the difference between the time the user sends a question and the chatbot returns an answer. The average response time is calculated as, $\tau = \frac{\sum_{i=1}^{m} rt_i}{m}$, where m is the number of questions sent by the user in a conversation and rt_i is the response time to answer i-th question.
- **Fallback Rate** (ϑ): This measure refers to the number of times the chatbot failed to understand the user's question in a conversation. Chatbots are expected to fail as sometime they are confused by the unexpected messages from users and reply with fallback messages. For example, *Question: Can you help me to find a nearest hospital?. Answer: Sorry I can not understand your question, can you paraphrase it*. A chatbot with high fall back rate is required to be re-trained periodically with new data to improve its performance. Hence, the fallback rate is an important metric when evaluating the performance of a chatbot. The fallback rate is calculated as, $\vartheta = \frac{N_f}{N_m}$, where N_f is the number of time the chatbot had to fallback and N_m is the total number of messages the user asks in a conversation.
- **Comprehensive Rate** (γ): The comprehensive capabilities of chatbots are used for measuring the ability to correct errors in user's inputs. This is used to ensure an error-free experience for the users. When a user makes a spelling mistake in a sentence, the chatbot should detect such errors to quickly understand the statement the user is providing. For example, ***Question**: I woud like to buy*

a ticket from Tullamarine arport to Sydney. ***Answer****: Yes sure, which date you want to travel from Melbourne to Sydney*. The comprehensive rate calculated as, $\gamma = \frac{N_c}{N_e}$, where N_e is the number of user messages that contain grammatical errors, typos and sentence composition mistakes and N_c is the number of times the chatbot returns a correct answer for those messages.

- **Accuracy (A), Precision (P), Recall (R) and F1-score (F1)**: These metrics are obtained from true positive (tp), false positive (fp), true negative (tn) and false negative (fn) values when the chatbot responses are compared against a trained model. $\mathrm{A} = \frac{\mathrm{tp+tn}}{\mathrm{tp+fp+tn+fn}}$, $\mathrm{P} = \frac{\mathrm{tp}}{\mathrm{tp+fp}}$, $\mathrm{R} = \frac{\mathrm{tp}}{\mathrm{tp+fn}}$ and $\mathrm{F1} = \frac{\mathrm{2PR}}{\mathrm{P+R}}$.
  - tp: for a given question relating to intent I, the user gets a correct answer from I.
  - tn: for a given question relating to I, the user gets an answer from intent I' or Fallback intent I^F
  - fp: for a given unrelated question, the user gets correct answer from I^F
  - fn: for a given unrelated question, the user gets wrong answer from I.

  P identifies the frequency of correct answers when the prediction is intent I that is, the number of correct answers in predictions for I. R identifies the frequency of detecting I, out of all examples pertaining to I in reality, that is, out of all the examples in I, how many are detected. $F1$ calculates the harmonic mean of precision and recall. It helps to identify the global performance of prediction with respect to I. A refers to the number of correct predictions made by the chatbot for I.

C. User Interface (UI) for Visualisation

ECHO incorporates a web-based UI to allow users to evaluate multiple chatbots using selected conversation domain and complexity (see Figure 2). A user van view evaluation result via a graphical representation.

IV. Implementation of ECHO

ECHO is developed using multiple software technologies. The UI is a web-based application. APIs are implemented using python 3 with Flask framework. In our implementation 2 conversation domains are selected and 3 complexity level question-answer scenario are used. Relevant intents are developed for possible users questions. These are described as follows:

- **Medical Recommendation (MR)**: Publicly available MR data are collected from health-related websites [15]. We develop multiple conversation scenarios - one for various diseases, their symptoms, treatment medication, recommendations, specialists, causes and hospital addresses and another for different type of injuries, injury position and treatment recommendations. Some examples of 3 complexity levels are
  - **Basic**: The user provides the disease and asks for recommendations for doctor/specialist.
  - **Medium**: The user provides symptoms, the chatbot gives possible disease; the user asks for medication recommendations, the bot suggests medications; the user ask how often they should take the medications, the bot suggests the dosages.
  - **Complex**: The user provides symptoms, the bot gives possible diseases; the user asks for the cause of the disease, the bot suggest the reason; the user asks for types of medications, the bot suggests medications; the user ask how often they should take the medications, the bot suggests the dosages; the user asks how to prevent the disease, the bot suggests the prevention.
- **Flight booking (FB)**: To construct a scenario for flight booking we create conversations using information from multiple sources such as Kaggle [16] and air travel data from 3 Australian airlines (Qantas, Tigerair and Jetstar). We collect flight details (e.g. departing location, destination, flight dates and flight time) between Melbourne, Sydney, Perth, Adelaide and Brisbane from 10 to 17 October 2019. Some examples of 3 complexity levels are
  - **Basic**: The user asks question about entering and departing time, destination airports, and travelling date. The bot suggests accordingly.
  - **Medium**: The user provides information (e.g. date, time, departure, destination) to purchase a ticket and the bot provides answers accordingly.
  - **Complex** In addition to medium conversation scenario the user also provide information such as number of users travelling, flight type, preferable time and prices. The bot provides relevant responses.

Fig. 2: Conversation domain and complexity level selection

V. Evaluation of ECHO

As discussed in Section IV, we demonstrate and evaluate ECHO using 7 evaluation metrics in 3 complexity levels for 2 conversation domains. We have used 3 cloud-based chatbots (Lex, Diaglogflow, Watson). A user scenario of the UI comparing Lex and Dialogflow is presented in Figure 3. Upon completion of automatic evaluation the user gets option to view the evaluation results using charts (Figure 4) that summarises the various metrics discussed in section III-B. Table I presents the comparative results produced by ECHO when comparing 3 cloud-based chatbots using the 7 metrics for the 2 conversation domains (MR and FB), It can be noted that Lex has the best response time compared to others in both scenarios. Since, τ (in milliseconds) is averaged based on the number of questions in the conversation they are mostly

TABLE I: Evaluation result for 3 cloud-based chatbots for 2 conversation domains and three conversation complexity levels

Chatbot	Level	τ (ms)		ϑ (%)		γ (%)		A (%)		P (%)		R (%)		$F1$ (%)	
		MR	FB	MR	FB	MR	FB	MR	FB	MR	FB	MR	FB	MR	FB
Amazon Lex	Basic	93.6	92.3	0	0	40	100	100	100	84	57	90	70	91	73
	Medium	97	95.6	0	0	40	100	94	92	94	81	93	76	94	86
	Complex	94.9	96.1	0	0	20	25	100	90	85	56	93	74	92	69
Google DialogFlow	Basic	467	410	3	0	40	100	89	88	94	78	90	70	91	82
	Medium	400	438	14	3	0	100	78	91	100	80	93	76	88	85
	Complex	433	463	19	5	20	75	83	76	100	84	93	74	91	80
IBM Watson	Basic	263	200	3.7	0	60	50	92	95	96	96	90	92	94	96
	Medium	230	212	3.5	0	80	75	88	78	92	80	83	90	90	79
	Complex	220	232	18	5	60	75	68	73	100	78	70	89	81	75

Fig. 3: A screenshot of ECHO web-based UI

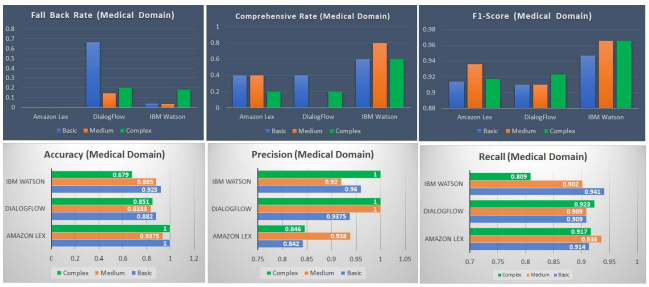

Fig. 4: Graphical view of evaluation result

similar across all 3 complexity levels. Lex also has the best fallback rate. Watson showed better comprehensive rate for MR data, specially for complex conversation, however, did not perform well for complex FB conversation. Lex ranks top in accuracy, however, it produced poor precision than others for both domains. Dialogflow and Watson has higher F1-Score for FB data than Lex. In summary Lex has better consistency for MR data whereas Watson performed well for FB data.

The results presented here demonstrates ECHO efficacy in enabling easy comparison and evaluation of cloud-based chatbots. It is to be noted that the current literature does not provide a common evaluations framework that enables comparison of results produced by multiple chatbots, a gap addressed by ECHO.

VI. Conclusion

In this paper, we presented, ECHO, a tool to evaluate and compare cloud-based chaotbots using various conversation domains. We proposed a systematic architecture and a novel evaluations framework that includes 7 key metrics that provides a common ground for comparing the results produced by chatbots. We demonstrated ECHO using 3 cloud-based chatbots, 2 conversational domains each including 3 levels of conversational complexity. We compared the outcomes produced by the chatbot using the proposed evaluations framework enabling users to easily compare and decide the right chatbot platform for their application domain. The ongoing work is to integrate more chatbots including and further enhance and refine the evaluation metrics.

VII. Acknowledgement

We acknowledge Swinburne computer science students Kai Leung Lui, Nguyen Tuan Anh Le, Chung Hoong Tsau, Wiyan Septio for implementing the ECHO tool.

References

[1] "The rise of automated customer care," https://www.promero.com/archive-press-release/gartner-artificial-intelligent-bots-oracle-bots/, 2018.
[2] M. Mnasri, "Recent advances in conversational nlp: Towards the standardization of chatbot building," *arXiv preprint arXiv:1903.09025*, 2019.
[3] K. Kuligowska, "Commercial chatbot: Performance evaluation, usability metrics and quality standards of embodied conversational agents," *Professionals Center for Business Research*, vol. 2, 2015.
[4] W. Maroengsit, T. Piyakulpinyo, K. Phonyiam, S. Pongnumkul, P. Chaovalit, and T. Theeramunkong, "A survey on evaluation methods for chatbots," in *Proceedings of the 2019 7th International Conference on Information and Education Technology*. ACM, 2019, pp. 111–119.
[5] M. Qiu, F.-L. 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, pp. 498–503.
[6] "Google dialogflow," https://dialogflow.com/, 2020.
[7] "Amazon lex," https://aws.amazon.com/lex/, 2020.
[8] "Ibm watson," https://www.ibm.com/cloud/watson-assistant/, 2020.
[9] "Microsoft bot framework," https://dev.botframework.com/, 2020.
[10] "Chatterbot," https://chatterbot.readthedocs.io/en/stable/, 2020.
[11] "Rasa," https://rasa.com/docs/rasa/, 2020.
[12] N. M. Radziwill and M. C. Benton, "Evaluating quality of chatbots and intelligent conversational agents," *arXiv preprint arXiv:1704.04579*, 2017.
[13] J. Sedoc, D. Ippolito, A. Kirubarajan, J. Thirani, L. Ungar, and C. Callison-Burch, "ChatEval: A tool for chatbot evaluation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demo)*, 2019, pp. 60–65.
[14] B. A. 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*. Association for Computational Linguistics, 2007, pp. 89–96.
[15] "Webmd," https://www.webmd.com/, 2020.
[16] "Kaggle," https://www.kaggle.com/, 2020.