Crowdsourcing on In-building Routing System with Personal Preferences across Building Groups

Junnan CHEN
Waterloo, Canada
j486chen@uwaterloo.ca

Henagzhi ZHANG
Waterloo, Canada
h246zhang@uwaterloo.ca

ABSTRACT

People travel inside building groups from a room in building A to another in building B frequently. Traditional routing systems usually provide the shortest routes between two coordinates consist of longitudes and latitudes, but do not have knowledge about building structures and detailed floor maps. These systems do not take users' personal preferences into account when suggesting routes as well. Such routes cannot meet peoples demand. In this paper, we introduce a new system, which provides in-building routes based on personal preferences across building groups.

Author Keywords

Crowdsourcing; In-building Routing; Personal Preferences; Mobile Interfaces; Concept of Association; Decision Tree.

INTRODUCTION

introduction In our world, most people are living, working or studying in a group of buildings which could be company buildings, neighbourhoods or campus buildings. People go from one location inside the building-group to another every hour every day. Traditional route planning system provides routes between two coordinates consist of longitudes and latitudes, focusing on finding the shortest route, but does not have knowledge about the inside structures of buildings and floor-detailed environments, and does not take users' personal perferences into account when suggesting the routes. Those routes cannot provide enough details and personal options demanded by people. For example, an employee may need to go to the coffee shop on the first floor of building B from his office on the third floor of building A several times a day. His route choices would be based on the weather condition and his physical condition. He would choose to go through the hallway between the two buildings if it rained or take the elevator if he feels tired. Traditional route planning system would only provide the shortest route from the exit of building B to the entrance of building A regardless of the building structures or floor environments, which cannot meet this employee's actual needs.

Here we want to introduce a new in-building routing system with persoanl preferences, a system with comprehensive knowledge of building structures, floor-level detailed maps, and relevant environments, which provides room-level detailed routes with personal preferences specified by the user. The focus of this system would be mapping the physical features of routes to human preferences which consist of physical requirement, personal feelings, emotional options and etc. The system shall collect route and preference data from people, and then analyze the relationships between human preferences and physical features.

Our system collect the trainning data, which are actual routes each has its personal preferences labeled, by crowdsourcing methods. Crowdsourcing has been successfully used in many fields. We provide an application on mobile devices, which encourages people to record their daily routes from one room location to another across building groups. Before they start to record routes, the application will ask the user about their personal feelings today (e.g. Are you hurry now? or Are you feeling tired now?), their special demands for the following route (e.g. Will you pass Tim Hortons on your way? or Do you need to use the bathroom on your way?) as well as some weather conditions (e.g. Is it sunny out side? or Is it snowing now?). Once the users finish the questionares, they will be presented an interface for them to record their routes. During the recording process, the application collects physical data of the route in the form of points on the floor map inputted by the user.

Our system will analyse the routes collected and the corresponding physical features of a particular route (e.g. the number of stairs, number of turns) with personal preferences such as physical requirement, personal feelings, and emotional options provided by the user. We will construct our model using the concept of decision tree. The model is trained by the collected data and will be able to predict the best route given a set of personal preferences.

Our final product aims to generate the proper routes upon request. Using this product, the user could specify two room locations, together with one or more personal preferences. Then, the system will run the model and return the most satisfying route. The system will display the route on the screen for the user.

In this paper, we describe the design of the courdsourcing routes collecting method, the feature analysing method and route generating system. We test the system with University of Waterloo campus building groups. We provide the test

results, the detailed evaluation of our system and the future work.

RELATED WORK

During the system design process, we reviewed literature related to our ideas. The Kimono [6] is a knowledge sharing system, which presents the concept of association. Our in-building routing system uses this idea as well; for example, each route collected in the crowdsourcing stage is associated with a set of personal preferences specified by the user. One of the prior works we reviewed has the similar idea since it [4] includes a physical kiosk as well. In addition, the physical kiosks [4] are used for crowdsourcing. Since our system collects in-building routes with personal preferences from our users, crowdsourcing plays a significant role here, and crowdsourcing needs to be studied from the prior work [5, 7, 1]. Since our system requires users to record routes on the map displayed on the screen and provide inputs through the questionnaire, user interface designs need to be taken into our consideration. Some prior works [3, 11] provide us with very helpful results for our system design. Our results are based on decision trees. In order to have some insights about decision trees, we reviewed and studied related work [10, 9]. In this section, we discuss how prior work helps us develop our in-building routing system with personal preferences across building groups.

Concept of Association

The Kimono paper [6] presents a system, which extends the function of an information kiosk. The method they used in order to achieve the extension is to allow interactions between a kiosk and a mobile device, such as a smartphone. We know that a kiosk is usually a physically located machine displaying information. A user could stand in front of a kiosk, and touch the screen to select information of his or her interest. The information from the kiosk is relevant to the particular event at its location. Some disadvantages of a kiosk can be seen. For example, the kiosk lacks mobility, and its difficult to add more information to it.

To conquer these inflexibilities, the main idea introduced in the paper [6] is to allow interactions between a mobile device (such as a smartphone) and a kiosk. People are familiar with smartphones as they use phones every day. Functionalities of a smartphone include taking pictures, taking notes, recording voices, and so on. Through these various ways, a person can easily create new contents at any time at his or her current location. Then he or she can associate the newly created contents with other selected contents or events presented on the kiosk. By uploading the new contents and its associations, other people can therefore have access to them.

The Kimono system [6] described in the paper is designed specifically for researchers attending a conference. A number of kiosks are placed in the lobby of the conference. They all display the same information relevant to the event, such as conference locations, routes to the airport, hotels, etc. Changes of talk schedules and meal plans are posted on the kiosks as well. Participants of the conference use the touch panel display of a kiosk to select information interested in. Then the information selected is transferred to his or her smartphone. On the smartphone, the program will alert the user about the next event and other relevant details such as starting/ending time, room location, routes to the destination, etc.

While attending a lecture at the conference, one can use their smartphones to record notes in the form of text, image, video, or audio. Then they associate newly created notes with selected event or items such as papers or posters presented at the conference. People can also exchange notes with each other directly between two mobile devices, or post the notes they took on the kiosk. For those users who previously downloaded information from the kiosk to their smartphones, if the information downloaded has changed, the updates will be copied to the mobile devices through the system when the next synchronization occurs.

The main insight we learned from the Kimono [6] is the concept of associations, which makes the organization of data and synchronization much more simple. Users can get information of his or her interest displayed on the screen and associated information on the smartphone. New information can be created by the user, associated with other relevant information, and made available to others.

For our in-building routing system with personal preferences across building groups, the concept of association is utilized. During the crowdsourcing stage, users record their routes while walking to the destination, and later associate personal preferences or features with the route they just recorded. Personal preferences in our system include physical requirements, personal feelings, emotional options, etc. By associating personal preferences with the routes, the system could provide in-building routes with particular personal preferences specified by a user, and later collect ratings from the user about the routes suggested.

Crowdsourcing

We can see that a lot of researches and experiments are conducted in prior works [4, 1] in the area of crowdsourcing. Here we discuss the paper [4], in which crowdsourcing concepts are presented and extended as communitysourcing. What we know is that crowdsourcing involves division and assignment of tasks to a number of online users. In order to have better quality work, the kiosk is used to attract the right crowds to perform tasks. In this case, the right crowds are required to be knowledgeable enough in the corresponding task domain assigned to them. The result the paper showed is that the crowdsourcing system was able to perform the grading task more accurately than traditional grading.

What we did is to incorporate the idea of crowdsourcing into our system. That is, we let crowds generate routes with personal preferences through our system. As the system provides in-building routes across building groups on campus, the potential crowds we are interested in are students, teachers, and staffs. One reason is that they are familiar with the building structures and floor-detailed environments, and therefore the quality of crowdsourced routes is greatly improved. Another reason is that we could obtain enough route data in a relatively

short period of time through the crowdsourcing stage. Furthermore, during the crowdsourcing stage, the system could capture a particular set of preferences associated with a route, and later suggest this route upon requests of same or similar preferences in the set.

User Interface

Since we require the routes to be recorded when users walk to their destinations, our system provides an interface on both computers and mobile devices. The user gets access to our system on the screen in the beginning of his or her journey. Partial map centered at his or her current position is displayed. The user periodically marks his or her next new position along the way to the destination, and the map accordingly centers at the newly marked position. A questionnaire is displayed to request his or her personal preferences associated with the recorded route before the user starts recording.

The questions on the questionnaire are asked before the recording process. The questionnaire of our system helps to collect the personal preference data with the crowdsourced routes of our interest. After thinking of what kinds of questions should be asked, we have the questions such as the weather conditions, the fatigue level, pass coffee shop or not, hurry level, and etc.

When designing such a crowdsourcing interface, we need to think about whether our user interface or task design is effective on both computers and mobile devices. We reviewed the prior paper [3] to gain better understandings and insights about crowdsourcing user interfaces and task designs for mobile users.

The main question in the paper [3] the author trying to answer is Which crowdsourcing platforms and which kinds of tasks are more adequate to mobile devices. It turns out that some of those inadequacy issues are superficial, and they can be resolved by providing better user interfaces and/or better crowdsourcing task designs. Several typical reasons for such inadequacy issues on mobile phones are found as well, such as redundant task description, unsupported formats of audio, scrolling problem, layout problem in a small display, and so on.

For our system, we also try to conquer these problems by designing simple tasks, requesting simplified inputs, providing concise interfaces, etc. Our system knows the building structures and all of the floor maps. Users records routes through their mobile devices while walking. Partial map centered at the current position is showed, and the user marks the next point along the journey. Upon request of a route with personal preferences specified by a user, the system provides in-building routes and collects ratings for the suggested route afterwards. Several benefits can be seen in our system. For example, instead of keeping track of the entire route walked, the user only needs to mark the next new point by touching the screen every time they move as the partial map will always centered at his or her current point. After gathering point information inputted by the user, the system itself can generate the corresponding route. By displaying only the partial map instead of the entire floor map, we successfully resolve the bad layout problem on small portable screens.

Decision Tree

Our results are based on decision trees. As described in the previous section, our system provides users with an interface through which users perform question answering and route recording tasks. The system requires users to answer a questionnaire before they start to record their in-building routes during the crowdsourcing stage. The questionnaire here in the system is used to collect personal preference data associated with the routes recorded. And decision trees are then built accordingly.

We built several graphs of decision trees. What we did was the review of several prior works [10, 9] involving the decision trees. A decision tree provides us with support in decision-making. It is known as a tool, which uses a tree like graph or model of decisions and their possible consequences. A decision tree is a useful technique, which learns a model from a provided data set inductively. In order to build a decision tree, we need to be clear about the structure of a decision tree. The tree structure consists of internal nodes, branches, and leaf nodes. We describe each of the above in turn. Each of the internal nodes is a test on an attribute from our crowdsourced data. Each of the branches represents the result from an internal node. Lastly, a leaf node can be seen as a class label. Three types of nodes are in a decision tree. They are decision nodes, chance nodes, and end nodes. The algorithm is shown to readers through a decision tree.

Existing algorithms are introduced in early papers [2, 8] to help build decision trees. These algorithms [2, 8] build decision trees from the given set of training data. Top down structures are built, partitioning instances into different classes. The splits of instances are based on different attribute values. The algorithm from early work [2] uses Gini index to select the splitting attribute. The Gini impurity measures how often a random element from a data set is incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Specifically, the Gini impurity is the summation of fi times (1-fi) where fi is the fraction of elements labeled with i.

Decision trees are traditionally drawn manually. However, they can grow to very large trees, which makes it difficult for a person to draw by hand. Here, we used a specialized package in Python to generate decision tree graphs for us. Our goal is to have a model that learns from the data gathered and later makes predictions on some variable of interest. In the graphs we created, colored nodes according to their classes together with explicit variable names are shown.

Decision trees are commonly utilized in many analytical procedures. What we learned from prior work are a number of advantages of such a decision support technique as follows. For example, a decision tree is relatively easy to understand and interpret, and it requires little data preparation. Furthermore, it allows adding new cases, and can be combined with other decision support techniques. The cost of prediction on

a targeted variable using the tree is logarithmic in the training data size.

SYSTEM DESCRIPTON

Our system shall be built with knowledge of building structures and floor-level detailed maps of a building group. The main problems that this system is trying to address are collecting routing data with corresponding personal preferences, constructing the routing model based on the collected data.

The route suggestion problem can be considered as a classification problem. The input data we get from the users are their current status (e.g. if they are tired) and environment conditions (e.g. is it sunny). Our goal is to generate an most suitable route based on all of these variables. For example, if the users are tired, a route using elevators should be more suitable than one which uses stairs intuitively. Since most of the status and conditions we currently consider, like if the user is tired or if the user needs to use a printer, are discrete variables. Taking this into considerations, we determine that a decision tree should be a good option.

Basic workflow

Our basic workflow of the modeling procedure is shown in Figure 1

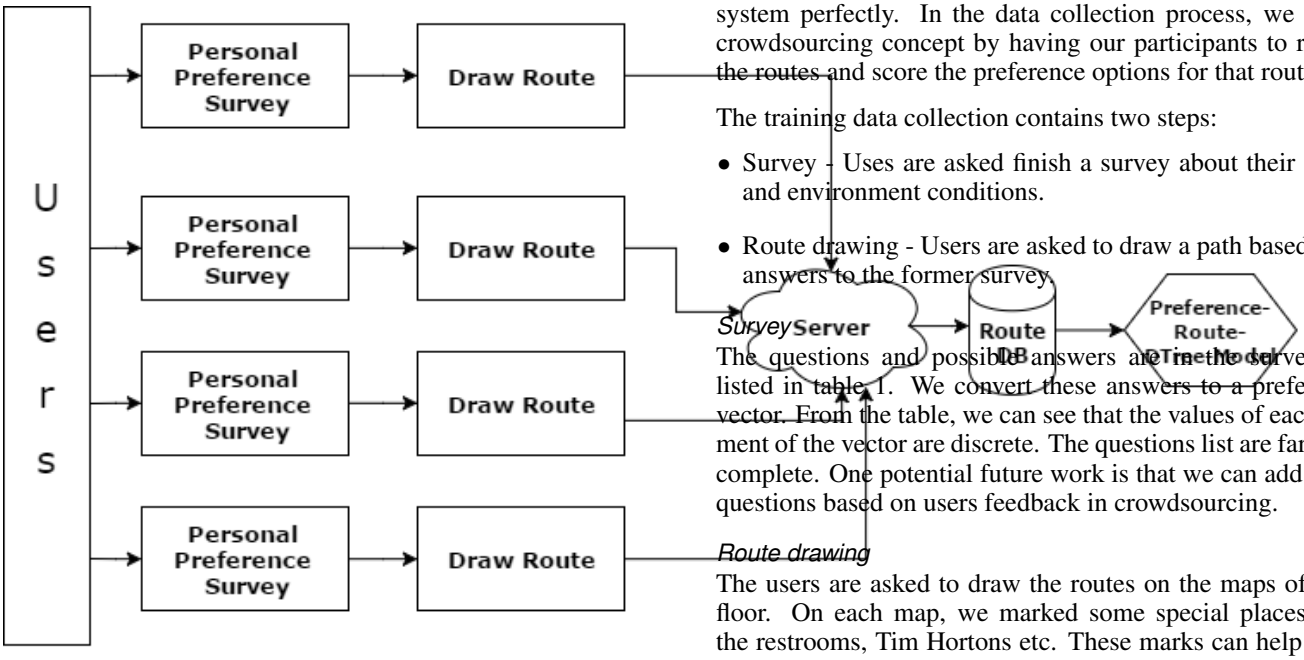


Figure 1. System flow chart

The modeling procedure mainly consists of two parts:

- Training data collecting - Crowdsourcing
- Decision tree constructing

Training data collecting - Crowdsourcing

To build a model for the route suggestion problem, we first need to collect data for training decision trees by taking advantage of crowdsourcing.

Table 1. Survey questions and answers

Questions	Answers
Is it sunny outside?	Yes/NO
Is it cloudy outside?	Yes/NO
Is it rainy/snowy outside?	Yes/NO
Do you feel tired now?	Yes/NO
Do you want to go to Tim Hortons on your way?	Yes/NO
Do you want to go to bathroom on your way?	Yes/NO
Do you want to avoid crowds at a particular time (e.g. heavy crowds outside MC2065 when class ends, crowds when special events take place, etc.)?	Yes/NO
Are you curious of the floor environment (e.g. first time visit, etc.)?	Yes/NO
Do you need the printer on your way?	Yes/NO
Are you hurry or not through this walk?	Yes/NO
Do you want some fresh air on your way?	Yes/NO

Crowdsourcing basically has two aspects, to break a large-scaled complex problem into several pieces of sub-problems, and to involve human efforts to solve or help solve the sub-problems which could be time saving because the solving of sub-problems are highly paralleled. Therefore, we found crowdsourcing has the properties which fit the demand of our system perfectly. In the data collection process, we apply crowdsourcing concept by having our participants to record the routes and score the preference options for that route.

The training data collection contains two steps:

- Survey - Uses are asked finish a survey about their status and environment conditions.
- Route drawing - Users are asked to draw a path based their answers to the former survey.

Survey

The questions and possible answers are in the survey are listed in table 1. We convert these answers to a preference vector. From the table, we can see that the values of each element of the vector are discrete. The questions list are far from complete. One potential future work is that we can add more questions based on users feedback in crowdsourcing.

Route drawing

The users are asked to draw the routes on the maps of each floor. On each map, we marked some special places, like the restrooms, Tim Hortons etc. These marks can help users to plan their route according to their answers (e.g. if they want to grab a cup of coffee). After the users finished their drawing, we will store this route in our server. The route data is in JSON format. One example is:

```
"data":{
  "paths":[
    [
      {
        "mapURL":"maps/dc1.png",
        "pointList":[{"x":1,"y":1}]
      },
```

```
      {
        "mapURL":"maps/dc2.png",
        "pointList":[{"x":2,"y":2}]
      }
    ]
  ],
  "id":["1449207710639"]
}
```


After that, we will do analysis on the route data. We analyze each route to derive several physical properties:

- Number of stairs
- Times of using elevators
- Times of exiting buildings
- If going near Tim Hortons
- If going near a printer
- If going a restroom
- The distance
- If the route never goes outside

A vector of the eight properties is called a physical property vector. A physical property vector should be related to the preference vector.

In summary, for each user, the data we collect is illustrated in Figure 2.

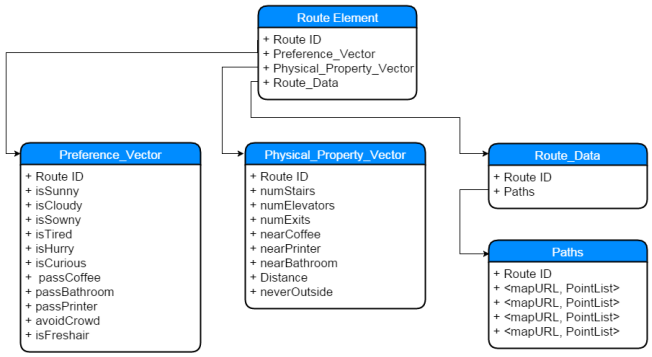


Figure 2. System flow chart

Decision tree constructing

A decision tree is a flowchart-like structure in which each internal node represents a ẗestön an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represents classification rules.

In decision analysis a decision tree and the closely related influence diagram are used as a visual and analytical decision support tool, where the expected values of competing alternatives are calculated.

We use Scikit-Learn, a python machine learning library, to construct these decision trees. The scikit-learn library is a powerful machine learning toolkit. Besides decision tree construction algorithms, it also provides a useful interface to generate a visual representation of the decision tree.

Using the training data we collected in the former step, we build two types of decision trees:

- physicalPropertyDecisionTree
- routeSuggestionDecisionTree

physicalPropertyDecisionTree

This tree is used to predict the desired physical properties by the users based on their current status and environment conditions. For each physical property, we build a decision tree independently. The reason we build a separate decision tree for each physical property is because we want to generate a physical property vector which doesnt happen in the data collection step. If we build a decision tree for the entire physical property vector, all leaf nodes of the decision tree represents a vector we have met in the collection step.

routeSuggestionDecisionTree

This tree is used to predict the most suitable route according to the physical property vector. With this decision tree, given a physical properties requirement (e.g. the route should pass by a restroom), we are able to suggest a route which match the requirement as much as possible.

Route Suggesting Procedure

In the route suggesting procedure, the basic workflow is described in Figure 3.

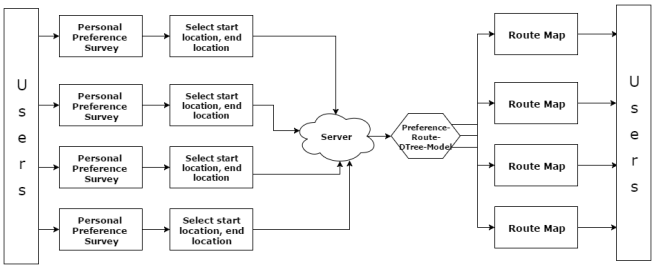


Figure 3. Route Suggesting flow chart

The suggested route is generated based on the users answers to the scenario form. The generation process consists of two steps.

Step 1: Physical Property Vector Generation

The data we received from the user, which is in the same format of the survey data in modeling procedure, is a preference vector. From the vector, we can derive the physical properties the user desires. For each property, we use the corresponding physicalPropertyDecisionTree to predict an estimated value. These values form a physical property vector.

Step 2: Suggested Route Generation

We have already generated a physical property vector in step 1. With this vector, the routeSuggestionDecisionTree can be used to generate a route suggestion.

The pseudocode for the route suggesting algorithm is:

```
function ROUTESUGGESTION(scenarioData)
    preferenceVector←PARSE(scenarioData)
    for 0 ≤ i < numOfPhysicalProperties do
        physicalVector[i]←physicalDT[i].predict(preferenceVector)
    end for
    return routeDT.predict(physicalVector)
end function
```

INTERFACES

We developed a web application for users to record data for us. Our web applications has the following advantages,

- It is light-weighted and do not require installation. Our users can visit our system by simply opening a website in their browser.
- It could be updated easily and quickly in the server end. Our users do not have the version problem.
- It has high adaptability on multiply devices. Both PC users and mobile device users can reach our system easily.
- It can be reached anytime, anywhere. Our users can label routes whenever they want. That makes it easier to collect various types of routes.
- Collected data are highly centralized in the server and no data would be stored locally. That ensures the security and stability of data.

Our web application generally has two interfaces, recordroute interface and searchroute interface. Recordroute interface allows users to fill in their personal preferences, select start location and end location and draw the route on the maps. Searchroute interface allows users to fill in their personal preferences, select start location and end location and view the route plan from the server.

Recordroute Interface

This interface is used for recording daily routes from one room location to another across building groups. Our users need to finish the survey about their current personal preferences first, and then choose the start location and end location. After that, they can start to record the route on the webpage. The basic workflow of the recordroute interface is shown in Figure 4.

Users will first be presented the personal preferences page, as shown in Figure 5, and finish the perference survey. At the bottom of this page, there is a G̈o recordb̈utton. After finishing the survey, users will click the button. Then users will be nevigated to the route drawing page, as shown in Figure 6.

Users will choose their start building, start room, end building, end room from the top menus on the page, as shown in Figure 7, and click G̈Ob̈utton on the right side of the menu.

Once the G̈Ob̈utton is clicked, the user will be presented the floor map of the start location he chose, a little wizard riding a broom is placed on the map indicating the location of the start room, as shown in Figure . There are many labels on the map (e.g. bathroom labels, the Tim Hortons labels, bus stop labels, printer labels) to help users to picture the environment in their brains better.

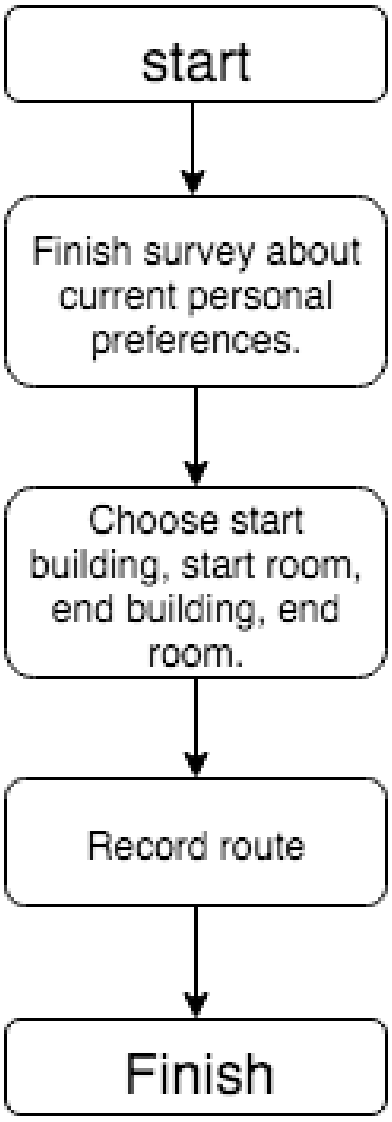


Figure 4. Workflow of Recordroute Interface

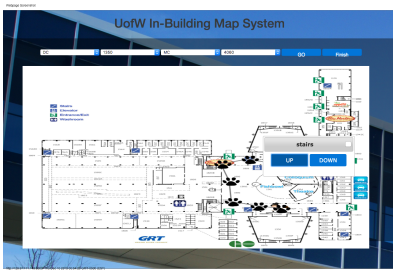


Figure 5. Personal Preference Page

Users can start drawing their routes by simply clicking on the map. Their first step shall be the start room. Each time they clicked, a foot print will be drawn on the map. Neighboring steps will have connected foot prints, as shown in Figure .

Stairs and elevators are also labeled on the maps. If our users need to go up stairs or down stairs, they can click the stairs or

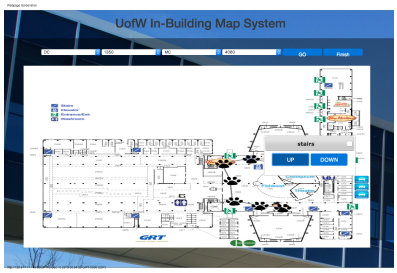


Figure 6. Welcome Page

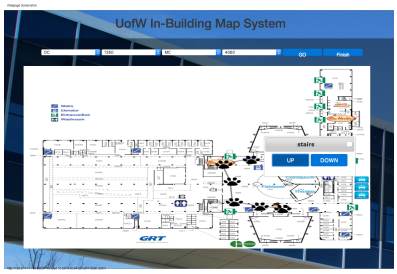
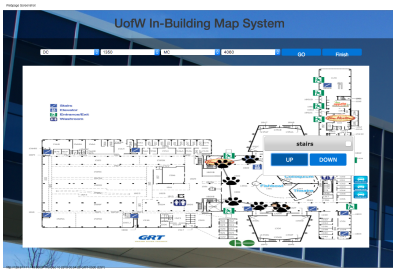
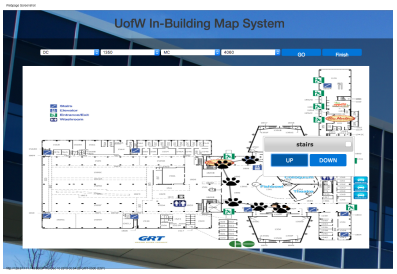
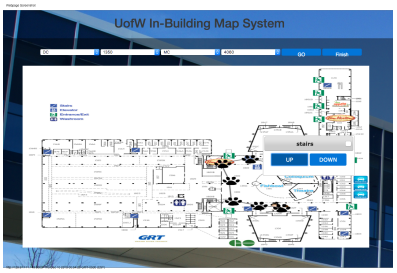


Figure 7. Choose Page

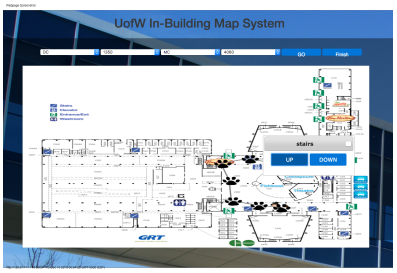




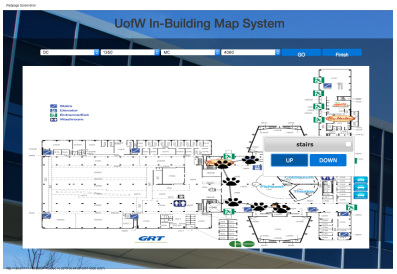
the elevators on the map. A popup dialog will be presented when those labels are clicked. Users can choose to go up, down or just close the dialog in the dialog, as shown in Figure .



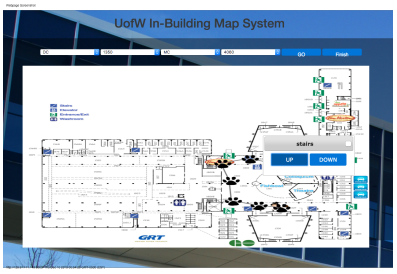
The corresponding floor map will be presented on the webpage if the users clicked going up or going down. Floor maps in the same building are aligned and in the same scale. Therefore, users can easily find out the stairs or the elevator they come from. Users can keep drawing routes on the new floor map by clicking on the map, as shown in Figure .



Exits and entrances are labeled on the maps as well. If our users need to exit the building ,they can click the exit on the map. The same as above, a popup dialog will be presented and users can choose to exit the building, enter the building or just close the dialog in the dialog, as shown in Figure .

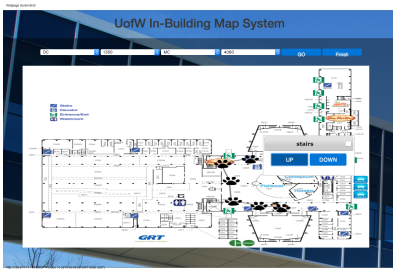


So far, our system support MC building and DC building. Therefore, we already calculated which floor in which building users can get after exiting each of the exits (e.g. users will get the third floor of MC building after exiting the exit on the second floor of DC building). When the users click the exit button on the dialog, they will be presented the corresponding floor map of the expected floor in the expected building. Users can adjust the floor by simply click the stairs labels if they found they are not on their expected floors. When get to the expected floor, users will need to specify the entrance they use to enter the building by click on the entrance label on the map and choose ënterön the popup dialog. Once they entered the building, they can continue to draw maps as before, as shown in Figure .



After the usrs arrived their destination, they will click the finish button on the right side of menus. Once the finish button

is clicked, the route is recored in the server and the users will be presented a thank you on the webpage, as shown in Figure



Seachroute Interface

This interface is used for search routes from one room location to another across building groups with personal preferences. Our users need to finish the survey about their current personal preferences first, and then choose the start location and end location. After that, the route will be presented on the webpage. The basic workflow of the recordroute interface is shown in Figure 8.

Same as above, users will first be presented the personal preferences page, as shown in Figure 5, and finish the perference survey. At the bottom of this page, there is a S̈earchb̈utton. After finishing the survey, users will click the button. Then users will be nevigated to the route presenting page, as shown in Figure 6.

Users will choose their start building, start room, end building, end room from the top menus on the page, as shown in Figure 7, and click G̈Ob̈utton on the right side of the menu.

Once the G̈Ob̈utton is clicked, the webpage will send the server the request and the server will reply the webpage with the proper route. Then the user will be presented the floor map of the start location he chose, with connected footprints on it, as shown in Figure . There are also many labels on the map (e.g. bathroom labels, the Tim Hortons labels, bus stop labels, printer labels) to help users to picture the environment in their brains better.

Our users will follow those footprints on the map. If the footprints on the current floor map end up on an stairs label or elevator label, the users can click on that label and a popup dialog will be presented, as shown in Figure . Users can choose to go up or down stairs in the dialog. However, this time only the correct direction will be responded. That is to say, if the planned route goes upstairs from the second floor of DC to the first floor of DC, the map will not change if users click go upstairs on the second floor.

If it is the correct direction of stairs or elevators, the corresponding floor map as well as the footprints will be presented, as shown in Figure .

The same rules apply on exits and entrances as well, as shown in Figure and Figure .

EXPERIMENT

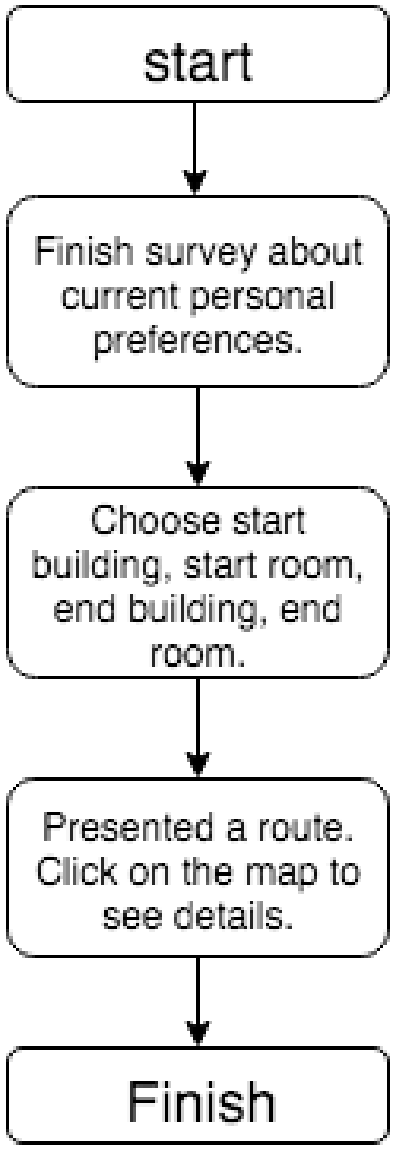
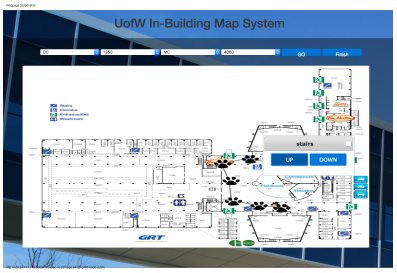


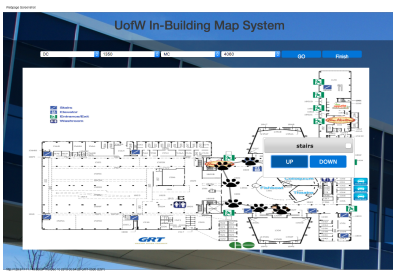
Figure 8. Workflow of Seachroute Interface

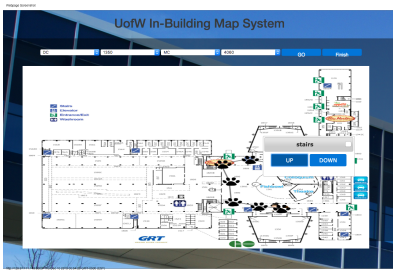


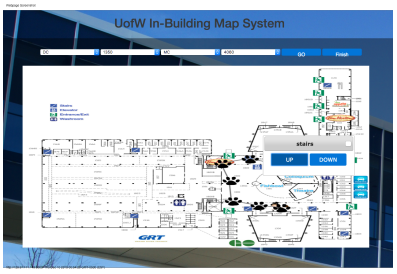
In this section, we explain how we execute the crowdsourcing to collect data for training the model based on decision tree algorithms.

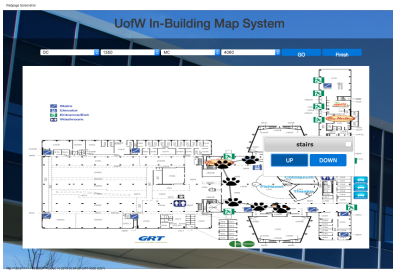
Experiment Setup

Participants









Since drawing a route between two specific rooms in the campus of University of Waterloo requires a relatively high degree of familiarity of the campus. Thus, the participants are UW students. 48 students in total participated in our experiment. These students have been UW for at least 2 months.

Source and Destination

To simplify our experiment, currently, we only support routes from DC 1350 to MC 4060. All of the participants are familiar with these two buildings. Since the number of participants are limited, supporting more source and destination pairs may reduce the data for each pair.

Experiment Methodology

The experiment consists of two steps:

Step 1: Take a survey about the current status of the participant and the environment conditions. All participants are asked to answer all questions listed in table-XX (see modeling procedure). All these data is sent to the server and associated with a route ID.

Step 2: Draw a route from DC 1350 to MC 4060. The users are asked to plan the route according to their answers to the survey taken in step 1. The coordinates of points the user click on the map will be sent to the server as the route data. The route data is also associated with the route ID in step 1.

After these two steps, all data collection work has been finished. In summary, for each user, the data we collect includes:

- Survey answers
- Routes data

Step 3: Derive physical properties from the data of each route. PhysicalPropertyDecisionTrees are generated based on the physical properties and survey answers. RouteSuggestionDecisionTree is generated based on the physical properties and route data.

RESULT

In our experiment, we generate 8 decision trees. 7 of them are for predicting physical properties based on a preference vector. The other one left is for predicting the route number based on a physical property vector.

The graph for all decision trees can be found below. There are several clarifications about the graph.

- For each non-leaf node of a decision tree, the first line is the condition to be checked.
- The gini value of each node is positive correlative to the probability of wrong prediction.
- The value of samples means how much samples are left after following all the path from the root to the current node.
- Each element in the value array means the number of samples of the corresponding class.
- The value of class means the prediction value. This value is only shown in the routeSuggestionDecisionTree.

Result Graphs of PhysicalPropertyDecisionTree

Figure 9 shows the decision tree for predicting the number of stairs the route should use based on the users preference.

Figure 10 shows the decision tree for predicting how many levels of elevators the route should use.

Figure 11 shows the decision tree for predicting how many exits the route should pass.

Figure 12 shows the decision tree for predicting if the route should pass by a coffee shop.

Figure 13 shows the decision tree for predicting if the route should pass by a rest room.

Figure 14 shows the decision tree for predicting how long if the route.

Figure 15 shows the decision tree for predicting if the route goes outside the building.

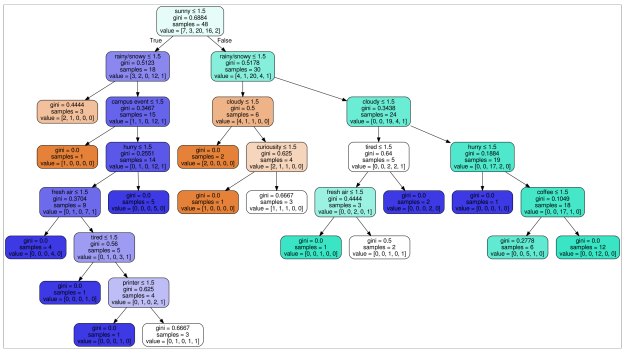


Figure 9. Decision tree for predicting the number of stairs the route should use based on the users preference.

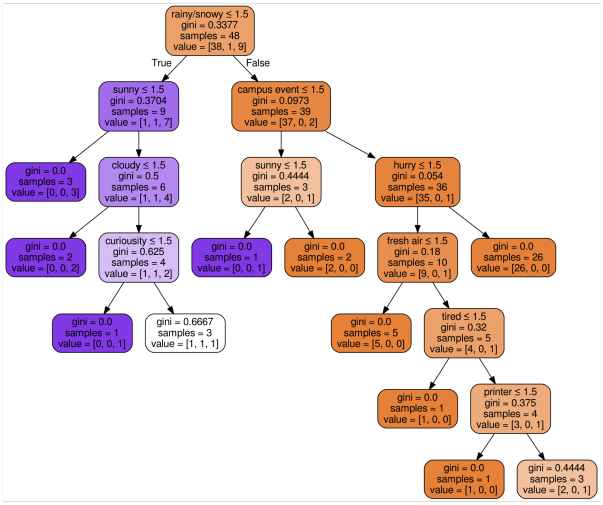


Figure 10. Decision tree for predicting how many levels of elevators the route should use.

Result Graph of RouteSuggestionDecisionTree

Figure 16 shows the decision tree for predicting which route should be picked.

DISCUSSION

From the routeSuggestionDecisionTree, we can see that only a few of the routes are picked. These means that the number of potential routes is quite small since there are only limited possible routes. Besides, one route can fulfill many physical requirements. We think there are more physical properties which are useful. Also, it will be great if we can support more sources and destinations.

At the beginning, we asked the participants to draw the route before answering the survey questions. However, according to the participants, it will be more difficult for them to recall how they fill when they draw the route. Besides, they think taking the survey at first will be better. We adapt their suggestions and update our project to the current version.

Besides of the decision tree algorithm we used, the k-NN algorithm also seems to meet our requirements and is better

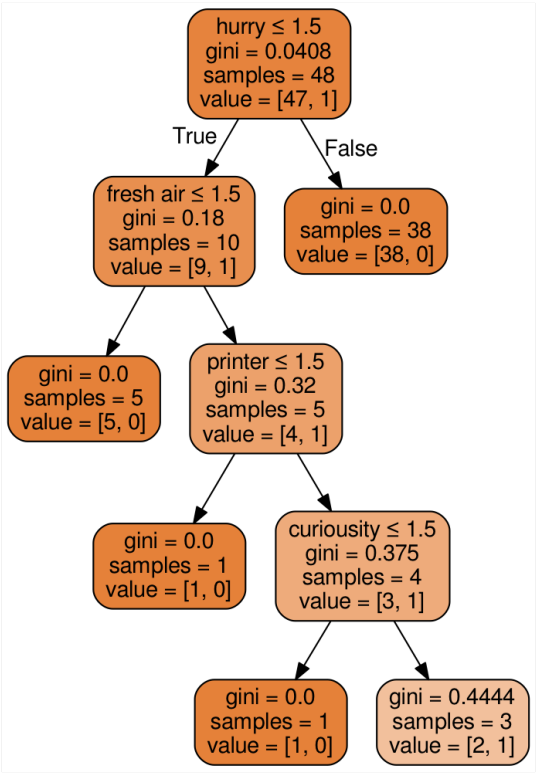


Figure 11. Decision tree for predicting how many exits the route should pass.

when there are more features to be considered. It will be a good idea to implement a k-NN classifier for route suggesting problem.

ACKNOWLEDGMENTS

We thank everyone in CS889.

REFERENCES

1. Alt, F., Shirazi, A. S., Schmidt, A., Kramer, U., and Nawaz, Z. Location-based crowdsourcing: extending crowdsourcing to the real world. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, ACM (2010), 13–22.

2. Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and regression trees*. CRC press, 1984.

3. Della Mea, V., Maddalena, E., and Mizzaro, S. Crowdsourcing to mobile users: A study of the role of platforms and tasks. In *DBCrowd* (2013), 14–19.

4. Heimerl, K., Gawalt, B., Chen, K., Parikh, T., and Hartmann, B. Communitysourcing: engaging local crowds to perform expert work via physical kiosks. In *Proceedings of the SIGCHI Conference on Human*

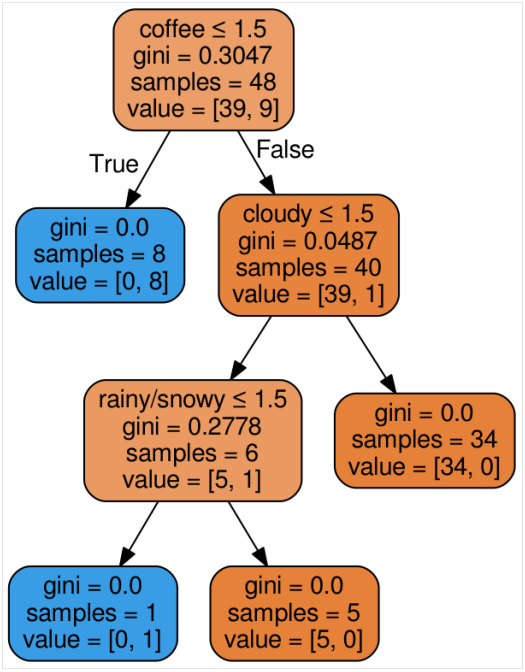


Figure 13. Decision tree for predicting if the route should pass by a rest room..

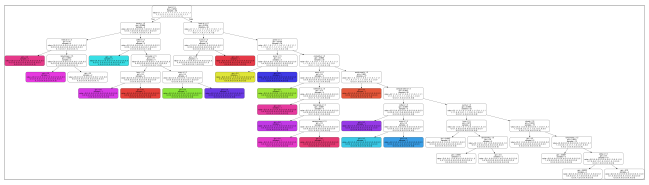
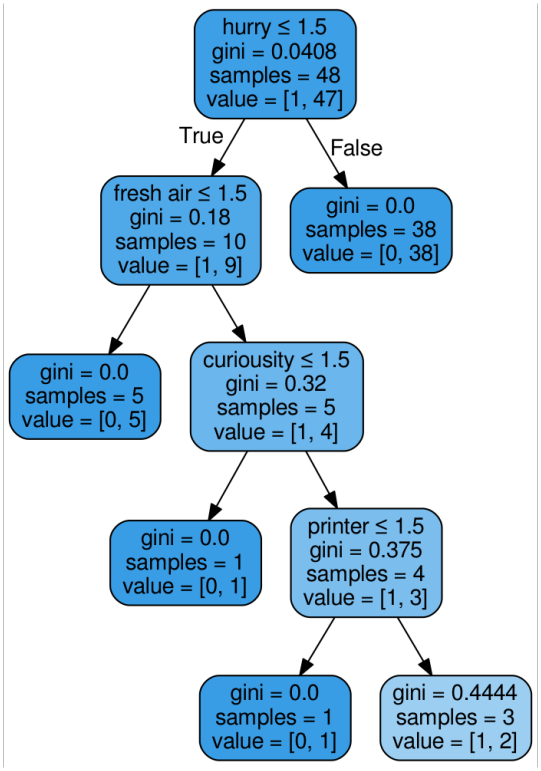


Figure 14. Decision tree for predicting how long if the route.



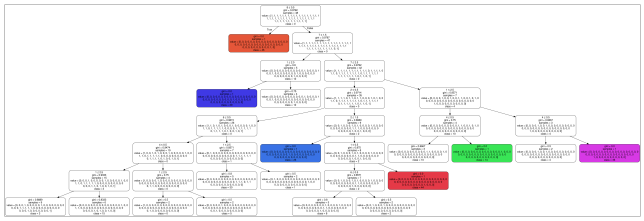


Figure 16. Decision tree for predicting which route should be picked.

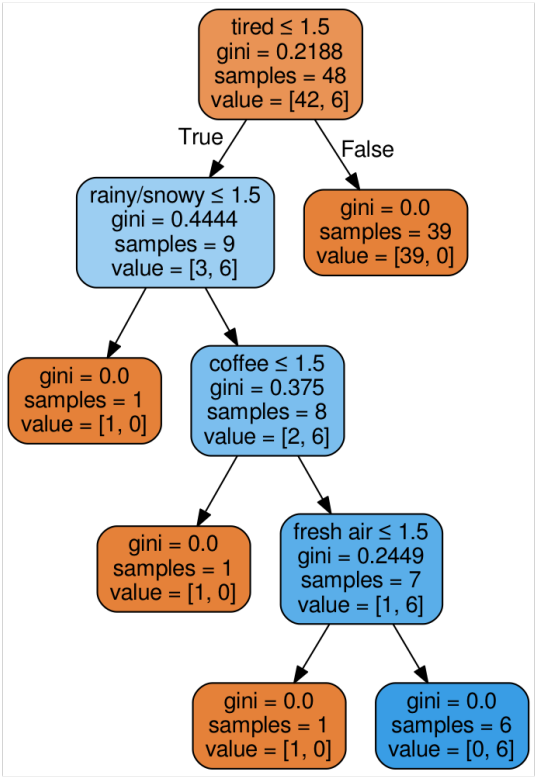


Figure 12. Decision tree for predicting if the route should pass by a coffee shop.

Factors in Computing Systems, ACM (2012), 1539–1548.

5. Howe, J. The rise of crowdsourcing. *Wired magazine 14*, 6 (2006), 1–4.

6. Huang, A., Pulli, K., and Rudolph, L. Kimono: kiosk-mobile phone knowledge sharing system. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, ACM (2005), 142–149.

7. Ipeirotis, P. G., and Gabrilovich, E. Quizz: Targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World wide web*, ACM (2014), 143–154.

8. Quinlan, J. R. Induction of decision trees. *Machine learning 1*, 1 (1986), 81–106.

9. Quinlan, J. R. Simplifying decision trees. *International journal of man-machine studies 27*, 3 (1987), 221–234.

10. Su, J., and Zhang, H. A fast decision tree learning algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2006), 500.

11. Väätäjä, H., Vainio, T., Sirkkunen, E., and Salo, K. Crowdsourced news reporting: supporting news content creation with mobile phones. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ACM (2011), 435–444.