Real-time Trip Planning with the Crowd

Joey Rafidi
MIT CSAIL
32 Vassar St
Cambridge, MA 02139 USA
jrafidi@mit.edu

CHI 2013 Extended Abstracts, April 27-May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract
When confronted with an unexpected turn of events or feeling spontaneous, travelers find themselves in need of on-the-spot planning assistance. We present *Crowdcierge*, a real-time crowd-powered trip planning application. Crowdcierge is capable of both planning new trips and re-planning on the fly based on unstructured natural language input. By using the retainer model, synchronous crowd collaboration, and crowdware, workers are quickly recruited to work together to accurately tag key ideas in a planning mission, plan the itinerary, and re-plan in response to problems that arise during the trip. This paper presents the design of each crowd task in Crowdcierge and preliminary results from the tagging task.

Author Keywords
Human computation; crowd collaboration; trip planning and re-planning

ACM Classification Keywords
H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces

Introduction
Planning an itinerary of activities that satisfies a traveler's desires is a complex task. While there exist numerous online resources to help find travel activities, parsing through all the information available and creating a plan is difficult and time consuming.

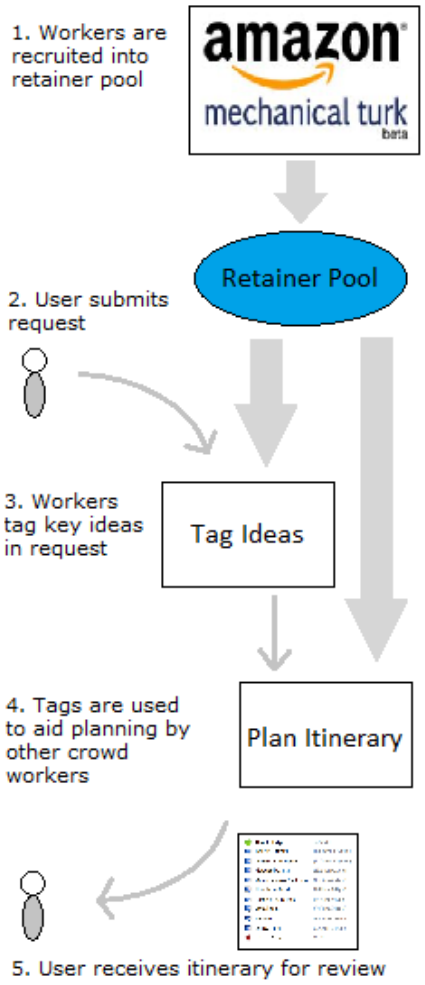

Figure 1: Full planning process from request to itinerary.

Alternatively, travel agents and concierge services can provide recommendations, but provided plans tend to be generic and touristy. Furthermore, none of these sources are available on the fly. If a traveler unexpectedly has an extra day to spend or a previously planned activity is closed, there is no immediate assistance available to them.

One possible solution is to recruit crowds of volunteers or paid workers to plan the trip. Previously, Zhang et al. developed *Mobi*, a crowd-powered trip planning application [1]. Users input a natural language request and a list of quantitative constraints on the types of activities they want to have planned for them. Through the *crowdware* paradigm, workers then make asynchronous contributions to planning the itinerary, while the system continually notifies workers of what constraints are not yet satisfied.

Mobi takes several days to plan a trip, making it helpful prior to traveling but not during. To give travelers quick and convenient travel planning assistance, we introduce *Crowdcierge,* a system that delivers real-time trip planning responses based on natural language input. Crowdcierge identifies the key ideas in the request in real-time, which can then be presented immediately to the user to ensure that the crowd has understood their interests prior to planning. Crowdcierge then plans the itinerary as quickly as possible for the convenience of spontaneous travelers. Finally, users can request changes in the plan at any time, as they encounter problems or wish to do something different during their trips.

To achieve real-time trip planning, Crowdcierge uses the *retainer model,* in which the crowd workers are paid a small wage to wait for a task to become available [2]. When needed, the system signals the workers to enter the tasks. This recruitment system is used for all three of the crowd-sourced tasks in Crowdcierge. First, the user sends a natural language request. Next, in the tagging task, the crowd works to identify the key ideas required in their trip. The tagged ideas are passed into the second task, the planning task, and are used as constraints on the itinerary. Workers then plan the trip through a *crowdware* system. Once the itinerary satisfies all the constraints, it is delivered to the user. These first two tasks are shown in the workflow in Figure 1. At any point in the trip, the user can request a change to the itinerary with natural language input. The crowd re-plans the trip using this input, the current itinerary, and all previous ideas from the planning task.

In building this application, we make the following contributions to crowdsourcing:

- We develop a collaborative tagging interface for workers to quickly extract key ideas from a trip request and in general extract preferences and constraints from natural language texts.
- We combine the retainer model and the crowdware paradigm to enable the crowd to collaborate and complete complex tasks in real time, such as planning or re-planning a trip.

Related Work

There have been a growing number of recent studies on how people plan their trips collaboratively, including the effectiveness of assigning explicit roles to group members in travel planning [3], the requirements for company employees in their social travel arrangements

[4], as well as the prototyping study of a social website to support more unstructured, ad-hoc trip planning activities by remote users [5].

From the crowdsourcing perspective, Crowdcierge builds directly on a crowd-sourced trip planning application called *Mobi*, which used the crowd to plan trip itineraries through the *crowdware* paradigm [1]. Our application also uses crowdware but in a synchronous task, giving us more potential collaborative interactions to consider. We currently use a slightly modified version of the Mobi planning interface in our planning and re-planning tasks.

To obtain real-time response from the crowd, recent work has developed the *retainer model*, a system capable of recruiting several crowd workers into a task quickly by maintaining a waiting pool of workers [2]. We use the retainer model for all three of the tasks in our application to produce a real-time response. By keeping workers waiting in the retainer system, we can bring multiple workers into a task when needed to start working as soon as possible.

Finally, Dow et al. have shown that crowd workers perform better when asked self-assess their work or when given external assessment feedback [6]. Similarly, the tagging task in our application asks crowd workers if they believe the task is done before they submit. The synchronous collaboration in our tasks also introduces an implicit peer-assessment system, where workers are effectively checking and reviewing each other's work.

Design and Implementation

Crowdcierge consists of three different crowdsourced tasks:

1. Tagging Task: Selecting and tagging the keywords or phrases in the trip request
2. Planning Task: Planning the itinerary based on the request and the tags
3. Re-planning Task: Re-planning the itinerary based on the previous solution context and the traveler's problem

Tagging Task

In order to extract the trip constraints from the traveler's natural language request, one or more crowd workers work simultaneously to identify and tag all the key ideas in the request (Figure 2). Workers highlight any key words or phrases they see in the text and then label these sections with a short tag. When more than one worker decides the task is complete by clicking the "I think we're done" button, the workers can submit the final list of tags. Alternatively, a single worker can submit given they mark the task as done and wait for five seconds. This does not prevent other workers from modifying the list of tags.

Based on user testing within our lab, we found that requiring workers to highlight parts of the request before making a tag ensured they captured the correct constraints. Without requiring a snippet from the request to define a tag, pilot workers sometimes extrapolated beyond what was mentioned in the request and created incorrect tags. For example, when a request mentioned that the user wanted a "relaxing" day, a pilot worker created a tag for "coffee breaks". While coffee breaks are relaxing, the request did not

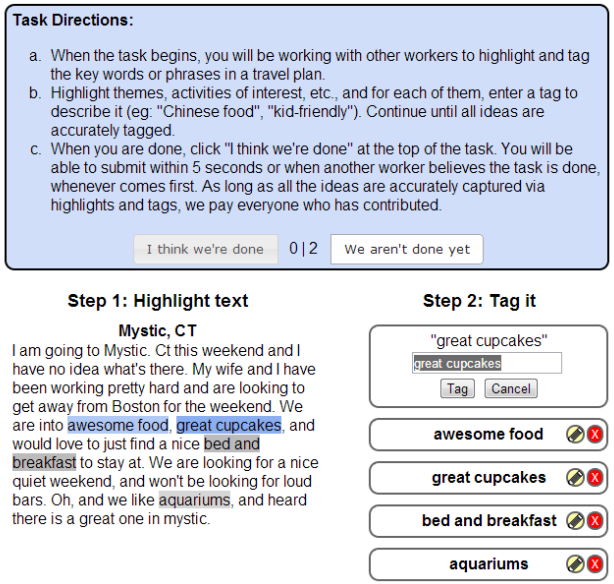

Figure 2: The tagging task interface, with the request in the bottom left and the tags in the bottom right. The task directions are at the top, with the buttons indicating if a worker believes the task is done.

explicitly mention coffee breaks. This kind of extrapolation does not properly capture the user's ideas and may lead workers in the planning phase to miss non-coffee related relaxing activities.

As shown in Figure 2, workers can see which areas of the request have been tagged by the colored highlights over sections of the request. From our prototype testing, we found that workers were less redundant in their tags when they could see what sections of the request were already tagged. In this way, they could focus on unfinished sections of the text. A blue highlight corresponds to a tag the worker created themselves, while grey highlights correspond to tags other workers created. Each of these highlighted sections maps to a tag in the rightmost column, and a tag can map to one or more highlighted sections. Regardless of who created a tag, any worker can edit or remove tags in the list.

To ensure that all areas of the request are covered in a timely manner, multiple workers are recruited to simultaneously tag the same request. While working, they see in real-time all tag creations and edits, and they are also informed as to how many workers believe the task is done. Finally, when the task is finished, the user can check to see if the crowd captured his or her ideas correctly before planning begins.

Planning Task

Once the tags have been generated and submitted for a given request, workers begin planning the trip based on the tags. The workers suggest activities, indicating which tags the activity is related to, and then place the activities in the itinerary. Crowdcierge enforces the constraints that each tag must have at least one activity related to it in the final itinerary. When these constraints are violated, Crowdcierge automatically generates and displays *to do* items to indicate to the crowd workers what constraints are not satisfied given the current state of the itinerary. By maintaining these "at least one" constraints, Crowdcierge ensures that all of the tagged key ideas are expressed in the itinerary without assuming anything about how much of a certain activity type the user wants. The interface for this task is identical to that of the re-planning task in Figure 3.

As in the tagging task, multiple workers work simultaneously in this task to plan the itinerary. While the system can automatically check and see if there is at least one of each tagged activity type, we also ask the crowd to assess each constraint once all the auto-checks are satisfied, linking the tags back to the corresponding text in the planning mission. The user can also make changes to the plan at any time.

Re-planning Task

If the user encounters any issues during their trip, they can ask the crowd to re-plan the itinerary. As shown in Figure 3, the user gives a natural language explanation that is turned into a to do message. The crowd sees the message, how much of the itinerary has already been completed, and all the previously suggested

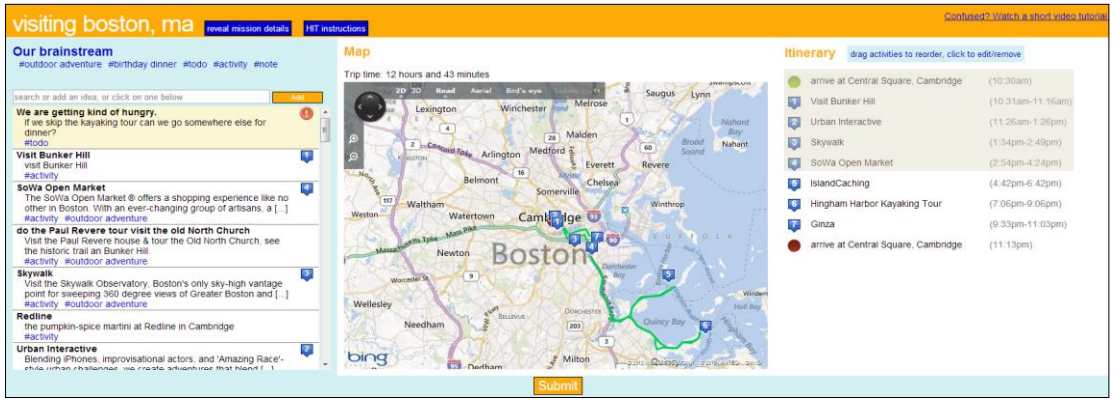

Figure 3: An example of a re-planning task for the crowd to complete.

ideas. Based on this context, the workers re-plan the itinerary.

The interface for this task builds off of the existing Mobi planning interface, indicating how much of the itinerary has already been completed by the traveler. Since real-time response is most important in this task, we aim to present the solution context as efficiently as possible, letting crowd workers make changes quickly.

Evaluation

We ran preliminary Mechanical Turk trials of the tagging task using individual workers and groups of two to three workers collaborating synchronously. In our crowd trials, we hired Mechanical Turk workers from the US with at least a 95% approval rating, or any worker with at least a 98% approval rating.

We hypothesized that individual workers would not complete the task as thoroughly as a group of workers. We expected individual workers would either fail to tag some key constraints in a request or create erroneous tags. In the group setting, we expected these errors to be corrected by the other workers in the task. To evaluate the submissions, we compared the tags created to an answer key we generated by doing the task ourselves. A total of 30 Mechanical Turk workers participated in the trial. 25 workers worked independently and 5 workers worked together in 2 separate sessions.

From these trials, we found that individual workers very rarely captured all of the key ideas in a request. Shorter requests of only a few sentences in length were usually completely tagged, but workers almost always missed ideas in the longer requests. Omissions were much more frequent than erroneous additions. Averaged across all trials with five different trip requests, workers successfully tagged 65% of the key ideas, with a success rate of 71% in the request with the least tags and 50% in the request with the most tags. Since most workers would directly quote one or two words of the request into a tag, only a couple workers ever made tags that directly contradicted the user's request.

In comparison, small group trials with equally qualified workers showed the expected error-catching behavior, resulting in more complete submissions. While one worker would quickly pass over the request and tag the obvious key points, others in the group would pick out ideas that the first worker missed. This gave a more complete set of trip ideas that nearly matched our answer key. For example, in one request, the user specified they would like to see the "Statue of Liberty" and eat at "good Indian restaurants". Individuals sometimes missed these ideas, whereas the groups captured both correctly.

Figure 4: From top to bottom: a Las Vegas trip request, the tags generated by the crowd, and the plan the crowd created.

From the crowd trials, we also observed that asking workers to determine if the task is complete before submitting improved the quality of result. We believe this self-assessment prevented eager workers from simply submitting the task whenever they wanted and made them consider the quality of their submission, similar to the self-assessment effect observed by Dow et al. [3].

Finally, we had the crowd plan a trip using tags made by an individual worker to conduct an end to end test of the full planning pipeline (Figure 4). The itinerary the crowd planned was coherent and satisfied all the constraints laid out by the tags. Most importantly, the itinerary appears to satisfy the user's preferences as stated in the planning mission.

Future Work

In order to fully understand the benefit of synchronous crowd collaboration, we plan to do a controlled study of the individual vs. group tagging task and analyze the results. Next, we will test the planning and re-planning tasks for feasibility and quality with individual workers or groups. Finally, we will test the tagging, planning, and re-planning tasks together using real-time crowd recruitment methods, hopefully achieving real-time trip planning.

We aim to redesign the planning and re-planning tasks to support additional means of communication between the user and the crowd. Along with the ability to modify the itinerary, we will allow users to give natural language feedback to the crowd based on the current solution, or they can receive periodic updates if they are not in a position to easily check on progress (i.e. they are driving in the car). Based on the updates, users can send a quick message back to the crowd as feedback.

We also plan to experiment with different methods for delivering real-time results. Instead of demanding the full solution as soon as possible, the crowd can compute the later parts of the itinerary while the traveler is enjoying the first few activities planned. Moreover, even when the itinerary is complete and the user is on the trip, the crowd can be left to compute alternatives in advance in case complications arise.

Acknowledgements

This work was made possible by Haoqi Zhang's and Robert Miller's mentorship, as well as the support of the MIT UID group. This work is partially sponsored by Google as part of the MIT EECS Super-UROP program.

References

[1] Zhang, H., Law, E., Miller, R. C., et al. 2012. Human Computation Tasks with Global Constraints. *CHI '12*.

[2] Bernstein, M. S., Brandt, J., Miller, R. C., and Karger, D. R. 2011. Crowds in Two Seconds: Enabling Realtime Crowd-Powered Interfaces. *UIST '11*.

[3] Imazu, M. 2011. Effect of explicit roles on collaborative search in travel planning task. AIRS '11, 205-214.

[4] Aizenbud-Reshef, N., Barger, A., Guy, I., Dubinsky, Y., and Kremer-Davidson, S. 2012. Bon voyage: social travel planning in the enterprise. CSCW '12, 819-828.

[5] Dean M. G. Hargreaves and Toni Robertson. 2012. Remote participatory prototyping enabled by emerging social technologies. PDC '12, 25-28.

[6] Dow, S. P., Kulkarni, A., Klemmer, S. R., and Hartmann, B. 2012. Shepherding the Crowd Yields Better Work. *CSCW '12*.