Noonchi-Baekdan: Real-time Crowdsourcing for Contextual Meaning of Chats over Messengers

Sohee Kim
KAIST Graduate School of
Culture Technology
Daejeon, Republic of Korea
sohee_kim@kaist.ac.kr

Ina Ryu
KAIST School of Computing
Daejeon, Republic of Korea
ryu_ina@kaist.ac.kr

Hyunwoo Kim
KAIST School of Computing
Daejeon, Republic of Korea
khw0726@kaist.ac.kr

ABSTRACT

Mobile instant messaging is one of the most common communication channel. However, as texts lack non-verbal cues for contextual information, mobile messaging is vulnerable to misunderstanding. As a solution, we propose Noonchi-Baekdan, a real-time crowdsourcing platform to help understand the true meaning of the conversation, powered by crowd gathered for fun. Deployment study showed that requesters were satisfied with the answers, and crowd enjoyed their tasks. Keeping the crowd engaged to the platform remains as a further goal.

INTRODUCTION

In 2017, the most popular instant messengers are mostly text-based messengers. However, text-based messages are inherently limited as they lack non-verbal contextual information cues. Despite ways to supplement texts, such as sound messaging and emoticons, are introduced, it is still not enough to precisely recognize other's intentions compare to the non-verbal cues from face-to-face conversations. As a result, it is often difficult to grasp the context when talking over messenger. As a proof, there are many cases where people upload their text-based conversations on online communities to ask for the context or the intention of the other person when they struggle to catch the meaning of them.

Language, especially natural language, is formed through the implicit agreement of the members of society using it over many years. Therefore, users who have been using the same language for a long time are most suitable for pointing out the context of text-based conversation. We propose Noonchi-Baekdan, a platform to ask for contextual meaning during the conversation with crowdsourcing. We designed a messenger-based question posting process to ask question easier during conversation. For the crowd, we used innate desire of humans wanting their voice heard. [1] Plus, we implemented some game elements to give feedback and make them remain longer in the system by making the task enjoyable.

Deployment study with mid-20 people showed that answers from crowdsourcing was useful. The crowd enjoyed the answering process, serving as a motivation for participation. However, further design refinements would be desirable for higher participation and prompt answers.

RELATED WORKS

According to Social Information Processing Theory developed by Joseph Walthers, computer-mediated communication (CMC) lacks cues apparent in face-to-face communication. [2] Although online communication opens up new potential relationships, social cues are deprived in it. Although emoticons and short voice recordings are introduced to compensate for this, it is still impossible for the user to catch the intention when the other person is using a distortion such as irony.

There is a similar service analyzing patterns of text-based conversations to reveal intimacy and favorability. It is different from the suggested service as it uses algorithm rather than human crowds. [3] On the other hand, Noonchi-Baekdan focuses more on the emotional support and attention people expects from other people.

NOONCHI-BAEKDAN

Noonchi-Baekdan targets to help people understand the contextual meaning during the chat. Our platform consists of requester's part and crowd's part. Requester captures the chat to ask, and posts it. Crowd see the question, and answer it. Then, the requester can see the aggregated answer. Crowd see their activity statistics as feedback.

Requester Workflow

To ask the question, the user should capture the chat to show problematic situation, as they do when they ask online communities about the true meaning of the conversation. In this way, the user could provide the conversation context to the crowd in a simple way.

Upload Image Using Kakaotalk PlusFriend

To post a new question, the user would send the captured conversation to our KakaoTalk PlusFriend. Then, the links to our web interface for writing the question, and checking the answers is sent. By using PlusFriend, users can easily make questions and get answers within the messenger.

Questioning Process on "Noonchi-Baekdan" Web Interface

After the user upload an image, she can start questioning process. First, the user select the category of the question among five choices: "Love", "Work", "Friend", "Family", and "Etc". Categorizing is for recruiting the crowd with specific interest or expertise, leading to reliable answers. Then, the user marks exact part of the chat to be asked, and writes down question. We limit the question to yes/no

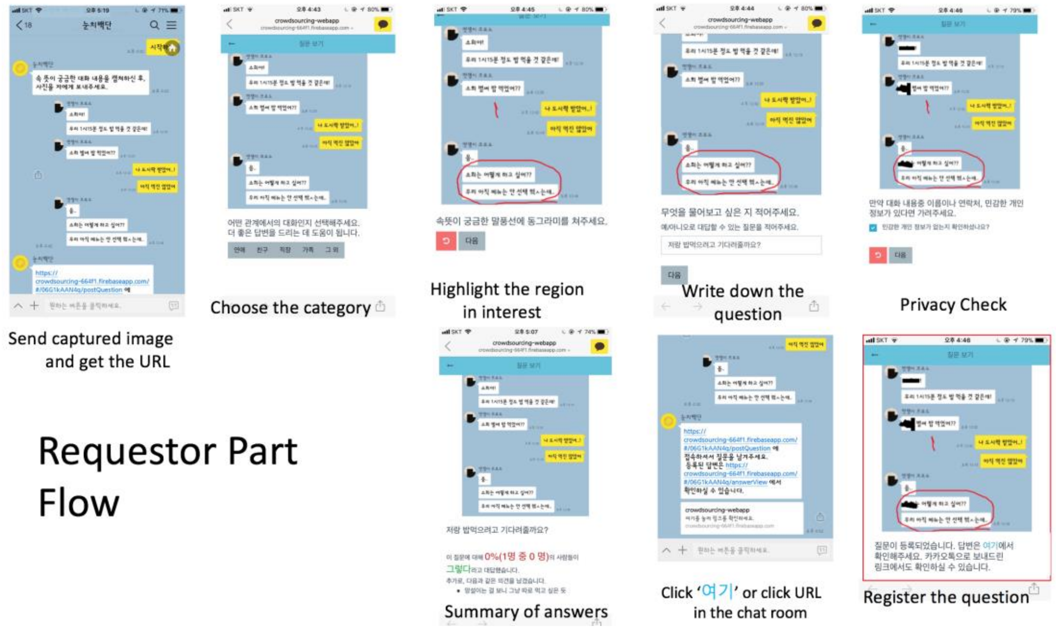

Figure 1 Workflow of Noonchi-Baekdan for requester is presented in clockwise order. Requester would capture the chat, upload the image via PlusFriend, write the question, black out private parts, and read the answers.

questions to make it easier to answer, and to get the overview of the answer easily. Finally, the user can hide some parts of the conversation with privacy issues.

Summary of Answers

The requester checks the crowd responses from the link to the web interface, given by the PlusFriend. Percentage of positive answers and the open-ended answers from the crowd are shown.

Crowd Workflow

Register as Crowd

When a user gets to NoonchiBaekdan for, the first thing to do is sign-up. Here, the crowd chooses categories of interest, and selects whether allow push notification or not. The system would notify interested crowd when a new question is posted. Then, the crowd could get to the answering page directly by clicking the notification. From this feature, we expect the crowd could keep engaged to the platform, so real-time answering could be possible.

Answering Process

Crowd could answer the questions via push notifications, or unanswered question list available on the web interface. Crowd could enjoy the process by looking at other's conversation, and having their voice heard by the requester. To make participation easier, we designed the answering task as simple as possible, by giving only yes/no options and a textbox for short, optional comment.

Self-Feedback via Statistics

Statistics page was designed to give feedback and reward to the crowd to keep them using the platform. Some of the data displayed are the rate of matching with the majority opinion, the number of answers for each category, and acquired badges. Badges were given based on the number of posted answers for each member of the crowd. We expect that badges encourage crowd participation.

EVALUATION

Deployment Results

For deployment study, we advertised our service to people from early 20s to early 30s. The deployment study was conducted for a week, from December 1 to December 8. During the deployment, 71 users joined the platform. 46 questions were posted on the platform (including 18 questions by our team), and total 645 answers were posted to the platform (including 134 answers by our team).

Number of questions by each category is shown in Figure 3. "Friend" category (47.8%) was the most popular, followed by "Love" (17.4%). For the number of answers, the trend was similar, but the proportion of "Love" category increased to 23.1%. Since most of the participants were college students, the categories of questions and answers would be biased to their interests.

The number of answers by each day is shown in Figure 4. Three peaks corresponds to each day we promoted our service. Figure 5 shows the number of users by the number of their answers. 7 people left no answers, and only 21 users answered more than 10 questions. On average, each user answered 9.1 questions. Excluding users with 0 answers, the average number of answers by each user was 10.1. These results show that the crowd participation was not continuous.

User Survey

We requested a survey on the user experience of the service while promoting. The survey measured usefulness and usability of the service, both for the requester's side and the crowd's side. 13 people answered our survey. In overall, 10

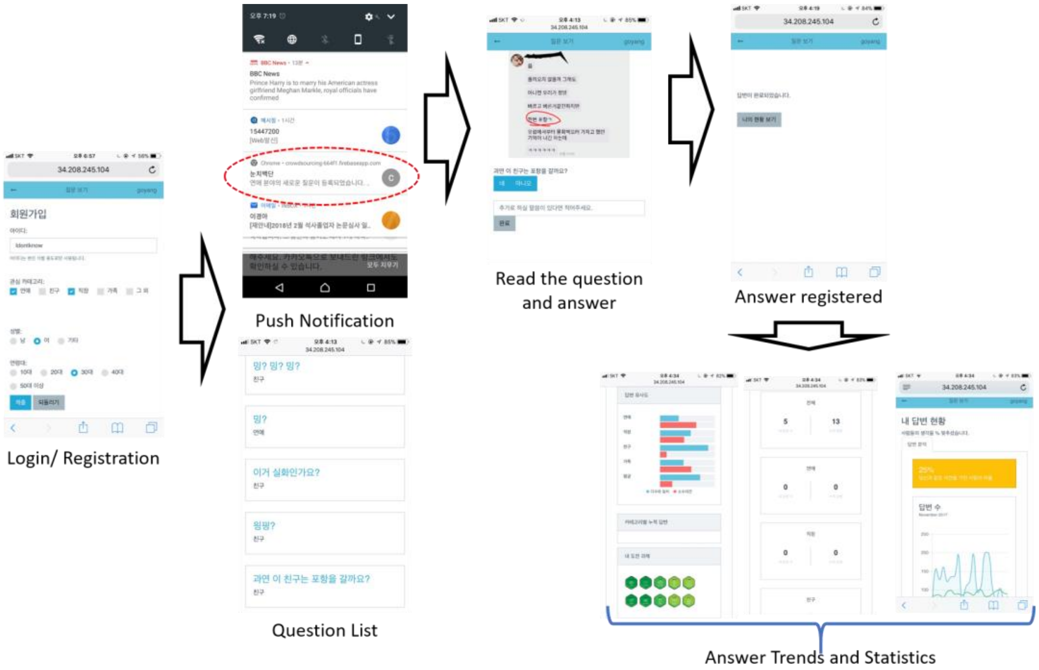

Figure 2 Workflow of Noonchi-Baekdan for crowd is presented in clockwise order. Crowd would see questions from the question list or get notified by push, answer the question, and see their statistics and the trends.

out of 13 people answered that they would want to ask questions on the service, and 10 out of 13 people said they would want to answer questions on the platform.

Answers were useful

10 out of 13 people said that statistics for yes/no answers were useful for them. One respondent said that, "most of the questions did not have a clear answer, so statistical numbers helped". Another respondent mentioned that listening to people with different opinion help her to get the meaning of the chat. On the other hand, a respondent pointed out that yes/no answers were not helpful because the number of answers was too small.

Respondents preferred open-ended answers. All respondent said open-ended answers were helpful, except for one respondent without any experience with open-ended answers. One respondent said, "yes/no answers helped me put my mind at ease, but open-ended answers helped me understand detailed thoughts of others by showing the reasoning behind the opinion". However, the only seven respondents said that the received answers were reliable. Also, only two respondents mentioned that the answers were prompt enough.

On the other hand, when the respondents were answering, only 7 respondents were sure that their answers would help the requester. One respondent said that she was not sure whether her answers were useful or not because she could not get any feedback from the requester. Also, there were three people said that they were answering questions just for fun, without expectation of being helpful for the requester.

Answering questions was fun

All respondents mentioned that answering questions was enjoyable. One respondent said, "It was fun to just read other's chats. Also, there were a lot of fun questions, somewhat like giving titles to the images. To those questions, I tend to post funny answers without any burden." With the similar point of view, one respondent mentioned that seeing what others answered would be more enjoyable. Furthermore, 8 out of 13 people said that statistics made them engage actively. These results suggest that using fun as a motivation for the crowd could be a possible approach.

Opinions on Task Design

For the requester workflow, 9 out of 13 respondents said that it was easy to make questions answered in yes/no manner. However, seven people answered that initiating questions by sending a capture of the chat was convenient, while five people did not. One of the five wrote, "It was bothersome to get to the link and post the question". This result show that there is a large room for the design to be improved.

For the crowd workflow, many respondents (4 out of 6) who used push notifications said that the time from receiving a notification to answer the question was short enough. 9 out of 13 respondents said that it was easy to answer questions in yes/no format. However, one mention that having a "not sure" as an answer would be useful. Two respondents wanted a detailed feedback on their response, by seeing others' answers or feedback of the requester. Those opinions gives design guidelines for improvements.

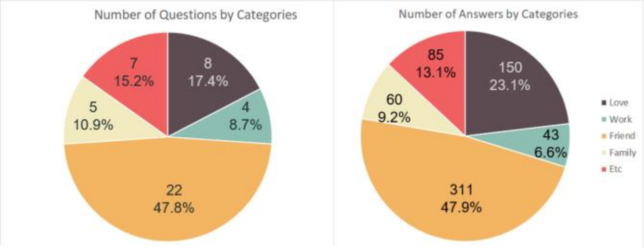

Figure 3 Number of questions and answers by their categories. "Friends" and "Love" categories were popular.

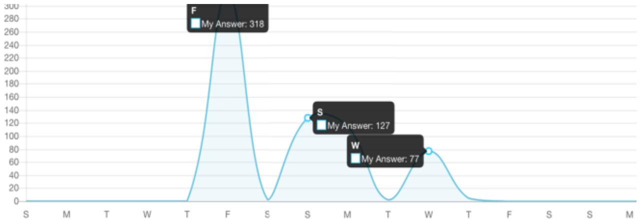

Figure 5 Number of answers on each day, from November 26th. Three dates were when we promoted the service.

DISCUSSION

Using Voluntary Crowd

The most important factor to make the system work with voluntary crowd was designing the motivation for the crowd to participate. So, the system should be designed to benefit not only the user of the system, but also the crowd participating in the system. Our approach for this system was giving fun to the crowd. Based on cases where people enjoyed reading other's chat and giving advice on it, we thought using fun could work for the system.

Our user study result showed that having fun could be a possible motivation for the crowd. However, the motivation was enough for the crowd to participate the platform once, but it was not enough to hold the crowd.

One point of concern is that the platform do not have strong quality control methods. Though no trolling behavior was observed during the deployment, quality control would be an essential feature for the platform to serve appropriately.

Real-time

Since our platform was targeted for getting the true meaning of the conversation during the chat, it was essential to give prompt answers to the requester. For the crowd to answer new questions promptly, they should keep concentrated to the platform. However, there was nothing to do after answering questions, so the crowd were not motivated to keep concentrated.

As a solution, we implemented a push notification for new questions. However, most of the crowd did not subscribe for the push notification. Its support was limited, and the subscription process was too bothersome. Having more contents to hold the crowd, such as showing other's answers in crowd's interface, could be a more fundamental solution.

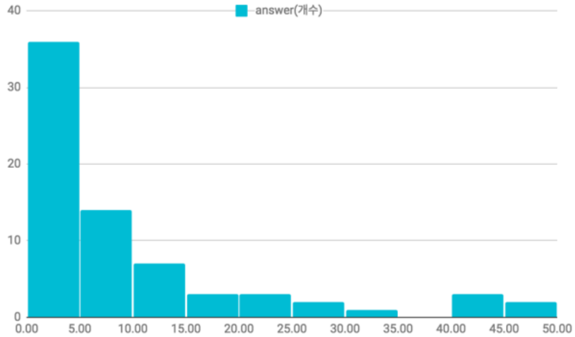

Figure 4 Histogram of the number of the users by the number of their answers. Most users posted less than 10 answers.

Barriers for Asking Questions

Deployment results showed that the number of questions posted was quite small. In some cases, the crowd did not have anything to do because they answered every question on the platform. There were two barriers inhibiting requesters to ask questions.

First, situations with vague true meaning were quite rare. There were several questions about the conversations from six month to a year ago, meaning that the questions were made up just to try out the service. Since the deployment period was limited, it could be possible that there were not enough realistic situations for question. Also, it could be possible that people would reluctant to share their chats, since we promoted the service with our close friends first.

Second, the process to ask questions was complex. For the requesters to ask questions easier in the messenger itself, we designed for the requesters to initiate the question posting process by sending a capture of the conversation to KakaoTalk PlusFriend. However, to finish posting the user should go to the web interface, which might feel awkward and complex for the users. Furthermore, we could not give a direct link to the requester's interface while we were promoting the service, so it could be less accessible. Integrating the service with the messenger more tightly would be a solution to simplify the process and help requesters to post question more comfortable if appropriate situation occurs.

Video available at https://youtu.be/24dy5Z9G1cg

REFERENCES

1. Glaser, J. Self-Expression. Psychology Today, 2016. https://www.psychologytoday.com/blog/conversational-intelligence/201602/self-expression.
2. Statista. Most popular messaging apps. 2017. *Statista*, 2017. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/.
3. Walther, J. B., & Parks, M. R. (2002). Cues filtered out, cues filtered in. *Handbook of interpersonal communication*, *3*, 529-563.
4. 연애의 과학. *카톡으로 보는 속마음*, 2017. http://scienceoflove.co.kr/.