Real-time On-Demand Crowd-powered Entity Extraction

Ting-Hao (Kenneth) Huang$^1$, Yun-Nung Chen$^2$, Jeffrey P. Bigham$^1$
$^1$ Carnegie Mellon University, $^2$ National Taiwan University

1. INTRODUCTION

Output-agreement mechanisms such as ESP Game [Von Ahn and Dabbish 2004] have been widely used in human computation to obtain reliable human-generated labels [Von Ahn and Dabbish 2008]. In this paper, we argue that a time-limited output-agreement mechanism can be used to create a fast and robust crowd-powered component in interactive systems, particularly dialogue systems, to extract key information from user utterances on the fly.

Modern dialogue systems often rely on entity extraction to understand user’s requests [Bohus et al. 2007]. For instance, when a user verbally asks about nearby restaurants, the entities that are mentioned in the user’s sentences, such as location (e.g., Pittsburgh) and preferred food (e.g., Chinese food,) are critical for the system to find the appropriate information. However, automated entity extraction is not perfect. While the common practice is to have crowd workers label data and then use the annotated data to train a supervised entity recognition model such as Conditional Random Fields (CRF) [Raymond and Riccardi 2007] and Recurrent Neural Networks (RNN) [Mesnil et al. 2015], these models can be vulnerable to unseen entities [Xu and Sarikaya 2014] or difficult entities such as Genre in movie domain [Wang et al. 2014]. Research has thus utilized the Web [Wang et al. 2014] or unsupervised methods [Chen et al. 2014; Heck et al. 2013] to improve the performance, although these technologies are currently still far from practical use.

This paper explores an alternative path to tackle this challenge: Having crowd workers to extract entities on-demand within few seconds. Even with a tight time constraint such as 60 seconds, human workers are known to perform better than automated approaches in many tasks such as answering questions and rating texts [Savenkov and Agichtein 2016]. Furthermore, when end-users conversing via instant messaging clients, they reportedly expect longer response times such as 30 seconds [Avrahami et al. 2008; Baron 2010]. This range of latency allows real-time crowdsourcing to intervene on the fly. While prior works used crowdsourcing to extract entities from texts [Huang et al. 2015; Lasecki et al. 2013] and to collect data [Wang et al. 2012], none of them reported performances in terms of time and accuracy under an on-the-fly condition. In this paper, we propose to use real-time crowdsourcing as on-demand entity extractors in dialogue systems. Our experiments show that our crowd-powered approach is robust, effective, and fast.

Dialogue ESP Game: We utilize real-time crowdsourcing with a multi-player time-limited ESP Game setting to extract the target entity from a dialogue. The ESP Game was originally proposed as a crowdsourcing mechanism to...
acquire quality image labels [Von Ahn and Dabbish 2004]. The original game randomly pairs two players and presents them with the same image. Each player guesses the labels that the other player would answer. If the players match labels, each is awarded 1000 points. Our approach replaces the image with a dialogue chat log, and players answer the required entity name within a short time (e.g., 20 seconds.) To increase speed, we relax the constraints of player numbers from two to five or ten. As Figure 1(a) shows, by aggregating input answers from all players, our Dialogue ESP Game is able to provide high quality results in seconds. Figure 1(b) shows the worker interface, in which workers are asked to answer “What is the <target_entity> in this dialogue?”, and the definition of the <target_entity> is displayed on the left side. When two answers agree, a notification pops up, and the workers earn 1000 points. The timer, which is displayed in the top right corner of the interface, starts counting down when the task begins. When the time is up, the task automatically closes. To recruit workers quickly, many approaches have been used in real-time crowd-powered systems such as VizWiz [Bigham et al. 2010]. In this paper, we first focus on the speed and performance of the Dialogue ESP Game itself instead of recruiting time, and briefly discuss the end-to-end response time in the Experiment 2.

2. EXPERIMENT 1: SIMULATION STUDY ON AMAZON MECHANICAL TURK

To evaluate the Dialogue ESP Game for entity extraction, we used a benchmark corpus for language understanding, the Airline Travel Information System (ATIS) dataset. ATIS contains conversational query sessions of flight schedules. We had Amazon Mechanical Turk (MTurk) workers to answer the destination city (tloc.city_name, can be null), which is the most frequent entity, in each given conversation of ATIS by using our interface with a time limit.

**Trade-offs between Accuracy and Speed:** The performances of Dialogue ESP Game correspond to its three main variables: the number of players recruited to answer each session, the time limit that each player has to answer a session, and the method to aggregate input answers. We have three ways to aggregate the input answers from the Dialogue ESP Game, each could result in different accuracies and speeds: (i) ESP Only: Return the first matched answer when it matches. If no answers match before the time limit, return an empty label; (ii) 1st Only: Return the first input answer when it arrives; and (iii) ESP + 1st: Return the first matched answer when it matches. If no answers match before the time limit, return the first answer. In this section, we first focused on the context-independent query set (Class A), which contains answerable single-sentence queries and has been widely used by the dialogue system community [He and Young 2003; Raymond and Riccardi 2007; Tur et al. 2010]. 200 queries were randomly extracted from the Class-A developing set for our study. We collected 10 MTurk workers’ results for each ESP-game trial, and randomly selected workers’ results to simulate various player numbers. All results reported in Experiment 1 are the averages of 20 rounds of this random-pick process. We ran two sets of studies with time limits set at 20 and 15 seconds, respectively. In the actual experiments on MTurk, 5 Dialogue ESP Games for 5 different Class-A conversations are aggregated in one task, with a extra tutorial game at the beginning. The results are shown in Figure 2 and Table I.

---

The source code of worker interface and the data collected in Experiment 2 are available at: [https://github.com/windx0303/dialogue-esp-game](https://github.com/windx0303/dialogue-esp-game)
Real-time On-Demand Crowd-powered Entity Extraction

<table>
<thead>
<tr>
<th>Query Category</th>
<th>Class D (context-dependent)</th>
<th>Class X (unevaluable)</th>
<th>Class A (context-independent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>Resp. Time</td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>CRF Baseline</td>
<td>0.043s</td>
<td>.776</td>
<td>.307</td>
</tr>
<tr>
<td>1st Only</td>
<td>5.460s</td>
<td>.658</td>
<td>.641</td>
</tr>
<tr>
<td>ESP + 1st</td>
<td>7.118s</td>
<td>.814</td>
<td>.797</td>
</tr>
</tbody>
</table>

Table I: Result for Class D, X and A. Crowd-powered entity extraction outperforms the CRF baseline in terms of F1-score on both Class D and X queries. Although the CRF baseline is well-developed on Class A, it is not effective on complex queries.

Fig. 3: Timeline of the real-time crowd-powered entity extraction system, Eatity. On average, the first worker reaches at 30.83 seconds, the first answer is received at 37.14 seconds, and the first matched answer occurs at 40.95 seconds.

Complex Queries: We extended our study to complex dialogues by using the other two categories in ATIS, context-dependent (Class D) and unanswerable (Class X), where the “context-dependent” queries require information from previous chat log in the same session, and “unanswerable” queries do not contain any requests that the ATIS system can respond to. We randomly extracted 200 and 150 queries from the Class-D and Class-X set, respectively, and conduct experiments on MTurk (time limit = 20 seconds, 10 players.) Note that for each extracted query, all previous chat log before it within the same conversational session were also obtained and displayed in the worker interface. The results (Table I) show that the proposed approach outperforms automated CRF baselines, which is trained on the Class-A training set by using neighbor words and POS tag features, with an average response time shorter than 9 seconds.

3. EXPERIMENT 2: END-TO-END SYSTEM STUDY

We created a Google Hangouts chatbot, Eatity, that automatically extracts food entities from received messages by using the Dialogue ESP Game, and conducted a lab-based user study. Eatity (Figure 3) recruits crowd workers on MTurk in real-time to perform the Dialogue ESP Game task upon receiving a message. To recruit workers quickly, each time Eatity posts 120 short lifetime (60 seconds) tasks to increases task visibility. This recruiting method bypasses maintaining a worker waiting pool and is easy to implement. 10 participants entered our lab and were asked to send 15 messages to Eatity via Google Hangouts. In these 15 messages, participants needed to mention 9 foods, 3 drinks, and 3 countries arbitrarily. Correspondingly, the instruction in workers’ interface was modified as “What is the food_name in this dialogue?”, and the description of food_name was modified as “Food name. The full name of the food. Including any drinks or beverages.” As a result, Eatity achieved an accuracy of 78.89% in extracting food entities and 83.33% in extracting drink entities. A user on average spent 27.05 seconds to type a message. If we align the user typing time along with the system timeline, the perceived response time to users falls within 10-14 seconds (Figure 4).

Fig. 4: The F1-score curves of RNN model trained on crowd-labelled and expert-labelled data w.r.t. number of training samples.

Collective Intelligence 2017.
4. WHEN TO USE THIS?

Even though automated annotators are often preferable to crowd-powered technologies due to cost and latency, our proposed method can be used as the backup system for unseen or difficult entities and complex dialogues. Furthermore, the collected annotations can serve as training data for automated models. For instance, we trained the state-of-the-art RNN language understanding model \cite{Mesnil2015, Chen2016}, respectively on the crowd-labelled and expert-labelled 200 Class-A ATIS conversations used in our study. The evaluation results by using the ATIS Class-A testing set (Figure 4) show that the crowd labels are as effective as expert labels of being the training data for bootstrapping the automated systems. In the future, we will generalize our approach by adding automated components, and also explore the possibility of using audio input.

5. ACKNOWLEDGEMENTS

This research was supported by the Yahoo! InMind Project \cite{Azaria2016} and the National Science Foundation (#IIS-1149709). We also thank the workers on Mechanical Turk who participated in our experiments.

REFERENCES


Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding.. In INTERSPEECH. INTERSPEECH, USA, 1594–1598.


Collective Intelligence 2017.