Enabling a Massive Data Collection for Hotel Receptionist Chatbot Using a Crowdsourcing Information System

Preprint · August 2020

DOI: 10.13140/RG.2.2.10370.91843

CITATIONS
0

READS
7,725

7 authors, including:

Syafira Nurkafianti
Binus University
2 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Kristianus Oktriono
ASIA University Taiwan
48 PUBLICATIONS **80** CITATIONS
SEE PROFILE

Chandra Kurniawan Wiharja
Binus University
18 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Tjeng Wawan Cenggoro
Binus University
136 PUBLICATIONS **1,349** CITATIONS
SEE PROFILE

Enabling a Massive Data Collection for Hotel Receptionist Chatbot Using a Crowdsourcing Information System

Reval Levannoza*
Computer Science Department,
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
reval.levannoza@binus.ac.id

Rizky Fauzi Latif*
Computer Science Department,
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
rizky.latif@binus.ac.id

Syafira Indah Nurkafianti*
Computer Science Department,
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
syafira.nurkafianti@binus.ac.id

Kristianus Oktriono
Language Center
Tourism Destination,
Faculty of Humanities
Bina Nusantara University
Jakarta, Indonesia, 11480
koktriono@binus.edu

Devina
Language Center
Industrial Engineering,
Faculty of Humanities,
Bina Nusantara University
Jakarta, Indonesia, 11480
devina@binus.edu

Chandra Kurniawan Wiharja
Language Center
Computer Science Department,
Faculty of Humanities
Bina Nusantara University
Jakarta, Indonesia, 11480
chandra.wiharja@binus.edu

Tjeng Wawan Cenggoro
Bioinformatics and Data Science Research Center
Computer Science Department,
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
wcenggoro@binus.edu

***Abstract*—With the heavy utilization of deep learning, modern chatbot systems require massive training dataset to achieve their peak performance. Without exception, the development of chatbot for the hotel receptionist tasks also demands a massive training dataset. Unfortunately, laborious effort and hefty time are needed to manually collect a dataset with that size. Therefore, an information system is needed to assist the dataset collection. In regards to solve the problem, this research contributes in developing the information system that enables a massive data collection for training a hotel receptionist chatbot in a reasonable time. In detail, the contribution of this research are twofold: (1) developing a realiable crowdsourcing information system design to collect and store the dataset for training hotel receptionist chatbot; and (2) developing a user experience that can stimulate the crowdworkers to provide more data.**

Keywords—crowdsourcing, data collection, chatbot, hotel receptionist chatbot

I. INTRODUCTION

The advancement of chatbot technology is growing expeditiously in recent years that it is approaching the naturalness of human conversation [1]–[3]. This advancement enables chatbots to be deployed in real industrial cases. For instance, Xu et al [4] and Cui et al. [6] built chatbots that can act as customer service. In another research, Ramoliya et al. [5] proved that chatbot can be developed to answer university FAQs.

Other than the aforementioned cases, a chatbot can also be deployed in the hotel industry. For instance, a chatbot can be utilized to automate the receptionist tasks. With the receptionist tasks being automated, the hotel staff can be assigned to un-automatable tasks that ultimately increase the effectiveness and efficiency of the hotel operation.

Despite the potential for application in the hotel industry, developing chatbots in such a narrow niche is difficult. In order to achieve its naturalness in general conversation, the current state-of-the-art chatbot needs 341 GB text for its training dataset [1]. The massive dataset for general conversation can be acquired by crawling public domain social media conversations. However, the strategy cannot be used for the case of a hotel receptionist chatbot development. Therefore, a different strategy is needed to collect a massive dataset for a hotel receptionist chatbot development in a reasonable timeframe. One of the strategies that can be used is crowdsourcing. This strategy utilized a massive number of workers, known as crowdworker, to contribute to a small part of the data. By using the crowdsourcing, it is possible to collect a massive dataset of conversation that involves hotel receptionist. This dataset eventually can be used to train a chatbot that can act as a hotel receptionist.

To answer the challenge of collecting dataset for a hotel receptionist chatbot development, we developed an information system that utilizes the crowdsourcing paradigm. The information system is designed to enable a rapid collection of curated question-answer pairs in the hotel receptionist context. The information system is named as Catbot and referred by that name for the rest of this paper.

II. RELATED WORKS

Not specific to chatbot studies, research in modern Natural Language Processing (NLP) also demand a massive dataset. For that purpose, datasets in NLP are constructed with techniques that allow a massive collection of curated data.

* Authors contributed equally to this work

One of the techniques is crowdsourcing. This technique is mostly used to collect question answering datasets in NLP. The trend was started by Rajpurkar et al. to develop the SQuAD dataset [7]. In SQuAD dataset construction, crowd workers were employed to first provide questions based on passages. Afterwards, the crowdworkers were asked to answer the questions. This approach resulted in more than 100,000 question-answer pairs. Since then, the popular question answering datasets were collected using crowdsourcing [8]–[11]. In the crowdsourcing paradigm, the data is collected by employing a huge number of workers each providing a small part of the data. The workers are called as crowdworker.

Crowdsourcing is not only used to collect datasets for NLP. Other domain that also demands massive datasets also applied crowdsourcing to develop datasets. For instance, the most popular dataset for image classification (ImageNet [12], [13]) was constructed by employing crowdworkers via Amazon Mechanical Turk (AMT). In several domains, a special information system was developed to assist the crowdsourcing process, for example in object categorization [14] and object counting [15], [16].

III. METHODOLOGY

This research was conducted with the steps as shown in Figure 1. These steps are: (1) requirement gathering, (2) creating the information system design, (3) the information system design, and (4) implementation result evaluation.

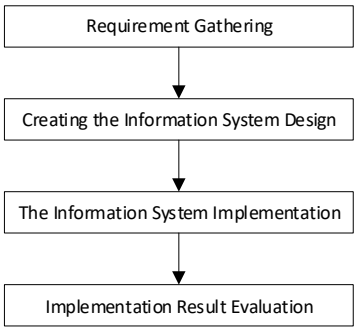


Fig. 1. The research framework.

A. Requirement Gathering

Before we design Catbot, we did a requirement gathering to find essential information system features that are suitable for the environment where Catbot will be implemented. We did the requirement gathering at a hotel management major in a university, the same environment where Catbot was implemented.

B. Creating the Information System Design

Catbot was designed to encourage the crowdworkers to submit questions and answers relevant to the conversation involving a hotel receptionist. The logical flow design of the system is created by using a flowchart and Data Flow Diagram (DFD). The flowchart is illustrated in figure 2. Meanwhile, the DFD level 0 and 1 are represented by figure 3 and 4, respectively.

As elaborated in the flowchart, the system starts by asking the crowdworkers to input their role and identity. In this case, the roles are either students or lecturers, because this system targets the hotel management undergraduate program community in a university. The role is necessary to be provided because of the assumption that the lecturer should provide more valid questions and answers.

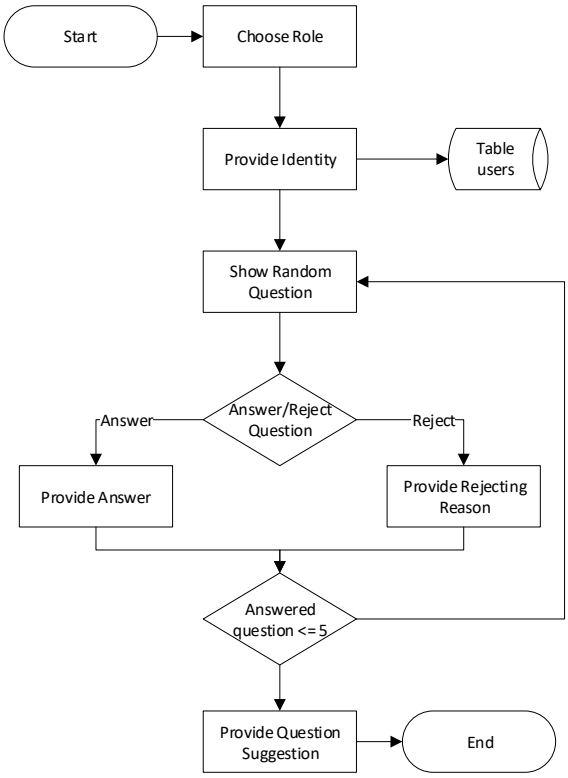


Fig. 2. The flowchart of the Catbot system.

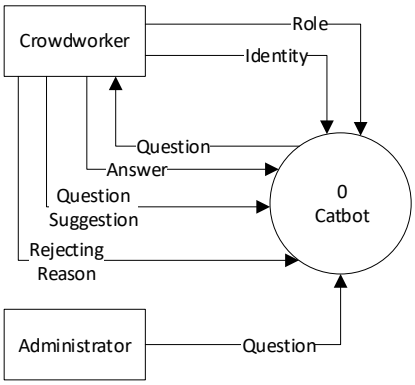


Fig. 3. DFD Level 0 of the Catbot system.

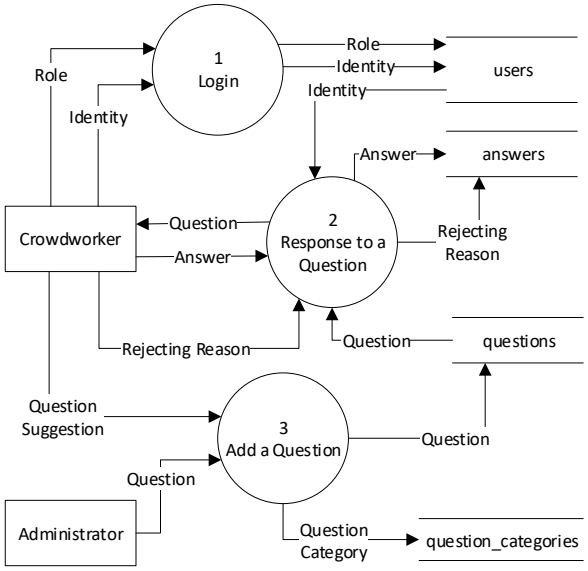


Fig. 4. DFD Level 1 of the Catbot system.

After providing their role and identity, five questions are given to the crowdworkers. The questions provided are a mix of initial questions inputted by the administrator and the suggested questions by other crowdworkers. The initial questions were designed in the context of a conversation between a hotel receptionist and a guest. The suggested questions were also asked to be provided in the same context to the other crowdworkers. The crowdworker is asked to provide a valid answer for each of the questions. After answering questions, the crowdworker is asked to suggest a question on the topic that involves a hotel receptionist. By using this strategy, the number of questions and answers can grow without any limit just by providing at least five initial questions. To control the quality of the questions, the system lets the crowdworker to reject any invalid question provided when they asked to provide answers.

Described by the DFD, there are two actors in the Catbot system: a crowdworker and an administrator. The crowdworker is tasked to provide questions and answers to the system as explained previously. Meanwhile, the administrator's responsibility is to provide initial questions. In the DFD level 1, the Catbot system is detailed to three subprocesses: (1) provide role and identity, (2) response to a question, and (3) add a question. The first and second subprocesses belong exclusively to the crowdworker, while the third process is shared between the crowdworker and the administrator.

The first subprocess stores the role and identity of the crowdworker and stores them in the users' table. The second subprocess handles the process where the crowdworker answer or reject the question. Whether it is an answer or a rejecting reason, the data is stored in the answer table flagged with different statuses. The third sub-process is responsible to receive question input from both the crowdworker and administrator in different cases: input question suggestion for the crowdworker and add initial questions for the administrator This subprocess takes a question as an input and store the question and question category in their respective table. The question category explains whether the question is provided by the crowdworker or the administrator.

To design the database of the Catbot system, we used the Entity Relationship Diagram. The diagram is shown in figure 5. In the diagram, two additional tables do not appear in the DFD. The data in this table are fixed, hence they were inputted by the administrator directly to the database without any module that can modify the data.

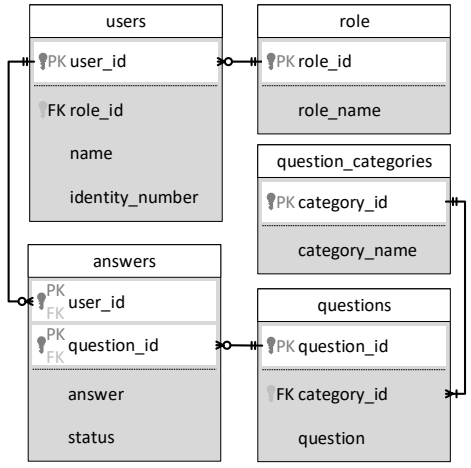


Fig. 5. The ERD of the Catbot Database.

In addition to the logical flow and database design, we designed Catbot to also have an interface that engages the crowdworker to provide valid questions and answers. Figure 6 depicts the interface design of the module to answer a question. The interface intuitively tells the crowdworker that he/she can either answer or reject the question.

If the crowdworker rejects the question, he/she will be prompted to confirm whether he/she decides to reject the question as in figure 7a. If the crowdworker chooses yes, he/she is asked to provide a reason why the question is rejected as in figure 7b. This reason is necessary for the future dataset compiler to validate if the question is truly invalid.

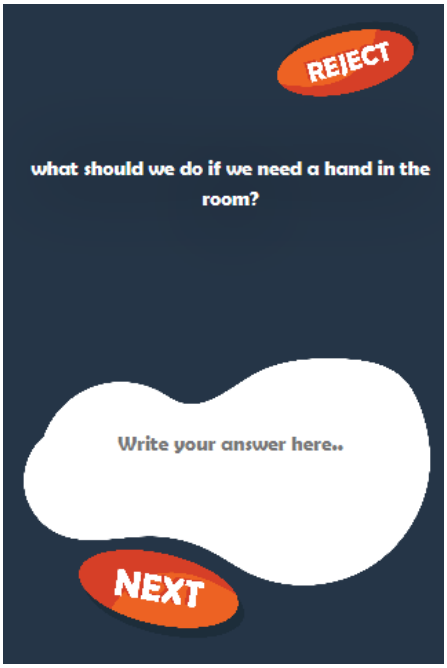
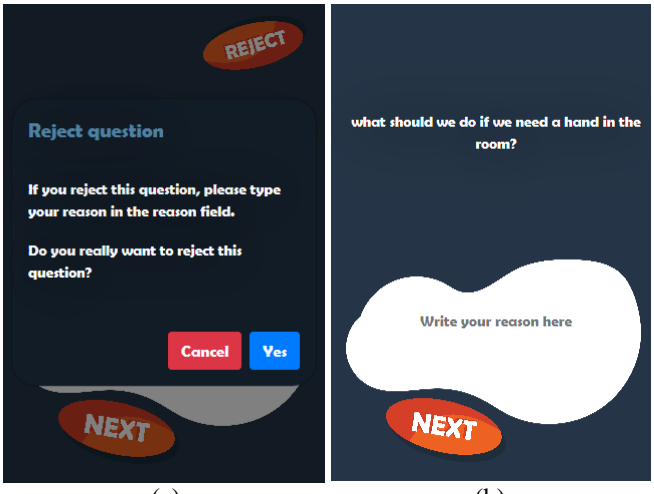


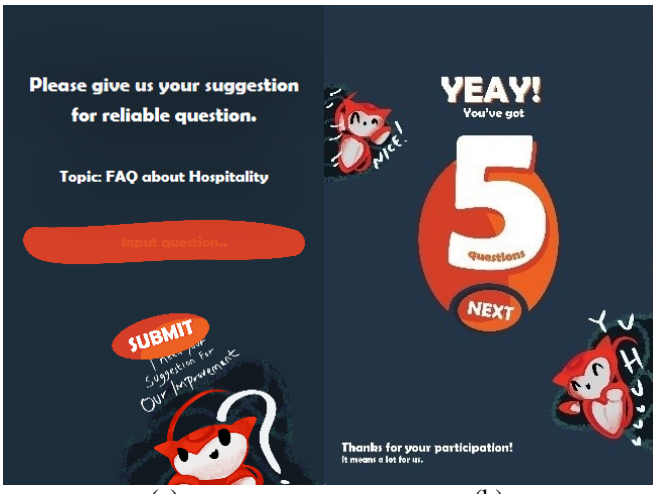
Fig. 6. Interface to answer a question



(a) (b)

Fig. 7. Interface to reject a question

After answering five questions, the crowdworker is asked to provide a suggestion of a valid question in the context of hospitality by using a page which design illustrated in figure 8a. At the end of the session, the crowdworker will be appreciated with a page with a design like in figure 8b to encourage him/her to provide more questions and answers in a different session.



(a) (b)

Fig. 8. (a) Interface to provide a question suggestion; (b) encouraging interface at the end of each session.

C. The Information System Implementation

Catbot was implemented in a community that has expertise in the hotel management domain. In particular, Catbot was implemented to be used by the lecturers and students in a hotel management major in a university.

D. Implementation Result Evaluation

To evaluate the effectiveness of the proposed system, we conducted a user acceptance test for the participating crowdworkers. The test used a questionnaire that consists of four questions:

- How easy it is to use the application? (answer choices: very easy/easy/normal/difficult)
- Are the features in this application work well? (answer choices: very well/well/poor)
- How fast the application runs each of its features? (answer choices: very fast/fast/slow)
- Is the interface of this application engaging? (answer choices: very engaging/engaging/not engaging)

Additionally, we also measured the speed of the collection and qualitatively assessed the rejected questions.

IV. Results and Discussions

A. User Acceptance Test

We recapitulate the result of the user acceptance test in table I to IV, in which each table contains the answers to the four questions in the test. The result of the user acceptance for all questions shows that the Catbot system is generally well accepted by the crowdworkers. 62% of the crowdworkers agreed that the Catbot system was easy to use, 92% voted that the features in the system worked well, 86% felt that the features run fast, and 78% believed that the system was engaging.

B. Dataset Collection Result

In five days, we successfully collected 234 question-answer pairs using Catbot. Among the submitted questions, 25 questions were rejected by other crowdworkers as depicted in figure 9. It can be seen that the rejection feature can capture invalid questions. However, not all the questions are invalid. In this case, the rejecting reasons (stored in the answer column in figure 9) play a role to help the future dataset compiler to judge whether a question is valid or not.

TABLE I. Recapitulation of answers for question no. 1

Question: How easy it is to use the application?		
Answer	**Number of responses**	**Percentage**
Very easy	0	0%
Easy	31	62%
Normal	16	32%
Difficult	3	6%

TABLE II. Recapitulation of answers for question no. 2

Question: Are the features in this application work well?		
Answer	**Number of responses**	**Percentage**
Very well	2	4%
Well	46	92%
Poor	2	4%

TABLE III. Recapitulation of answers for question no. 3

Question: How fast the application runs each of its features?		
Answer	**Number of responses**	**Percentage**
Very fast	1	2%
Fast	43	86%
Slow	6	12%

TABLE IV. Recapitulation of answers for question no. 4

Question: Is the interface of this application engaging?		
Answer	**Number of responses**	**Percentage**
Very engaging	8	16%
Engaging	39	78%
Not Engaging	3	6%

question	answer
This website can be developed become better.	The question doesn't match as instructed
How far and how long will it take to reach the hot...	the question is ambiguous
May I help you with anything today?	the question is ambiguous
How can I propose special request to the hotel?	I never heard that question
I'm sorry, we don't have any rooms available. Will...	This question is not related
How many guests are with you?	the question is ambiguous
Would you please show me your details? Your ID and...	the question is ambiguous
yoo	dummy question
Can I use the beach before check-in and after chec...	The question is ambiguous
Why haven't I received a booking confirmation in t...	Wrong move
Is there anything you need?	the question is ambiguous
When will I get a refund?	i cant understand the question
did you have a elevator in the hotel?	the question is ambiguous
Can I, as a hotel agent, check room availability a...	i dont understand the question
yoo	the question is ambiguous
When will I get a refund?	Why would u ask for a refund from me😂
Please sign your name here. Do you need a pen?	Yes
How may I help you?	incorrect question
review again ur question, bcs i think u all guys c...	Ok
Please sign your name here. Do you need a pen?	The question doesn't match as instructed
Do you need anything more sir/madam?	The question doesn't match as instructed
hehe	Not a question
What's the password of wifi?	The question is ambiguous
How long can I swim?	the question is ambiguous
What can I do if I find wrong the order details?	I don't udnerstand this question

Fig. 9. Rejected questions

V. Conclusion

In this paper, we presented Catbot, a crowdsourcing system that enables the collection of a massive dataset for

developing a chatbot in the context of hotel receptionist conversation. The system received a good response by the crowdworkers based on the result of the user acceptance test. The rejection feature also assists the future dataset compiler to clean the dataset collected by this system. In the future, this system can be deployed on a large scale to collect the massive dataset needed. Future studies can use the constructed dataset to develop a hotel receptionist chatbot.

REFERENCES

[1] D. Adiwardana *et al.*, "Towards a human-like open-domain chatbot," *arXiv Prepr. arXiv2001.09977*, 2020.

[2] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The design and implementation of xiaoice, an empathetic social chatbot," *Comput. Linguist.*, vol. 46, no. 1, pp. 53–93, 2020.

[3] Y. Zhang *et al.*, "DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation," *arXiv Prepr. arXiv1911.00536*, 2019.

[4] 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, pp. 3506–3510.

[5] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1525–1530.

[6] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, "Superagent: A customer service chatbot for e-commerce websites," in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 97–102.

[7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv Prepr. arXiv1606.05250*, 2016.

[8] E. Choi *et al.*, "QuAC: Question Answering in Context," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2174–2184.

[9] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 784–789.

[10] S. Reddy, D. Chen, and C. D. Manning, "Coqa: A conversational question answering challenge," *Trans. Assoc. Comput. Linguist.*, vol. 7, pp. 249–266, 2019.

[11] T. Kwiatkowski *et al.*, "Natural questions: a benchmark for question answering research," *Trans. Assoc. Comput. Linguist.*, vol. 7, pp. 453–466, 2019.

[12] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[13] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[14] C. Sun, N. Rampalli, F. Yang, and A. Doan, "Chimera: Large-Scale Classification using Machine Learning, Rules, and Crowdsourcing," *Proc. VLDB Endow.*, vol. 7, no. 13, pp. 1529–1540, Aug. 2014.

[15] T. W. Cenggoro, F. Tanzil, A. H. Aslamiah, E. K. Karuppiah, and B. Pardamean, "Crowdsourcing annotation system of object counting dataset for deep learning algorithm," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 195, no. 1, 2018.

[16] B. Pardamean, T. W. Cenggoro, B. J. Chandra, and Rahutomo, "Data Annotation System for Intelligent Energy Conservator in Smart Building," *IOP Conf. Ser. Earth Environ. Sci.*, 2020.