Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 7084–7093
Marseille, 11–16 May 2020

Gamification Platform for Collecting Task-oriented Dialogue Data

Haruna Ogawa **Hitoshi Nishikawa** **Takenobu Tokunaga**
Tokyo Institute of Technology
ogawa.h.ai@m.titech.ac.jp, {hitoshi, take}@c.titech.ac.jp

Hikaru Yokono
Fujitsu Laboratories Ltd.
yokono.hikaru@fujitsu.com

Abstract

Demand for massive language resources is increasing as the data-driven approach has established a leading position in natural language processing. However, creating dialogue corpora is still a difficult task due to the complexity of the human dialogue structure and the diversity of dialogue topics. Though crowdsourcing is majorly used to assemble such data, it presents problems such as less-motivated workers. We propose a platform for collecting task-oriented situated dialogue data by using gamification. Combining a video game with data collection benefits such as motivating workers and cost reduction. Our platform enables data collectors to create their original video game in which they can collect dialogue data of various types of tasks by using the logging function of the platform. Also, the platform provides the annotation function that enables players to annotate their own utterances. The annotation can be gamified as well. We aim at high-quality annotation by introducing such self-annotation method. We implemented a prototype of the proposed platform and conducted a preliminary evaluation to obtain promising results in terms of both dialogue data collection and self-annotation.

Keywords: task-oriented dialogue, situated dialogue, data collection, gamification, dialogue act, annotation, Minecraft®

1. Introduction

Demand for massive language resources is increasing as the data-driven approach using machine learning has established a leading position in natural language processing (NLP). The dialogue research is no exception to this tendency. Due to the complexity of the human dialogue structure and the diversity of dialogue topic, the data-driven approach for dialogue research requires a large amount of data (Ferreira et al., 2015) compared with other tasks like named entity recognition and syntactic analysis. For the efficient collection of dialogue data in various domains, we propose a collection platform using gamification. Particularly we target collecting task-oriented dialogues.

Dialogue can be classified into two types: non-task-oriented and task-oriented dialogues. The former has no specific goal to achieve through dialogue; it is also called chit-chat. Smooth continuation or just passing the time by chatting is the "goal" of this type of dialogue. In contrast, the task-oriented dialogue has a specific goal for dialogue participants to achieve together through dialogue. Recently, conversational assistants such as Apple's Siri and Amazon's Alexa are gaining popularity. In the dialogue with those systems, users usually have a specific goal, e.g. seeking certain information such as a nearby restaurant. Aside from the copyright issue, we could technically collect chit-chat data by crawling chit-chat logs on the Internet. We can also extract dialogue parts from existing texts, e.g. novels and movie scripts. On the other hand, the same methods are difficult to apply to the collection of task-oriented dialogues. Since the goal of the dialogue is given in advance for the task-oriented dialogue, we hardly find the chit-chat logs that are compatible with the goal on the Internet nor in the existing texts. Therefore, individual environments were set up according to the goal for collecting dialogue data in the past.

Recently, crowdsourcing has become a popular method to collect linguistic data, where non-expert anonymous workers do small tasks for a small payment. As the crowdsourcing enables low-cost and rapid data collection, it has been applied to various kinds of linguistic data (Sorokin and Forsyth, 2008; Snow et al., 2008; Lasecki et al., 2013; Saeidi et al., 2018). Dialogue data collection is no exception. However, collecting task-oriented dialogues by using crowdsourcing needs to cope with several issues. The first issue concerns the worker's motivation. The workers are told to complete the task through dialogue. Their motivation comes primarily from its reward instead of their intrinsic will to complete the task. The lack of intrinsic motivation might make the collected dialogue unnatural or irrelevant. Related to the motivation, eliminating low-quality workers would be a problem of crowdsourcing in general. Some workers try to obtain a reward at the lowest effort regardless of the relevance of their responses (Vannella et al., 2014; Radlinski et al., 2019). Secondly, Even though crowdsourcing is cost-effective, i.e. being able to collect a large amount of data at low cost, the cost rises as the amount of data increases. Lastly, this is particularly problematic in collecting human-human dialogues, arranging worker's schedule for making dialogue pairs is a crucial problem. In many tasks that are submitted to the crowdsourcing system, each worker works alone; therefore, they can do the task at any time at any place they want. However, when we collect human-human dialogues, a pair of workers must be online at the same time. Manually scheduling a large number of workers' time slots is impractical. Even if the data collector provides a scheduling system, workers might wait for their partner for a long time or fail to find them (Lasecki et al., 2013).

In this paper, we propose a data collection platform for task-oriented dialogue using gamification. We aim to resolve the crowdsourcing problems mentioned above by introducing gamification into the platform. As the base of gamification, we use Minecraft[1] developed by Mojang. Minecraft provides a framework for chatting in a virtual world. Thus, as

[1] https://www.minecraft.net/

a by-product, we obtain an environment for the situated dialogue where dialogue participants need to consider more diverse contextual information than the text-based dialogue, e.g. spatial relations, deictic reference and gesture.

On this platform, we also propose an annotation method where the participant annotates their own utterances. We call this *self-annotation*. Dialogue data have been usually annotated by a third party who has no relation with the dialogue. There might be cases where the third party has difficulties to understand the speaker intention completely without being a dialogue participant. It will be particularly the case in the situated dialogues where information flows through various modal channels other than a linguistic one. All of the information are not always available or easy to access at the third-party annotation. The self-annotation forces participants to achieve two different kinds of tasks: the primary task to be achieved through dialogue and the annotation to their utterances. This makes a multi-task that leads to participant's high cognitive load (Sweller, 1994). To reduce their cognitive load, we apply gamification to the self-annotation task as well. As there have been few studies on such self-annotation method, we explore its possibility through our platform.

2. Related Work

Several studies have shown the benefit of gamification in data collection and annotation. The ESP game (von Ahn and Dabbish, 2004) applied gamification to image annotation with labels. In the ESP game, players can earn scores by labelling images on the Internet with keywords, such as "brown" and "bag" for a picture of a brown handbag. This interactive system enabled the construction of labelled image resources while people enjoy themselves playing a game, without noticing that they are, in fact, doing an annotation task. They succeeded to gather more than ten thousand players, who produced high-quality image labels and showed the usefulness of gamification in annotation tasks. Vannella et al. (2014) proposed video games for a validation task. They created games for the purpose of word validation in extended WordNet synsets, where they compared the data collected by crowdsourcing with the equivalent collected from the games. They showed that game-based validation leads to higher quality result at a lower cost. These studies adopted gamification for the validation and annotation of existing data, while we aim at collecting new dialogue data by gamification.

There are several attempts at collecting dialogue data by gamification. Asher et al. (2016) implemented a chat system on an online multi-player board game and constructed a multi-party conversation corpus. The chat log was annotated by novice and expert annotators. They labelled dialogue acts and discourse structure in the environment without gamification. Manuvinakurike and DeVault (2015) presented a browser-game that collects spoken dialogue data via crowdsourcing. The goal of this two-player game was to identify a target image among eight pictures displayed on each player's screen. The aspect of dialogue collection was gamified in these games. However, they cannot use their games for other tasks. If they want to collect dialogue data for other tasks, they have to implement a new game from scratch. Our platform provides elementary functions for dialogue collection on top of Minecraft, which means that we can utilise the original functions of Minecraft. This architecture decreases the implementation costs significantly.

3. Gamification

Gamification transforms a task into video games to reward the workers (players) with entertainment. This practice has attracted much attention in various areas until today.

3.1. Benefits

We expect several benefits of gamification for the collection of task-oriented dialogue data. First, the game motivates workers to do the task. Most crowdsourcing tasks, including the creation of task-oriented dialogue data, motivate the workers with payment. Since in these cases the requester gives the worker a goal, the task is not necessarily what workers would like to do. This might lead to unnaturalness in the collected data, or the data might look like scripted. On the other hand, a game with a goal that matches the dialogue task can motivate workers to engage in the task on their own volition (Flatla et al., 2011). It can attract the workers with stories, graphics or by giving them virtual rewards (game scores). Moreover, motivating workers by entertaining them can reduce the number of lazy workers (Vannella et al., 2014). These lazy workers have no intention to do the task if there is no monetary reward for it; moreover, the game can offer mechanics to automatically repel players who do not follow the guidelines. For example, some players might intentionally input invalid data, but the system can filter them by simply making them "game over".

The second benefit is the monetary cost of data collection. The cost of the dialogue data collection will not be proportional to the size of the data. The cost of crowdsourcing with monetary compensation increases depending on the amount of tasks workers finished, requiring large sums of money for the collection of big data. By publishing the game for dialogue data collection on the Internet, we will be able to gather huge amounts of data through game-play interaction. This entails that the cost will not increase proportionally to the amount of collected data. In other words, we would be able to pay workers with "fun" instead of money. The cost will depend not on the number of workers and completed tasks but on designing the game. Creating a game with a lightweight model can be cost-effective compared to crowdsourcing with monetary compensation as the data scale increases.

The third benefit concerns game design. Various video games contain situations that requires the players to do several actions simultaneously. In many cases, such multi-task situations are intentionally created to entertain players. In general, multi-tasking increases cognitive load, which degrades task performance (Sweller, 1994). Su (2016), however, suggested that gamification could reduce the cognitive load in educational applications. Against these backgrounds, we also gamify self-annotation by participants at the same time of collecting dialogue data.

Figure 1: An example of the player's view in Minecraft. Player 829 is holding an item (pickaxe) and standing under the tree. The white nameplate is added afterwards and is not displayed in the game.

3.2. Minecraft

Though gamification has several advantages, there are difficulties in designing a game. The flexibility of the system is crucial to make an adaptive data collection platform, but the creation of a video game of this nature from scratch costs too much. Furthermore, a gamified platform must be attractive enough to make the workers engaged in it. We avoid this problem by using an existing game, Minecraft, a video game developed by Mojang. Minecraft is a sandbox-type game in which players can explore a virtual world made of 3D blocks, and interact with the blocks, items and other players. Figure 1 shows a snapshot of the in-game scene. This game is suitable for a data collection tool for five reasons.

(1) Minecraft does not have an explicit goal by itself; i.e. users can set the game goal freely.

(2) Minecraft includes numerous types of blocks and items that make it possible to create various situations.

(3) A "mod" culture is popular in the game. "Mod" comes from "modification" and stands for an extension of games created by unofficial developers. Using the "mod" mechanism, we can extend the game system to add functions for data collection.

(4) Minecraft is one of the most popular video games in the world, with over 112 million active players. Abundant players make the collection of large amount of data easier.

(5) Minecraft is a multi-player game. Players from different locations can play in the same virtual world through the Internet.

Minecraft has been gathering interest as a platform for various applications including virtual agents (Johnson et al., 2016; Gray et al., 2019), human behaviour analysis (Müller et al., 2015), and dialogue tasks (Dumont et al., 2016; Narayan-Chen et al., 2019). These studies showed the usefulness of Minecraft as a flexible platform.

4. Platform Architecture

4.1. Overview

Our platform provides a virtual world in which a specific task is performed by players through dialogue and functions to collect various information along with dialogue logs. Players access to a Minecraft server run by the data collector via their Minecraft client. All players start the game in the lobby world, where they can apply for the task and wait for other players. By default, the lobby world is an empty place with no objects, but the data collector can decorate it as needed by modifying the lobby world template. The system automatically matches up the players in the waiting queue to make pairs. This pairing function provides a powerful tool for scheduling players time slots as we discussed in Section 1 . When a pair is made, the system creates a task world, where the pair performs the specified task. The task world is created from the task world template that is designed by the data collector for the task. The data collector who is familiar with Minecraft can easily build the task world template by assembling the Minecraft blocks. They are not required special knowledge of programming to create the world templates. Programming is needed only when they modify the platform to add their original functions. The system then sends the player pair to the dedicated task world where they can start working on the task.

There can be multiple task worlds simultaneously and they are independent of each other, i.e. every event in a task world is recorded independently from other task worlds. Once the players finish the task, the system will send them back to the lobby world and destroys their task world.

An overview of our platform is shown in Figure 2. The platform consists of one Minecraft server run by the data collector and multiple Minecraft clients used by the players. Both server and clients have extended functions, which are realised by three modules: a task world manager, an annotation tool and a logger. The task world manager, named TaskWorldMod, is used only on the server. The annotation tool and the logger are implemented together as ChatAnnotatorMod, which is used in both server and clients. The detail of this function will be described in Section 5 .

Each module is implemented as a Minecraft Forge mod. Minecraft Forge is an open-source API for extending Minecraft, which is a popular way to modify the game (Gupta and Gupta, 2015). Forge enables us to extend the game system easily with Java codes, contributing to the thriftiness of mod culture in the Minecraft community. We are using version 1.12.2 of the API, which requires the same version of Minecraft.

4.2. TaskWorldMod

TaskWorldMod is an implementation of the task world manager for the server. This mod manages a pair of players and their task world. It makes player pairs, creates task worlds, and sends the players to their task world. The player needs to apply to the system first in order to receive a partner. The system makes a pair in response to the player's demand. The task world is created from the task world template that should be prepared by the data collector. Designing the task world template is the primary work

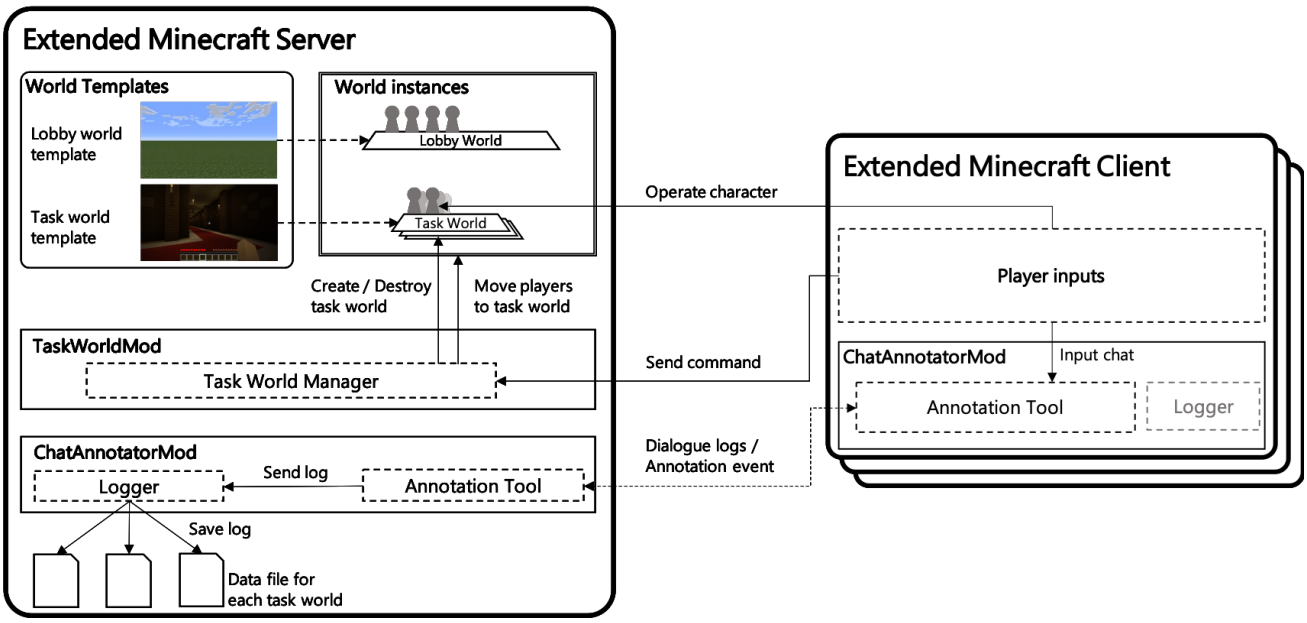

Figure 2: An overview of the platform architecture

for the data collector, as the nature of the collected dialogues depends on the types of task worlds designed. Our platform enables the data collector to implement a game by defining a world by using Minecraft intrinsic editing interface. The constructed world is converted to the task world template. In comparison to creating a game from scratch, our platform saves tremendous work. For instance, the example described later in Section 6 were mostly made by using Minecraft intrinsic functions except for the map and the scoring mechanism that needed extra mods. As we mentioned, however, creating new mods is popular in the Minecraft community, and the documents and tools for mod building is abundant.

Since the individual player pairs can perform the task in their world independently from and simultaneously with other pairs, the data collector can assemble dialogue data efficiently. Moreover, since TaskWorldMod handles making players pairs automatically, the data collector does not have to worry about arranging the players' schedules manually.

5. Annotation

5.1. Self-annotation by Participant

In general, dialogue corpus annotation has been done by third parties not involved in the dialogue. In many cases, dialogue collection and annotation are distinct entities, i.e. the creator of the dialogue data might not always expect the data to be annotated in the first place (Serban et al., 2015). However, these third-party annotators cannot always fully understand the speaker intention. They have to guess the context or situation of the dialogue and possibly mis-annotate due to the misunderstanding. Conversely, if the speaker annotates their utterance by themselves, there should be no misunderstanding. This is a definite advantage of self-annotation by speakers. Also, the monetary cost and time for creating an annotated dialogue corpus can be reduced by performing the collection and annotation at the same time. From these substantial benefits, we claim that self-annotation is worth considering as an annotation method. Despite the advantages of this method, there are some disadvantages as well. As it is a multi-task where the participants have to annotate while chatting and performing the given specific task, the participant's cognitive load will increase. A high cognitive load reduces the annotator's motivation and might degrade their annotation performance. Nonetheless, as we described in Section 3.1., gamification might remedy this disadvantage. To validate the effectiveness of self-annotation, we added a self-annotation function to our platform.

5.2. Target annotation: Dialogue Act

The target of our annotation is the dialogue act of utterances. The dialogue act is a representation of the speaker's intention for each utterance in dialogues. It is considered as primary information for dialogue structure and is common as an annotation label for such structure (Core and Allen, 1997; Stolcke et al., 2000). Compared to other shallow information such as dependency and POS, the dialogue act is more dependent on the annotator's interpretation. Hence, self-annotation should be more suitable for annotating the correct dialogue acts than the shallow information mentioned above. On our platform, we created a simple label set based on the ISO 24617-2 standard (Bunt et al., 2012), which is shown in Table 1.

The ISO standard label set is exhaustive and multi-dimensional. We simplify it to eight labels for introducing annotation into the game. It is well known that human's short-term memory has a size of around seven chunks (Miller, 1956). Considering this fact, we reduced the number of dialogue act labels to eight. We tried to reduce the player's cognitive load at the cost of granularity of dialogue act categories.

Dialogue Act	Description
QUESTION	An utterance to ask or confirm something to the partner.
REQUEST	An utterance to make the partner do something, or to show the speaker's action. Suggestion, request, instruction, offer, and so forth.
GREETING	An utterance to make the communication smooth. Includes self-introduction, thanking or apology.
YES	An utterance to tell agreement or confirmation.
NO	An utterance to tell disagreement or disconfirmation.
CONVEY	An utterance to tell information to the partner, including an answer to a question.
EXCLAMATION	An utterance to express emotion, such as "hooray" or "uhh".
CORRECT	An utterance to correct a mistake, such as a misspelling, in the previous utterances.

Table 1: The annotation label set of dialogue acts

5.3. ChatAnnotatorMod

Figure 3: Snapshot of the annotation tool. The bottom buttons are labelled with dialogue acts. Unannotated utterances from the partner are coloured with yellow and underlined. The white nameplates are added afterwards and are not displayed in the game.

ChatAnnotatiorMod implements both the logger and annotation tool functions as described in Section 4 . The annotation tool extends the Minecraft chatting system and enables us to annotate each utterance. The logger records utterances and annotation event logs into the files. Due to a technical reason, we implemented the logger in the same mod as the annotation tool; the logger is used only in the server, while the annotation tool is used in both server and client. In the client, it works as a GUI system. A toolbar is displayed above the native Minecraft chat interface, as shown in Figure 3. When making an utterance, the speaker needs to select a dialogue act from the buttons in the toolbar. The dialogue act selection is obligatory, i.e. the speaker can not make their utterance without the dialogue act selection. Another player (hearer) can annotate the speaker's utterance in the dialogue history. When the hearer clicks the unannotated speaker's utterance in the dialogue history, the array of the dialogue act buttons pops up above the selected utterance. They can choose one of the dialogue acts from the buttons. The annotated utterance changes its appearance from a yellow-coloured and underlined string to a normal string in the dialogue history. Unlike the speaker, the hearer's annotation is optional. This allows the hearer the freedom of choosing to respond to the speaker promptly or to confirm the speaker's intention through annotation. Each dialogue act button contains both an icon and a short label so that the players easily recognise the choice in order to reduce their cognitive loads (Mayer and Moreno, 2003). The dialogue act buttons are hard-coded in the current version, but we are planning to make it customisable so that various kinds of annotation other than the dialogue act are available.

On the server, the annotation tool checks the annotation status of each utterance and shares it with the client so that unannotated utterances are highlighted with yellow colour and underline. It also issues an annotation event for each time an utterance is annotated, to let other mods can cause an action such as giving scores to the players. In addition, the annotation tool sends the utterance and annotation event to the logger. The logger records the utterances and event logs in real-time to the files in the server. Also, at the time of the task world closing, it outputs the final result of annotation constructed from the log to a file, which includes the content of the utterances, speaker's annotation, hearer's annotation, speaker's id with a timestamp. Dialogues and event logs in individual task worlds are independently saved in different files.

6. Example: Mansion Task

As an example of the task to be performed on our platform, we designed *Mansion Task* for the collection of dialogue data. The purpose of this example task is to collect the cooperative task-oriented situated dialogue with dialogue act annotation at the same time of testing our proposed platform. The task was inspired by Map Task (Anderson et al., 1991), which is a cooperative task that involves two participants: an instruction giver and an instruction follower. Each participant has their own map, which is almost identical but has some discrepancy. For instance, the maps have different labels for the same landmark, and landmarks appear only in one of the maps. The most significant difference between the maps is that a route is marked only in the giver's map. The goal of the task is to replicate the giver's route in the follower's map only through dialogue.

Map Task is an asymmetric task where participants have different roles. In an asymmetric task, a participant with abundant information tells another participant what to do. There have been dialogue corpora collected through asymmetric tasks in the past. The Fruit Cart Corpus (Aist et al., 2012) was collected by a two-person task performed on a PC, where one participant (director) instructs another participant (actor) to place objects on a displayed map and to change their colour. Both participants share the displayed map. The ArtWalk Task (Liu et al., 2016) assigns two par-

ticipants different roles: a director and a follower, where the follower walks around the real town to find the public art following the director's instructions via Skype. In these asymmetric tasks, the number of participant's utterances is often unbalanced due to their different roles. The followers tend to speak less, and their utterances tend to be shorter than the givers (Anderson et al., 1991; Tokunaga et al., 2010).

In the symmetric task where the participants are not assigned a specific role a priori, i.e. they are equal partners, the imbalance of the utterance numbers between participants is less likely to occur than in the asymmetric task. For instance, He et al. (2017) conducted the Mutual-Friends task, where two participants are given a different list of "friends" and have to find a common term of the list through dialogue. Though the given lists are different, the participants communicate with each other without any information skew and the difference in their role.

Figure 4: The pair of maps of Mansion Task

Figure 5: Screenshot of Mansion Task

Our Mansion Task is a symmetric task based on Map Task. Like Map Task, the players in this game have different maps that cannot be seen from the partner. The Mansion Task map shows the room layout in a mansion, as shown in Figure 4. The maps also indicate names of the rooms (in black) and objects in the rooms (in blue). The indicated information has no discrepancy between the maps, but some information is missing from one of the maps. Therefore, two players need to communicate with each other to recover the full information of the mansion. The goal of the task is reaching the specified place that is indicated in both maps. As no route is explicitly indicated in the maps, the players need to find a route leading to the goal place through communicating each other and moving together in the virtual space of the mansion. On the route towards the goal there are several obstacles, such as locked doors. To open the door, they might need to find a key in a treasure box in elsewhere. The players are required to achieve those sub-goals to achieve the primary goal. We implemented Mansion Task on the proposed platform. Figure 5 shows a screenshot of the player's view of Mansion Task.

On top of our proposed platform, we implemented a scoring system of the task, ScoreMod, to enhance the player's motivation. When both players annotate an utterance, they earn points based on the result of the annotation. They get points no matter what label they choose, but if both players select the same label for the utterance, they earn five times more points. This means that we reward bonus points when the hearer correctly interprets and annotates the speaker's dialogue act. We expect this scoring mechanism motivates the players to annotate seriously. Our task world template for Mansion Task does not include ScoreMod, which needs to be implemented separately. As we discussed in Section 4., implementing mods is not so difficult. As a further extension, we are constructing a ranking system where players can compare their score against others, to arouse players' competitive motivation.

7. Experiments

To evaluate the platform, we conducted two small experiments: one in-house and one external.

7.1. In-house Test

Setting

We tested the prototype of the platform with eight participants (four pairs) in-house. They are graduate and undergraduate students from the same research group of the authors; their major is computer science and artificial intelligence. Their mother tongue is Japanese and all dialogue were in Japanese. In addition to checking whether the implemented functions work well, we aimed at evaluating the effectiveness of gamification by comparing the gamified and ungamified version of the same task. Among four pairs, two pairs performed the gamified version of Mansion Task that is described in Section 6 , and the other two pairs performed a plain (ungamified) Mansion Task. In the plain version, the players have paper-printed maps, and they draw the resultant route on the map. We used the same chat tool as the gamified version, but with an empty black world where the players see nothing, i.e. they are not-situated. The plain version does not equipped the scoring system either. We recorded the time spent to solve the task, the number of utterances and the match rate of the players' annotations.

Result

Table 2 outlines the collected dialogues. The experiment size is too small to conduct a precise quantitative evaluation. From the viewpoint of dialogue collection, there

Pair	Time spent	# of utterances	Annotation match rate	Breakdown of annotations		
				Matched	Unmatched	Null
G1	24m53s	44	0.886	39	2	3
G2	59m22s	83	0.747	62	20	1
P1	37m07s	62	0.694	43	15	4
P2	26m53s	26	0.577	15	6	5

Table 2: The result of in-house pilot test (G1 and G2: gamified version, P1 and P2: plain version, "Null" means the case where the hearer selected no label.)

is no significant difference in dialogue time and utterance numbers between the gamified and plain versions. However, the utterance contents are different between them. The utterances in the plain version contain many macro-perspective explanations of the situation, e.g. "The black key is in the treasure box in the room at the end that you can reach by turning to the left from the entrance.". In contrast, the utterances in the gamified version contain more micro-perspective and situated explanations such as "The black key means the one behind the flower we saw earlier, right?". This difference is possibly caused by the situated environment created by the game.

Concerning the annotation quality, the gamified version shows the higher annotation match rate and less null annotation, i.e. no annotation by the hearer. Although the result is not decisive due to the small size of samples, this implies the effectiveness of the gamification to motivate the players. In the gamified platform, a score is given to a pair each time two players annotate the utterances to motivate them as described in Section 6 . Since the given score becomes higher if the pair choose the same label to the same utterance, it suggests that the gamified system affected the players to annotate seriously and resulted in higher annotation match rate.

7.2. External Test

Setting

We also conducted a small experiment on ten external players (five pairs) to test the platform and considered the feasibility of self-annotation. They are all NLP researchers including students, who are not necessarily familiar with the dialogue field. They volunteered for our call for participation. The players performed the gamified Mansion Task with our platform, where we collected the dialogue logs and the annotations by the players. Besides, two annotators who have experience of dialogue act tagging annotated the collected dialogues. This means that each utterance was annotated by four different annotators: two players (speaker and hearer) of the dialogue and two experienced annotators.

Result

Table 3 show the stats of the collected dialogues. The table shows no significant difference from that of the in-house experiment (Table 2). Among five dialogues, Dialogue 3 seems be an outlier in terms of its short dialogue time, a low annotation match rate and many Null annotations. The Dialogue 3 pair stopped the game halfway through and failed to achieve the goal.

Table 4 shows pair-wise Cohen's kappa (Cohen, 1960) between annotators, indicating that the agreement between the experienced annotators is relatively high compared with the agreement of the player-involved pairs. Each player plays two different roles in the dialogue: a speaker and a hearer. As the speaker chooses their dialogue act when making an utterance, we assume this label must be correct. Therefore we calculate the annotation accuracy by considering the speaker's label is gold. Table 5 shows the annotation accuracy indicating that the accuracy of experienced annotators is not so high. This means that even though the experienced annotator agreed on a decision, the agreed decision is not always correct with respect to the speaker's intention.

This result suggests that there might be difficult cases for the third-party annotators to understand the speaker's intention correctly, even though they have experience of dialogue act tagging. For instance, the two players annotated an utterance "There seems to be a blue-locked treasure box in the room named *student room*." with the REQUEST tag, but both annotators did with the CONVEY tag. The speaker made this utterance to propose their partner to go to the student room, and the hearer recognised their intention correctly, but both experienced annotators interpreted it as just conveying the information. The contextual information like a situation and atmosphere is necessary to understand the intention of the utterance correctly, but it is difficult for the third-party annotators to capture such information.

Table 6 shows the comparison of the selected labels for each utterance by the players and the experienced annotators in five dialogues. Since there are two players and two experienced annotators in each dialogue, four combinations of pair exists per dialogue and results of all pairs are shown in this table. The table indicates large proportion of annotation disagreement between the annotators involves the CONvey tag. In particular, confusing the CONvey and REQuest tags is prominent. As in the previous example, it may be difficult to tell these tags apart with limited information in utterances. This suggests that the hearer in this particular example has succeeded to recognise the speaker's purpose because of the shared situation on the platform.

However, the overall accuracy of the players looks lower than that of the experienced annotators in Table 5. Further investigation revealed that two hearers gave up a significant number of annotation during the dialogue, i.e. 21 out of 22 (95%) utterances and 16 out of 39 (41%) utterances. Note that annotating utterances is not obligatory for hearers. Furthermore, one of these players tends to give up the anno-

Dialogue	Time spent	# of utterances	Annotation match rate	Breakdown of annotations		
				Matched	Unmatched	Null
1	32m26s	90	0.800	72	17	1
2	41m07s	132	0.697	92	38	2
3	21m35s	51	0.353	18	8	25
4	28m14s	69	0.565	39	12	18
5	32m40s	44	0.750	33	9	2

Table 3: The result of external preliminary test ("Null" means the case where the hearer selected no label.)

Dialogue		P1	P2	A1
1	P2	0.752		
	A1	0.725	0.697	
	A2	0.767	0.683	0.889
2	P2	0.530		
	A1	0.627	0.677	
	A2	0.627	0.579	0.668
3	P2	0.140		
	A1	0.159	0.468	
	A2	0.175	0.552	0.764
4	P2	0.318		
	A1	0.362	0.512	
	A2	0.363	0.491	0.901
5	P2	0.595		
	A1	0.574	0.710	
	A2	0.574	0.645	0.908

Table 4: Cohen's kappa between annotators
(P1 and P2: players, A1 and A2: experienced annotators)

Dialogue	H	A1	A2
1	0.800	0.756	0.789
2	0.697	0.750	0.750
3	0.353	0.588	0.627
4	0.565	0.667	0.667
5	0.750	0.773	0.773

Table 5: Annotation accuracy
(H: hearer, A1 and A2: experienced annotators)

tation in the latter part of the dialogue. They might have focused on achieving the primary task goal, i.e. reaching the goal place, and have had less interest in the annotation. We need to refine the scoring mechanism and introduce a ranking system to keep the players motivated to annotate utterances.

Table 5 counts unannotated utterances as incorrect cases. We considered only utterances annotated by all four annotators and broke down the table into player-basis tables (Table 7). The boldface value indicates the best accuracy in the row. The tables suggest the diversity of individual player quality in terms of dialogue act annotation. Also, we find that the hearer performed the annotation slightly better than the experienced annotator.

8. Conclusion

In this paper, we proposed a platform for situated task-oriented dialogue data collection using gamification. Constructing a large corpus of human dialogues is challenging due to the difficulties in collecting participants and in pairing them. We tackled these problems by using gamification. We constructed a dialogue data collection platform based on Minecraft, which facilitates the implementation of a versatile gaming environment. As Minecraft does not have a game goal by itself, we can design various situated tasks on the platform.

Also, we proposed self-annotation, a novel annotation method for the dialogue data that requires the players to annotate their utterance. We assume that the speaker can annotate their utterance correctly in principle, while their partner or third-party annotators might not as they have to "guess" the speaker's intention. They might misinterpret the speaker's utterance. Since annotating while chatting is a resource-intensive task, we introduce gamification again to reduce the cognitive load of the player and motivate them to do annotate seriously.

We evaluated our proposed platform through small scale experiments. We designed a routing task named "Mansion Task" which is similar to Map Task. Through this game, we collected dialogue data and analysed it. The results indicated that the annotation quality of players is comparable to the experienced annotators. This is due to the fact that the players can refers to the contextual situation during the course of the dialogue, while the off-line third-party annotators can not.

As we have already described, our future work includes the following research items.

- Implementation of customisable annotation tool
- Ranking system for visualising players' results
- Large scale evaluation in the real world

After further refinement, we plan to publish the platform software to the public.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP19H04167, Grant-in-Aid for Scientific Research (B).

		Experienced Annotators								
		QUE	REQ	GRE	YES	NO	CON	EXC	COR	No label
Players	QUE	215	13	0	0	0	19	10	0	0
	REQ	17	167	2	1	0	109	2	0	0
	GRE	1	1	103	15	0	20	11	0	0
	YES	2	7	24	205	0	45	13	0	0
	NO	0	0	0	0	11	7	0	0	0
	CON	36	137	20	26	5	1001	40	6	0
	EXC	10	7	17	13	0	42	171	0	0
	COR	0	0	0	0	0	1	0	6	0
	No label	16	15	16	8	0	62	27	0	0

Table 6: Comparison of selected labels in each pair of the player and the experienced annotator in five dialogues combined (The first three letters for the dialogue acts, and "No label" for utterance the annotator did not select any dialogue act)

Dialogue	H=P1	A1	A2
1	**0.870**	0.804	0.826
2	0.792	**0.849**	0.792
3	0.000	0.000	0.000
4	**0.696**	0.609	0.609
5	0.750	**0.813**	**0.813**
Dialogue	H=P2	A1	A2
1	0.744	0.721	**0.767**
2	0.649	0.688	**0.714**
3	**0.720**	0.520	0.600
4	**0.821**	0.679	0.643
5	**0.808**	0.731	0.731

Table 7: Broken-down annotation accuracy
(H: hearer, A1 and A2: experienced annotators)

References

Aist, G., Campana, E., Allen, J., Swift, M., and Tanenhaus, M. K. (2012). Fruit carts: A domain and corpus for research in dialogue systems and psycholinguistics. *Computational Linguistics*, 38(3):469–478.

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. S., and Weinert, R. (1991). The HCRC map task corpus. *Language and Speech*, 34(4):351–366.

Asher, N., Hunter, J., Morey, M., Farah, B., and Afantenos, S. (2016). Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Bunt, H., Alexandersson, J., Choe, J.-W., Fang, A. C., Hasida, K., Petukhova, V., Popescu-Belis, A., and Traum, D. (2012). ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 430–437, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Core, M. G. and Allen, J. F. (1997). Coding dialogs with the damsl annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Cambridge, MA, November.

Dumont, C., Tian, R., and Inui, K. (2016). Question-answering with logic specific to video games. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4637–4643, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Ferreira, E., Jabaian, B., and Lefèvre, F. (2015). Zero-shot semantic parser for spoken language understanding. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 1403–1407.

Flatla, D., Gutwin, C., Nacke, L., Bateman, S., and Mandryk, R. (2011). Calibration games: Making calibration tasks enjoyable by adding motivating game elements. In *UIST'11 - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 403–412, 10.

Gray, J., Srinet, K., Jernite, Y., Yu, H., Chen, Z., Guo, D., Goyal, S., Zitnick, C. L., and Szlam, A. (2019). Craftassist: A framework for dialogue-enabled interactive agents. *ArXiv*, abs/1907.08584.

Gupta, A. and Gupta, A. (2015). *Minecraft Modding with Forge: A Family-Friendly Guide to Building Fun Mods in Java*. O'Reilly Media, Sebastopol, CA, April.

He, H., Balakrishnan, A., Eric, M., and Liang, P. (2017). Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1766–1776, Vancouver, Canada, July. Association for Computational Linguistics.

Johnson, M., Hofmann, K., Hutton, T., and Bignell, D. (2016). The malmo platform for artificial intelligence experimentation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 4246–4247. AAAI Press.

Lasecki, W., Kamar, E., and Bohus, D. (2013). Conversations in the crowd: Collecting data for task-oriented dialog learning. In *In Proceedings of the Human Computation Workshop on Scaling Speech and Language Understanding and Dialog through Crowdsourcing at HCOMP 2013.*, January.

Liu, K., Tree, J. F., and Walker, M. (2016). Coordinating communication in the wild: The artwalk dialogue corpus of pedestrian navigation and mobile referential communication. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3159–3166, Portorož, Slovenia, May. European Language Resources Association (ELRA).

Manuvinakurike, R. and DeVault, D. (2015). Pair me up: A web framework for crowd-sourced spoken dialogue collection. In G.G. Lee, et al., editors, *Natural Language Dialog Systems and Intelligent Assistants*, pages 189–201. Springer International Publishing, Cham.

Mayer, R. E. and Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, 38(1):43–52.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.

Müller, S., Kapadia, M., Frey, S., Klingler, S., Mann, R. P., Solenthaler, B., Sumner, R. W., and Gross, M. H. (2015). Statistical analysis of player behavior in minecraft. In *Proceedings of the 10th International Conference on the Foundations of Digital Games, (FDG'15)*, June.

Narayan-Chen, A., Jayannavar, P., and Hockenmaier, J. (2019). Collaborative dialogue in Minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy, July. Association for Computational Linguistics.

Radlinski, F., Balog, K., Byrne, B., and Krishnamoorthi, K. (2019). Coached conversational preference elicitation: A case study in understanding movie preferences. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue*.

Saeidi, M., Bartolo, M., Lewis, P., Singh, S., Rocktäschel, T., Sheldon, M., Bouchard, G., and Riedel, S. (2018). Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium, October-November. Association for Computational Linguistics.

Serban, I., Lowe, R., Henderson, P., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems: The journal version. *ArXiv*, abs/1512.05742.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October. Association for Computational Linguistics.

Sorokin, A. and Forsyth, D. (2008). Utility data annotation with amazon mechanical turk. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, June.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C. V., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Su, C.-H. (2016). The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study. *Multimedia Tools and Applications*, 75(16):10013–10036, August.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295 – 312.

Tokunaga, T., Iida, R., Yasuhara, M., Terai, A., Morris, D., and Belz, A. (2010). Construction of bilingual multimodal corpora of referring expressions in collaborative problem solving. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 38–46, Beijing, China, August. Coling 2010 Organizing Committee.

Vannella, D., Jurgens, D., Scarfini, D., Toscani, D., and Navigli, R. (2014). Validating and extending semantic knowledge bases using video games with a purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1304, Baltimore, Maryland, June. Association for Computational Linguistics.

von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, NY, USA. ACM.