Utilizing Crowdsourced Asynchronous Chat for Efficient Collection of Dialogue Dataset

Kazushi Ikeda, Keiichiro Hoashi
KDDI Research, Inc.
{kz-ikeda, hoashi}@kddi-research.jp

Abstract

In this paper, we design a crowd-powered system to efficiently collect data for training dialogue systems. Conventional systems assign dialogue roles to a pair of crowd workers, and record their interaction on an online chat. In this framework, the pair is required to work simultaneously, and one worker must wait for the other when he/she is writing a message, which decreases work efficiency. Our proposed system allows multiple workers to create dialogues in an asynchronous manner, which relieves workers from time restrictions. We have conducted an experiment using our system on a crowdsourcing platform to evaluate the efficiency and the quality of dialogue collection. Results show that our system can reduce the necessary time to input a message by 68% while maintaining quality.

Introduction

Dialogue systems like Siri and Google Assistant have increasingly been important as interfaces of complex computer systems (McTear 2002). Except for some human-powered conversation assistants (Lasecki et al. 2013a; Huang et al. 2016), most dialogue systems respond to users based on pre-defined rules (Weizenbaum 1966; Bennacef et al. 1996) or machine learning methods (Raymond and Riccardi 2007; Henderson et al. 2013). Such dialogue systems require large scale dialogue datasets for making rules or training.

Existing datasets can be used to train dialogue systems for specific purposes like Q&A systems (Kiyota et al. 2002). In most cases, however, existing datasets consist of a limited number of conversation patterns. In order to increase the coverage of dialogue datasets, the utilization of crowdsourcing has been proposed (Bessho et al. 2012).

Conventional crowd-powered systems for collecting dialogue datasets create a pair of crowd workers, ask them to converse with each other on an online chat system, and record their interactions (Lasecki et al. 2013b; Tsukahara and Uchiumi 2015). Although these crowd-based systems can create highly qualitative task oriented datasets, there are issues related to the time necessary to create such datasets, which decrease the efficiency of the overall process. One is that the workers must work simultaneously to generate a dialogue, which imposes restriction on their working time. Furthermore, even when a pair of workers are able to work on a task, one worker must wait for the other when her partner is inputting a message. Reducing the time of crowd-working is essential to save cost (Krishna et al. 2016) and also to increase wage rate, in order to attract more participation of workers (Mason and Watts 2009).

We propose a novel design of a crowd-powered system to collect dialogue data efficiently. Basically, our system collects dialogues on a chat-like interface similar to conventional systems. However, unlike conventional systems, our system enables multiple workers to create dialogues in an asynchronous manner. This framework frees crowd-workers from time constraints, and increases work efficiency.

We have conducted an experiment using our system on a crowdsourcing platform to evaluate the efficiency and the quality of dialogue collection. Efficiency has been evaluated based on required time and cost to create dialogues. Quality has been evaluated by quantitative measurements such as message length and amount of information, and qualitative measurements such as consistency and ease of answering.

The experimental results show that our system can reduce the necessary time to input a message by 68%, and also reduce the cost by 62%. Worker questionnaires show that the satisfaction with reward was higher and the burden was lower in our system than that of the conventional system. Quality measurements show that our system maintains the same quality level as the conventional system in fun-

damental measures for creating dialogues, such as amount of information and ease of answering.

The rest of the paper is organized as follows. We first review related work on conversational dialogue systems and crowd-powered systems for collecting dialogues. Second, we define our research problem, and explain both conventional crowd-powered systems and our proposed systems. We then introduce the experimental design to compare the performance of our system to conventional systems and present experimental results. Finally, we describe limitations and future work.

Related Work

Conversational Dialogue Systems

Many existing dialogue systems have been used as interfaces of smartphones and information retrieval systems (McTear et al. 2002). One of the representative use cases is a question answering system. The system accepts questions by natural language and searches for required information from the database (Fader et al. 2014; Ferrucci et al. 2010). Expert systems that can respond to questions on specialized domains such as medical care have been proposed (Cao et al. 2011). Dialogue systems for Web search (Jiang et al. 2013) and navigation (Belvin et al. 2001) have also been proposed.

These systems are called task-oriented dialogue systems, where the user has a clear purpose to use the system. On the other hand, non-task-oriented dialogue systems have also been attracting attention recently, where the purpose of the user is to make a casual conversation with the system. Some dialog systems can talk about specific topics like movie reviews (Dodge et al. 2016). Other systems are designed to make humorous conversations (Augello et al. 2008). Dialogue systems designed for chatting with elderly people have also been proposed (Vardoulakis et al. 2012; Sabelli et al. 2011).

Except for some human-powered conversation assistants (Lasecki et al. 2013a; Huang et al. 2016), most dialogue systems respond to users based on pre-defined rules or machine learning methods. In the rule based method (Weizenbaum 1966; Bennacef et al. 1996), rules are manually created according to the assumed patterns of user's utterances. The system responds to users based on the predefined rules. In machine learning methods (Raymond and Riccardi 2007; Henderson et al. 2013), dialogue datasets which consist of user's utterances and the system's responses are used for training. The system responds based on the pre-trained relevance of utterances. In recent years, response generation methods using Deep Learning (Lowe et al. 2017; Li et al. 2016a) have also been drawing attention.

In both methods, in order to respond to user's various utterances in an accurate and comprehensive manner, the system requires large dialogue datasets describing the assumed conversational scenarios. Existing datasets can be used for specific purposes like Q&A systems (Kiyota et al. 2002). Although some systems use Twitter data (Bessho et al. 2012) or movie scripts (Li et al. 2016b), existing datasets are often insufficient to cover the wide variety of expected user utterances.

Crowdsourced Scenario Writing

In order to increase the coverage of dialogue datasets, the utilization of crowdsourcing has been proposed. Bessho et al. have proposed a method to use conversations on Twitter as dialog datasets (Bessho et al. 2012). In case there is no tweet on database similar to the user's utterance, the system requests workers on crowdsourcing platforms to supplement the missing scenarios.

ChatCollect (Lasecki et al. 2013b) is a system which hires a pair of crowd workers, asks them to converse with each other on an online chat system, and records their interactions as dialogue datasets. This framework enables construction of large scale dialogue datasets in a short period of time by recruiting a large number of workers. In this framework, however, the workers must work simultaneously to create a dialogue, which imposes restriction on their working time. Furthermore, even when workers are able to work on a task simultaneously, one worker must wait for the other when her partner is inputting a message. Lasecki et al. report that workers have been required to wait for about six minutes when they fail to find their partner, and for about two minutes even when they successfully find their partner.

Wang et al. propose a method to acquire a language corpus for dialogue system using crowdsourcing (Wang et al. 2012). Workers are asked to create sentences assuming designated situations such as reserving a hotel. Since sentences created by a single worker are often biased in terms of expressions, they ask multiple workers to create sentences to get various expressions. Tsukahara et al. propose a system for collecting dialogues and annotating tags such as topics and speech acts to the dialogues by chatting on Slack (Tsukahara and Uchiumi 2015).

Problem Definition

As mentioned in related work, many task-oriented dialogue systems have been proposed, and some of them have already been practically used. Assuming that users interact with dialogue systems on a daily basis, non-task-oriented conversations like chat will be essential for the dialogue system as well as performing tasks. Dialogue systems require larger dialogue datasets for chatting than for perform-

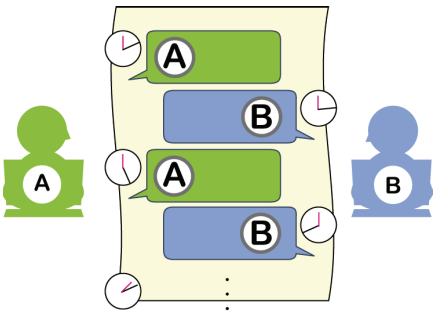

Figure 1: Overview of the Conventional System.

ing specific tasks due to the necessity to respond to various topics. In this paper, our focus is on collecting dialogue datasets for casual conversations.

Existing crowd-powered systems can create large scale dialogue datasets in a short period of time by recruiting a large number of workers. However, there are issues related to the efficiency to create dialogues. Although it is conceivable that a single worker plays two roles to create a dialogue, as reported by Wang et al. (Wang et al. 2012), a single worker may create limited conversation patterns.

Considering these problems, we propose a novel design of a crowd-powered system, where multiple workers can create dialogues in an asynchronous manner. In order to evaluate efficiency of dialogue creation and quality of dialogues, we have conducted experiments of constructing dialogue datasets on a crowdsourcing platform.

Conventional Crowdsourced Scenario Writing

Several methods have been proposed for constructing dialog datasets using crowdsourcing (Lasecki et al. 2013b; Tsukahara and Uchiumi 2015). Either method hires a pair of crowd workers and ask them to converse with each other on an online chat system in real-time. In order to understand the features of the conventional methods, we describe the details of ChatCollect system (Lasecki et al. 2013b).

The ChatCollect system is designed to collect task-oriented dialog data. The system manager sets tasks like flight reservation, car dealing, etc. which he/she wants to collect dialogue about. The system hires a pair of workers from a crowdsourcing marketplace in real-time. Figure 1 illustrates the overview of the system. The system assigns roles of an assistant (role A in Figure 1) and a user (role B in Figure 1) to a pair of workers. The pair of workers play their roles on an online chat interface. The first worker who accesses to the system is assigned the role of the assistant and is instructed to help someone with the task at hand, such as finding a flight. The second worker is assigned the role of the user and is instructed to complete a task with the help of the assistant (e.g., find flight).

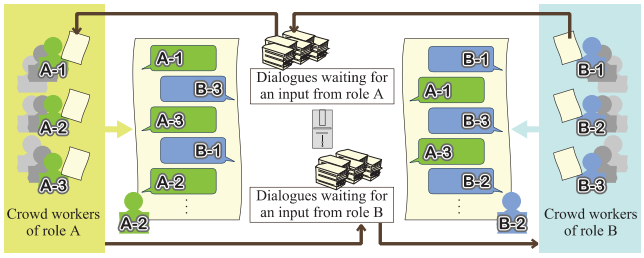

Figure 2: Overview of our Proposed System.

Since the assistant is hired first and has to wait for the user, the assistant needs to be paid for the waiting time. Since the worker cannot perform any task in this waiting time, the payment can be considered as unnecessary cost. Furthermore, one worker must wait for the other when her partner is inputting a message, which also decreases the efficiency of the task performance.

Asynchronous Scenario Writing

We introduce the design of our proposed method of asynchronous scenario writing system. The main feature of our method is that more than two workers can cooperate to create one dialogue. With this mechanism, workers are not required to be paired in real-time and can work alone in their available time. As a result, this framework liberates crowd-workers from time constraints, and increases work efficiency.

Figure 2 illustrates the overview of our system. In our system, multiple workers are assigned to one of the roles (role A or B). Workers can join the system whenever they are available. When workers join the system, the system assigns one of dialogues that require input from a worker. Like this, each dialogue is input by more than two workers.

In Figure 2, for example, when worker A-2 joins the system, one of the dialogues in which an input from workers of role A is required, is selected. Then, worker A-2 inputs her message on the dialogue. Five messages have been input in total by five workers, A-1, B-3, A-3, B-1 and A-2. The dialogue is stored on the queue of dialogues waiting for an input of role B.

To explain the mechanism of the proposed method in a clearer way, we show the interface of the system in Figure-3. When a worker starts a task, a partially completed dialogue is displayed. Here, the worker of role A accesses the system. The topic of the dialogue is about “Work”, and four messages have already been input by other workers. The worker reads these four messages, and inputs the fifth message as role A. After submitting the message, the next task, i.e., a different dialogue, is displayed on the system. By repeating this procedure, workers can continuously work on tasks without waiting for other workers.

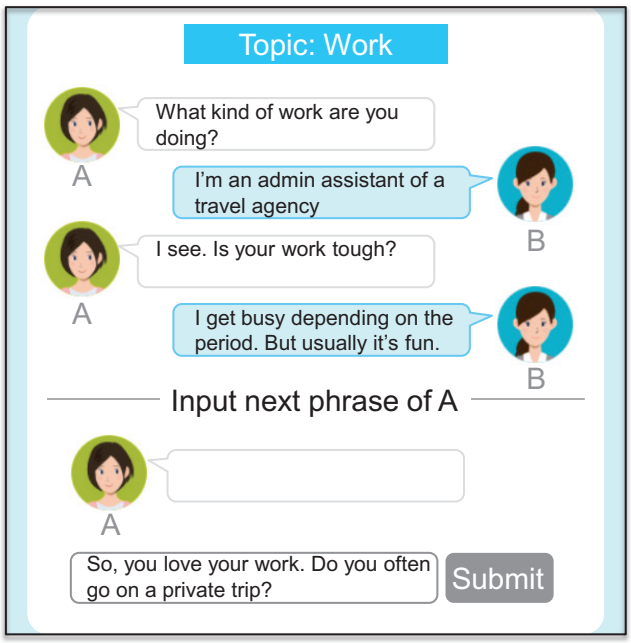

Figure 3: Interface of the Proposed System.

As the number of messages in a dialogue increases, time required for reading previous messages becomes longer. Since it is not necessary to present all messages to understand the conversation background, the system displays only the last N messages (e.g. N=4 in Figure 3) to the worker. We evaluate the optimum value of N in the experiment.

Experimental Design

To evaluate the effectiveness of our system, we conduct an experiment to collect dialogues on a crowdsourcing platform. We explain about the task details, conditions to compare, measurements for evaluation and worker pool.

Task

In this study, we aim at collecting datasets for a dialogue system designed for chatting with users. In order to create a dialogue for chat, it is necessary to decide topics, attributes of the dialog system, its relationship with the user, and so forth. It has been reported that conversation topics vary depending on the attributes of a dialogue system (gender, age, resemblance to a human or a robot, etc.) (Sugiyama et al. 2014). These attributes are decided according to the design concept and targeting user of the dialogue system.

As an example, in this paper, we assume a situation where a developer designs a dialogue system that has a female appearance and friendly converses with human female users. Under this design concept, we assume a situation where a developer wants to construct large scale dialogue datasets of conversations between two females who want to build good relations with each other.

Basic Information	
Name	Mami Sasaki
Age	25
Gender	Aug 11th, 1990
Living Area	Yokohama, Kanagawa
…	
Family	
Structure	Parents and an elder brother
Marriage	No
Children	No
Live with	Parents
…	
Job	
Occupation	Admin assistant of a travel agency
Office location	Shinagawa, Tokyo
…	

Table 1: Example of Dialogue System Personality (Role B)

Topic	Initial Utterance
Family	Are you living with someone?
Holidays	How do you spend holidays?
Friend	Do you meet with your school friends?
Sports	Are you doing sports?
Learning	Is there anything you are studying now?
Favorite food	What kind of foods do you like?
Cooking	Do you like cooking?
Work	What kind of work are you doing?
…	…

Table 2: Example of Topics and the Initial Utterance

We set a detailed personality of 75 items as shown in Table 1 for one of the two females that corresponds to the dialogue system (Role B). The other female who corresponds to the user (Role A) is only defined as a female of the same generation as role B. Workers of role A are instructed to respond based on their attributes in case information about other attributes is required for conversations, so that we can collect user dialogues of various attributes. We asked four female subjects what conversation topics are taken up between females in such a relationship, and set 50 representative topics for conversations shown in Table 2. We construct 30 dialogues for each topic starting from initial utterances in the table to collect various conversations in the same topic.

Method	Real-time	Roles	Rewards
RC	Yes	Either A or B	Fixed payment of 4 yen / minute for waiting time Per task payment of 4 yen / message
AC (N=2,6,16)	No	Either A or B	Per task payment of 4 yen / message
SW	No	Both A and B	Per task payment of 4 yen / message

Table 3: Summary of Experimental Conditions.

The workers input messages following the initial utterances for each topic. Unlike task-oriented dialogues, it is difficult to define a completion state in chat. Therefore, the length of a dialogue (the number of messages in a dialogue) was fixed to sixteen (16). Since we recruit workers on a Japanese crowdsourcing platform, the messages are input in Japanese. We think the influence of language on the experimental result is not significant. The workers are instructed fundamental conditions such as the minimal number of characters per message (10), and prohibition of copying sentences from public articles.

Conditions

We compare the performance of three methods. The first one is the conventional Real-time Chat method (RC) like ChatCollect. The second one is our proposed Asynchronous Chat method (AC). The last one is Single Worker method (SW), where a single worker plays both two roles, and inputs the entire dialogue by herself. Table 3 summarizes the features of the three methods.

In RC, two workers are paired and work in real-time. Each worker is assigned either role A or role B. Rewards are provided in similar manner to Chat Collect as fixed payment for waiting time and also per task payment for each completed messages.

In AC, each worker independently and asynchronously inputs messages. Each worker is assigned either role A or role B. Rewards are provided only as per task payment for each completed messages. In AC, latest N messages are presented to workers when they input a message. In the experiment, we evaluate three conditions of N = 2, 6 and 16.

In SW, each worker independently inputs messages. Each worker plays both role A and role B, and inputs 16 messages continuously. Rewards are provided only as per task payment for each completed messages.

Measurements

We compare the five conditions from the viewpoint of efficiency, quality and worker behaviors which are commonly used for the evaluation of systems utilizing crowdsourcing.

Efficiency

Efficiency is evaluated from (1) required time and (2) cost to create dialogues. These are related to each other, and the cost can be reduced as the required time becomes smaller. The required time is defined as mean time to input a message (Huang et al. 2016; Wang et al. 2012). In RC, we evaluate the waiting time after one worker accesses the system until the other worker arrives (Lasecki et al. 2013b). Regarding the cost, fixed payment is provided in all conditions. In RC, a fixed reward is additionally provided for waiting time. Therefore, we compare total cost required to construct dialogue datasets after the experiment.

Quality

Quality of dialogue systems has been evaluated from several viewpoints in existing studies. Based on such studies, we define evaluation measurements for dialogues. The measurements can be classified into (1) quantitative and (2) qualitative.

As quantitative measurements, we evaluate the number of characters per message (Huang 2016 et al.; Laseck et al. 2013b), grammatical correctness (Li et al. 2016a) and amount of information (Bessho et al. 2012; Li 2016a). Workers are instructed to input at least 10 characters, but they may input messages as shortly as possible to reduce time to complete tasks. Grammatical correctness is manually evaluated whether messages contain misspellings or grammatical errors. For each condition, 100 dialogues are randomly selected from the created dataset. Then, four annotators label grammatical errors to 16 messages included in each dialogue. We consider messages have grammatical errors when three or more annotators labeled the message as erroneous.

Dialogues that contain diverse topics are more beneficial for dialogue systems than those only contain repetition of the same sentences or short responses like “Yeah” and “I see.” Li et al. defined the amount of information in conversations of dialogue systems based on the number of different words appearing in the sentences (Li et al. 2016a). We conduct morphological analysis on the collected dialogue sentences and count the number of different morphemes per fixed amount of morphemes.

Measurements such as consistency (Laseck et al. 2013b, Li et al. 2016b), malicious worker behavior (Huang et al. 2016) and ease of answering (Li et al. 2016a) are used for qualitative evaluation of dialogues. These measurements are labeled by four annotators in the same manner as grammatical errors. Consistency is labeled for each message whether the message is consistent with the content of previous messages in the dialogue, and also consistent with the personality of the roles. To evaluate malicious worker

behavior, annotators detect messages that are meaningless, input randomly, offensive and against public order. Ease of answering is an important measurement for activating conversation between dialogues and users. We present a part of collected dialogues to the annotator, and ask to review how easily they can input the next message in a five-grade scale.

Worker Behavior and Worker Feedback

On crowdsourcing market, it has been reported that a certain percentage of workers do not work as instructed or perform malicious behavior (Bernstein et al. 2015). By analyzing the aforementioned quantitative and qualitative measurements on a worker basis, we compare the proportion of these lazy workers among conditions.

After the experiment, we ask a questionnaire about their motivation and work load in each method. Regarding motivation, we asked about (1) satisfaction level of their payment, (2) whether they feel the task fun, and (3) whether they want to do the same task again. For the worker's burden, we referred to NASA TLX (Hart and Staveland 1988), and ask 5 questions about Mental Demand, Temporal Demand, Effort, Performance and Frustration Level. Since the task is conducted online, we omitted the question about Physical Demand.

Workers

We recruit 60 participants for each of the five conditions on a major crowdsourcing platform in Japan. In order to match the attributes of workers to the roles, we hire female workers in their 20s to 40s. In RC and AC, 30 workers are assigned to role A and role B respectively. In RC, workers require a partner to work with. However, if they are allowed to work at an arbitrary time, their access time to the system would be dispersed. As the result, their waiting time would increase. To avoid this problem, we limit time period to access the system from 12 PM to 3PM. The experiment period is two weeks in total.

Results

Efficiency

Table 4 shows time to input a message per worker in the five conditions. In RC, mean time to input a message per worker was 133 sec. In AC, input time was affected by N, and 52 sec when N = 16, 42 sec when N = 6 and N=2. In SW, mean time was 29 sec. In RC, since two workers made a pair to create a dialogue, if one worker took t seconds to input a message, 2t seconds were estimated to be spent to input the message. Even taking this into consideration, RC requires more than three times compared with AC (N=2 and 6) and SW. In AC, input time for N=6 is shorter than N=16. This is because time to read past conversations was shorter. However, the mean input time was almost the same when N = 2 and N = 6.

	Mean (sec)	Std. dev	n	p-value
RC	133	124	16,704	$2.2e^{-16}$ **
AC(N=16)	52	51	22,500	
AC(N=6)	42	43	22,500	
AC(N=2)	42	47	22,500	
SW	29	38	21,447	

Table 4: Time to Input a Message per Worker

	# of messages	# of dialogues	Total cost (yen)	Cost per message (yen)
RC	16,704	1,058	175,148	10.5
AC(N=16)	22,500	1,500	90,000	4
AC(N=6)	22,500	1,500	90,000	4
AC(N=2)	22,500	1,500	90,000	4
SW	21,447	1,500	85,788	4

Table 5: Numbers of Input Messages, Number of Created Dialogues, Total Cost and Cost per Message.

Kruskal-Wallis test shows significant difference ($H = 7407, p < .01$) among the five conditions in the mean input time. Wilcoxon rank sum test and multiple comparisons based on Bonferroni method with the nominal significant level ($\alpha' = .005$) show significant difference for all ${}_5C_2$ combinations of conditions, because the number of messages n was extremely large. For the rest of the results that show significant differences in the table, the tests were conducted in the same manner.

Table 5 shows the numbers of input messages, number of created dialogues, total cost and cost per message. In AC and SW, all 30 dialogues for each of the 50 topics have been created in the two weeks. However, since we limited the work time on 12PM to 3PM in RC, workers could not complete all dialogues. Although each dialogue consists of 16 messages, some dialogues was not completed because the workers left the system before completing the dialogue. In RC, fixed payments were provided for workers' waiting time. The mean waiting time for a dialogue was 93 sec. As the result, cost per message was high as 10.5 yen in RC whereas cost per message was 4 yen in other per task payment conditions.

Quality

Table 6 shows the number of characters per message. The mean length of a message was longest as 28.3 in RC, and

shortest as 19.2 in SW. In AC, the mean length was not affected by the value of N, and was between that of RC and SW.

	Mean (char.)	Std. dev	n	p-value
RC	28.3	12.9	16,704	$2.2e^{-16}$ **
AC(N=16)	24.2	10.2	22,500	
AC(N=6)	20.7	7.7	22,500	
AC(N=2)	22.3	8.9	22,500	
SW	19.2	10.5	21,447	

Table 6: Number of Characters per Message

	Grammatical error rate	# of *different* morphemes
RC	0.063	1,752
AC(N=16)	0.038	1,759
AC(N=6)	0.049	1,708
AC(N=2)	0.013	1,731
SW	0.023	1,583

Table 7: Grammatical Error Rate and Number of Different Morphemes

	Inconsistency Rate	Malicious Work
RC	0.07	0.000
AC(N=16)	0.09	0.000
AC(N=6)	0.03	0.000
AC(N=2)	0.14	0.000
SW	0.12	0.083

Table 8: Inconsistency Rate and Rate of Malicious Work

Table 7 shows the rate of messages that contains grammatical errors and the number of different morphemes per 10,000 morphemes. The grammatical error rate was highest in RC conditions, probably because the workers feel time pressures in the real-time conversations and could not take time to correct grammatical errors.

The number of different morphemes was relatively small in SW conditions. In SW, we often observed dialogues where most of the messages consist of short responses and contain few content. Some workers entered the same dialogues apart from designated topics. We will discuss this case with examples in the discussion section.

Table 8 shows the rate of inconsistent messages and the rate of messages labeled as malicious work. The inconsistency rate was relatively high in AC (N=2) and SW. The annotators detected two types of inconsistency, where the message is inconsistent with (1) the content of previous messages in the dialogue and (2) the personality of the roles. The former case was often observed in AC (N=2), whereas the latter case was often observed in SW. This is probably because the workers could not understand the context of conversations from the presented two messages in AC (N=2) condition. In SW, workers might be confused because they were asked to perform the two roles at the same time.

	Mean	Std. dev	n	p-value
RC	3.13	1.28	400	0.0053**
AC(N=16)	3.18	1.17	400	
AC(N=6)	3.10	1.20	400	
AC(N=2)	2.93	1.06	400	
SW	2.79	1.31	400	

Table 9: Ease of Answering Labeled by Four Annotators.

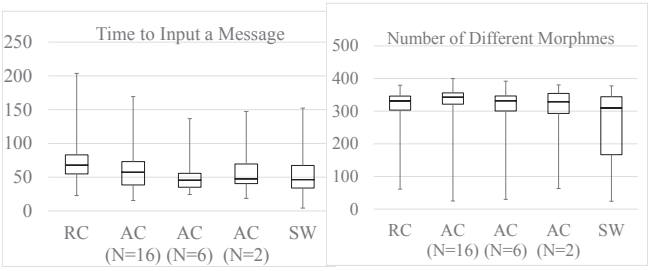

Figure 4: Worker Distribution in Input Time and Number of Different Morphemes

In SW, we observed many inappropriate conversations. Many messages were labeled as to be input randomly. In other conditions tasks were performed in cooperation with other workers. In SW condition, however, workers work independently, so they are not conscious of being seen by other workers. As a result, the number of lazy workers increased.

Table 9 shows the ease of answering labeled by four annotators. The annotators were presented a part of the collected dialogues, and were asked to rate how easily they can input the next message in a five-grade scale. The mean score was low in SW. In SW, each message seemed easy to answer for the worker who created it, but it was difficult to answer for other people.

Worker Behavior and Worker Feedback

Figure 4 shows worker distribution in time to input a message and the number of different morphemes. In AC and

SW, we confirm that most of workers have input a message in less time than the median time of workers in RC. In SW, many workers have created dialogues with very few morphemes. We confirmed that these workers repeatedly input the same contents in different dialogues. These results are consistent with the experimental results shown so far.

Table 10 shows the summary of the worker questionnaire conducted after the experiment. Each question was answered in 7-grade scale (1: low worker burden or low motivation and 7: high worker burden or high motivation).

	RC	AC (N=16)	AC (N=6)	AC (N=2)	SW	p-value
Worker Burden						
Mental Demand	2.00	2.36	2.36	1.93	2.48	0.393
Temporal Demand	**2.84**	**1.83**	**2.05**	**1.80**	**2.34**	**0.029***
Effort,	2.75	2.81	2.79	3.17	3.02	0.781
Performance	2.66	2.81	2.62	3.10	3.02	0.571
Frustration Level	2.88	2.23	2.72	2.20	2.07	0.211
Worker Motivation						
Payment	**4.63**	**5.11**	**4.99**	**5.13**	**5.45**	**0.047***
Fun	5.88	5.96	5.92	5.73	6.11	0.892
Repetition	**6.22**	**6.87**	**6.54**	**6.63**	**6.70**	**0.042***
n	32	53	39	30	44	

Table 10: Summary of Worker Questionnaire

For worker burden, temporal demand was higher in RC conditions than other conditions, which is related to time pressure of work. Since conversations were performed in real time, workers may have felt time pressure not to let their partner wait. For worker motivation, the satisfaction for the payment and intention to accept the same task were lower in RC than other conditions. This result indicates that workers in RC thought the tasks are inefficient that does not match the rewards.

Discussion and Limitations

We discuss the experimental results. In AC, when N = 16, input time increased compared to N = 2 or 6. In terms of efficiency, N = 2 or 6 is effective. AC (N=2, 6) and SW reduced input time per worker by 68% and 78% compared to that in RC, respectively.

In RC, the workers were provided fixed payment for their waiting time in addition to the per task payment. AC and SW could save this cost for the waiting time and reduced cost per message by 62% compared to RC. Worker questionnaire shows that the levels of satisfaction with reward in AC and SW were higher than that of RC. This is probably because the task is sufficiently efficient for the reward. This result indicates the task price can be further reduced in AC and SW.

From the viewpoint of quality, in SW, amount of information and ease of answering decreased compared to those in RC. These measurements are especially important in creating dialogues. In AC (N=2), consistency of dialogue decreased because contextual information of dialogues

Dialogue created in RC
A: Where do you usually do shopping around?
B: Mostly, near my office in Shinagawa or my local area in Yokohama.
A: I see. Do you have a favorite shop?
B: Speaking of clothes, I like the brand of earth music ecology.
A: I am not familiar with fashion, what kind of fashion is it?
B: Well, how to say… They have kind of dressy clothes that matches any situations.
A: I adore dressy clothes. Clothes that matches any situations are very convenient. I always buy clothes online.
B: What kind of clothes do you buy online?
Dialogue created in AC
A: Where do you usually do shopping around?
B: I often do shopping around my office in Shinagawa.
A: Do you have any shop recommendation around Shinagawa?
B: It depends on your preference. Any shops are good!
A: Is that so? I thought Shinagawa is an office town.
B: I often hang around after work.
A: It sounds good you have time to hang around after work. I am always busy recently.
B: The station building is fulfilling and is easy to go casually.
Dialogue created in SW
A: Where do you usually do shopping around?
B: It depends on occasions, right?
A: Is that so? I see.
B: How about you?
A: I'm similar to you.
B: I see. That's right.
A: Yes, that's true.
B: We have a lot in common.

Table 11: Examples of Dialogues about the Topic of "Favorite Town" Created in RC, AC (N=6) and SW.

was not sufficiently provided to the workers. In the condition of N=6 and 16, the quality measurements were as high as those of RC except for the message length. Together

with the efficiency aspect, N=6 gives a good balance in AC.

Table 11 shows examples of dialogues about the same topic created in RC, AC (N=6) and SW. In RC, the third message of role B delves into her previous message. Since the same pair of workers create whole dialogue, they sometimes provide additional information to previous messages. In RC, short responses like "I see" and "Well" were frequently observed due to the real-time conversation nature. In AC, the workers take previous messages into consideration but do not deepen the topics too much because they cannot share background information nor have deep knowledge about a specific topic which another worker inputs. This could cause the differences in the mean length of messages between RC and AC. In SW, some workers frequently input unnecessary responses in order to satisfy the task condition of 10 characters. Other workers repeatedly input the same dialogues in different topics by ignoring designated topics. Unlike RC and AC, since workers do not cooperate with other workers in SW, they input dialogues as shortly as possible, which results in lack of information and makes difficult to answer for other people. This could be a fatal problem as dialogues for conversational agents.

From the discussion so far, we confirmed that our proposed system (AC) can increase the efficiency while maintaining the same quality as the conventional system (RC). In addition, the worker questionnaire indicates the burden of the task was lower in our system.

The limitation of our proposed system is that there is no mechanism to efficiently share the background information which did not appear in messages. In our system, it is difficult to supplement information to previous messages, as shown in the example of RC in Table 11. In order to apply our system to task-oriented dialogues, mechanisms to inherit content such as retrieved Web pages to answer users' questions to other workers of the same role, are desired. The evaluation of the efficiency including the overhead of the inheritance is our future work.

Conclusion

In this paper, we propose a novel crowd-powered system to efficiently collect dialogue datasets. In our system, multiple workers create dialogues in an asynchronous manner. This framework frees crowd-workers from time constraints and increase work efficiency. Experiments have proved that our system reduces the time to input a message by 68% and also reduces the cost by 62% while maintaining quality.

One limitation of our method is that it cannot inherit background information to other workers. In order to construct task-oriented dialogues, a mechanism to share notes with other workers is desired. The evaluation of the efficiency including the overhead of the inheritance is our future work.

References

Augello, A., Saccone, G., Gaglio, S., and Pilato, G. 2008. Humorist bot: Bringing Computational Humour in a Chat-Bot System. In Proceedings of CISIS 2008, 703-708.

Belvin, R., Burns, R., and Hein, C. 2001. Development of the HRL Route Navigation Dialogue System. In Proceedings of the First International Conference on Human Language Technology Research, 1-5.

Bennacef, S., Devillers, L., Rosset, S., and Lamel, L. 1996. Dialog in the RAILTEL Telephone-Based System. In Proceedings of ICSLP 1996, pp. 550-553.

Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., and Panovich, K. 2015. Soylent: A Word Processor with a Crowd Inside. Communications of the ACM, 58(8): 85-94.

Bessho, F., Harada, T., and Kuniyoshi, Y. 2012. Dialog System Using Real-Time Crowdsourcing and Twitter Large-Scale Corpus. In Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 227-231.

Cao, Y., Liu, F., Simpson, P., Antieau, L., Bennett, A., Cimino, J. J., Ely J., and Yu, H. 2011. AskHERMES: An Online Question Answering System for Complex Clinical Questions. Journal of Biomedical Informatics, 44(2): 277-288.

Dodge, J., Gane, A., Zhang, X., Bordes, A., Chopra, S., Miller, A., Szlam, A., and Weston, J. 2016. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. In Proceedings of 4th International Conference on Learning Representations.

Fader, A., Zettlemoyer, L., and Etzioni, O. 2014. Open Question Answering over Curated and Extracted Knowledge Bases. In Proceedings of the 20th Conference on Knowledge Discovery and Data Mining (SIGKDD), 1156–1165.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J., Nyberg, E., Prager, J., Schlaefer, N., and Welty, C. 2010. Building Watson: An Overview of the DeepQA Project. AI magazine, 31(3): 59-79.

Hart, S. G., and Staveland, L. E. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. Advances in Psychology, 52: 139-183.

Henderson, M., Thomson, B., and Young, S. J. 2013. Deep Neural Network Approach for the Dialog State Tracking Challenge. In Proceedings of SIGDIAL Conference, 467-471.

Huang, T. H. K., Lasecki, W. S., Azaria, A., and Bigham, J. P. 2016. "Is There Anything Else I Can Help You With?" Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent. In Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing.

Jiang, J., Jeng, W., and He, D. 2013. How do Users Respond to Voice Input Errors?: Lexical and Phonetic Query Reformulation in Voice Search. In Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval, 143-152.

Kiyota, Y., Kurohashi, S., and Kido, F. 2002. Dialog Navigator: A Question Answering System based on Large Text Knowledge

Base. In Proceedings of the 19th International Conference on Computational Linguistics, 1: 1-7.

Krishna, R. A., Hata, K., Chen, S., Kravitz, J., Shamma, D. A., Fei-Fei, L., and Bernstein, M. S. 2016. Embracing Error to Enable Rapid Crowdsourcing. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, 3167-3179.

Lasecki, W. S., Wesley, R., Nichols, J., Kulkarni, A., Allen, J. F., and Bigham, J. P. 2013a. Chorus: A Crowd-Powered Conversational Assistant. In Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology, 151-162.

Lasecki, W. S., Kamar, E., and Bohus, D. 2013b. Conversations in the Crowd: Collecting Data for Task-Oriented Dialog Learning. In Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. 2016a. Deep Reinforcement Learning for Dialogue Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1192-1202.

Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B. 2016b. A Persona-based Neural Conversation Model. In Proceedings of 54th Annual Meeting of the Association for Computational Linguistics, 994–1003.

Lowe, R., Noseworthy, M., Serban, V. I., Angelard-Gontier, N., Bengio, Y., Pineau, J., 2017, Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In Proceedings of the 55th Annual Meeting on Association for Computational Linguistics, 1116-1126.

Mason, W., and Watts, D. J. 2009. Financial Incentives and the "Performance of Crowds". In Proceedings of HCOMP, 77-85.

McTear, M. F., 2002. Spoken Dialogue Technology: Enabling the Conversational User Interface. ACM Computing Surveys (CSUR), 34(1): 90-169.

Raymond, C., and Riccardi, G. 2007. Generative and Discriminative Algorithms for Spoken Language Understanding. In Proceedings of Interspeech, 1605-1608.

Sabelli, A. M., Kanda, T., and Hagita, N. 2011. A Conversational Robot in an Elderly Care Center: An Ethnographic Study. In Proceedings of ACM/IEEE International Conference on Human-Robot Interaction, 37-44.

Sugiyama, H., Meguro, T., Higashinaka, R., and Minami, Y. 2014. Large-Scale Collection and Analysis of Personal Question-Answer Pairs for Conversational Agents. In Proceedings of International Conference on Intelligent Virtual Agents, 420-433.

Tsukahara, H., and Uchiumi, K. 2015. System Utterance Generation by Label Propagation over Association Graph of Words and Utterance Patterns for Open-Domain Dialogue Systems. In Proceedings of 29th Pacific Asia Conference on Language, Information and Computation, 323-331.

Vardoulakis, L., Ring, L., Barry, B., Sidner, C., and Bickmore, T. 2012. Designing Relational Agents as Long Term Social Companions for Older Adults. In Proceedings of Intelligent Virtual Agents. 289-302.

Wang, W. Y., Bohus, D., Kamar, E., and Horvitz, E. 2012. Crowdsourcing the Acquisition of Natural Language Corpora: Methods and Observations. In Proceedings of Spoken Language Technology Workshop (SLT), 73-78.

Weizenbaum, J. 1966. ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine. Communications of the ACM, 9(1): 36-45.