

Toward Blind Travel Support through Verbal Route Directions: A Path Inference Algorithm for Inferring New Route Descriptions from Existing Route Descriptions

Vladimir Kulyukin^{*1} and John Nicholson²

¹Department of Computer Science, Utah State University, USA

²Department of Computer Science, Austin Peay University, USA

Abstract: The work presented in this article continues our investigation of such assisted navigation solutions where the main emphasis is placed not on sensor sets or sensor fusion algorithms but on the ability of the travelers to interpret and contextualize verbal route directions en route. This work contributes to our investigation of the research hypothesis that we have formulated and partially validated in our previous studies: if a route is verbally described in sufficient and appropriate amount of detail, independent VI travelers can use their O&M and problem solving skills to successfully follow the route without any wearable sensors or sensors embedded in the environment.

In this investigation, we temporarily put aside the issue of how VI and blind travelers successfully interpret route directions en route and tackle the question of how those route directions can be created, generated, and maintained by online communities. In particular, we focus on the automation of path inference and present an algorithm that may be used as part of the background computation of VGI sites to find new paths in the previous route directions written by online community members, generate new route descriptions from them, and post them for subsequent community editing.

Keywords: Independent blind travel, path inference, informat.

1. INTRODUCTION

Commercial and research systems have been developed to increase travel independence for visually impaired (VI) and blind travelers. Various technologies, including GPS [1, 2], Wi-Fi localization [3, 4], and infrared beacons [5], have been proposed. Cost and quality of service have been adequately discussed in the literature and found to play a significant role in the slow or low adoption rates of these technologies [6-10]. Another contributing factor is the *trust-me-you-are-here* approach: assisted navigation systems take readings from their sensor sets, localize those readings on some maps, and instruct travelers on where to move next to reach destinations.

Unfortunately, sensor readings can be noisy, irrelevant, or absent. Garmin (www.garmin.com) reports that its GPS units are accurate to within fifteen meters. Place Lab [11], a Wi-Fi indoor solution, achieves a median location error of fifteen to twenty meters. Since navigation can be viewed as a time series that unfolds as the traveler moves through an environment, sensor signal latency may render automated decisions irrelevant when they are based on the sensor readings that are no longer representative of the traveler's actual location [12]. Such automated decisions, although they purport to help the traveler, may in reality hinder the traveler's situational awareness [13].

Golledge, Klatzky, and Loomis [14] demonstrate that sighted individuals are able to "process data in a continuous, integrative, and gestalt-like manner." Sighted travelers localize on routes through large-scale geographic observations. For example, while traveling an outdoor route, a sighted traveler notices that the next sidewalk intersection is approximately fifty feet away from his current location and that he is walking towards the main clock tower on campus, which happens to be hundreds of feet away.

The same researchers [14] argue that, unlike sighted travelers, travelers with visual impairments, especially those with complete vision loss, "actively search the environment in a piecemeal manner." For example, a cane user with complete vision loss must physically encounter the same sidewalk intersection to actively sense it with the cane and may not be aware of his orientation in relation to the clock tower. Consequently, erroneous or irrelevant assistance may be more harmful than helpful to this traveler, because it takes the traveler's cognitive energy away from critical decision making.

One sensor that many *trust-me-you-are-here* solutions attempt to replace or tend to downplay is the traveler's brain. Many VI and blind people receive extensive O&M training. During training, individuals learn valuable skills that enable them to safely and successfully navigate many indoor and outdoor environments independently [13, 14]. They learn to perform actions such as following sidewalks, detecting obstacles and landmarks, and crossing streets. They also learn how to remain oriented inside buildings or on sidewalks and streets. Many individuals subsequently improve their O&M skills through independent traveling and

*Address correspondence to this author at the Department of Computer Science, Utah State University, Logan, UT, USA; Tel: 435.787.8163; Fax: 435.797.3265; E-mails: vladimir.kulyukin@usu.edu, vladimir.kulyukin@aggiemail.usu.edu

acquire new wayfinding skills specific to their life experiences.

The work presented in this article continues our research of such assisted navigation solutions where the main emphasis is placed not on sensor sets or sensor fusion algorithms but on the ability of the travelers to interpret and contextualize verbal route directions received en route. This work contributes to our investigation of the research hypothesis formulated and partially validated in our previous studies [9, 15-17]: *if a route is verbally described in sufficient and appropriate amount of detail, independent VI travelers can use their O&M and problem solving skills to successfully follow the route without any wearable sensors or sensors embedded in the environment.*

In this article, we investigate the question of how these route directions can be automatically generated from existing route directions. In particular, we focus on the automation of path inference and present an algorithm that may be used as part of the background computation of VGI sites to find new paths in the previous route directions written by online community members, generate new route descriptions from them, and post them for subsequent community editing.

Our article is organized as follows. In Section 2, we review current Web map services and our previous research on community route information sharing. Section 3 discusses automated landmark autotagging in written route descriptions. While landmark autotagging is not the focus of this article, it is an important step in the automated route analysis done by the path inference algorithm presented in this article. Section 4 gives a detailed path inference example. Section 5 shows how route descriptions with tagged landmarks are converted into a directed weighted graph. Section 6 presents a detailed description of the path inference algorithm. In Section 7, we give examples of the paths that were inferred by our algorithm and the corresponding route descriptions that were generated from the inferred paths. Section 8 places our research in the context of related work in natural language processing (NLP) and information extraction (IE).

2. MAP SERVICES

Passini and Proulx [18] show that VI travelers make more decisions en route and use more information than sighted travelers. The Internet abounds in map-oriented sites and services, e.g., Google Maps (maps.google.com), MapQuest (<http://www.mapquest.com/>), and Yahoo! Maps (maps.yahoo.com). While these websites are useful to sighted travelers, VI travelers are generally unable to use these sites, because much of the information is presented visually.

Google Maps has addressed this problem to a degree. When using the mobile version of Google's route direction service at <http://www.google.com/m/directions>, the user is presented with a simpler, text-based user interface that works with screen readers and allows users to receive text-only route directions. Unfortunately, these directions are not yet adequate for the needs of VI travelers. For example, when a warning "Use caution - This route may be missing sidewalks or pedestrian paths" is issued, the segments with the missing features are not textually identified. No information is given

on turn locations, one-way or two-way streets, street intersections, and stop signs or traffic lights.

Another limitation of current map services is lack of information for indoor environments. To the best of our knowledge, one exception is the site www.clicandgomaps.com that offers a narrative mapping service to render public facilities such as schools, airports, and hotels accessible to blind and deafblind travelers through route descriptions written by professional O&M instructors. Another promising approach is <http://redpin.org/>, an open source indoor positioning system that seeks to omit the time-consuming training phase and rely on travelers to train the multi-sensor (GSM, Bluetooth, and Wi-Fi) system while using it in specific environments.

2.1. Volunteered Geographic Information

Volunteered geographic information (VGI) [19] is a bottom-up volunteer movement that sprang up to address shortcomings of commercial mapping services discussed in the previous section. VGI communities seek to provide free community-based GIS tools, data, and websites. VGI sites encourage volunteers to provide GIS data. Many volunteers upload small pieces of appropriate information. A VGI website's back-end software, over a period of time, assembles these pieces into cohesive data sets. Although the data quality may be inferior to that provided by professional services, VGI sites often contain unique data not available elsewhere. For example, volunteers who live in a particular area are more familiar with that area than professional GIS cartographers located elsewhere.

Many VGI sites provide the same services as professional sites. OpenStreetMap [20] is a VGI site for creating large map data sets. Such data sets are similar to those of commercial sites such as Google Maps in that both use vector map data representations. Google, because it uses commercial sources for its map data, offers satellite images unavailable in OpenStreetMap. However, OpenStreetMap features sidewalks and lesser known buildings. Wikimapia (<http://wikimapia.org>) is a VGI site that enhances Google Maps with dynamic annotations. Trailpeak (<http://www.trailpeak.com>) is a VGI site where users add, edit, view, and download free-text trail descriptions, trail directions, reviews, and GPS waypoints related to trails for hiking, mountain biking, and kayaking in Canada and the U.S.

A common strength of the VGI sites is the local user knowledge of GIS data. VGI site volunteers, because of their local experience and knowledge, create data sources that can be as credible as traditional sources. VI travelers also possess credible sources of route knowledge for the areas where they live, work, or travel. Their expertise is a combination of familiarity with routes and a deep understanding of VI travelers' cognitive and O&M skills. If this knowledge could be shared, it may be possible for VI travelers unfamiliar with the area to use this knowledge for traveling unfamiliar routes.

2.2. Community Route Information Sharing

Our previous research on independent blind travel offers some evidence that, given sufficient context-sensitive verbal assistance, independent VI and blind travelers can navigate

complex and dynamic environments such as supermarkets [15, 16] and college campuses [21]. There exists research evidence that VI travelers successfully share route descriptions with each other *via* spoken language [22], especially *via* smartphones.

We have previously proposed a Community Route Information Sharing (CRIS) framework for VGI sites to capture and share the local expertise of VI travelers [17]. CRIS is intended to provide user-adjusted levels of information for independent VI travelers to follow routes in indoor and outdoor environments *without* external sensors. External sensors, when available, can still be put to use. The simplest implementation of a CRIS site is a Wiki where users add, create, and edit data, resulting in large and dynamically managed repositories of knowledge.

Since many VI travelers perceive the world in terms of routes rather than layouts [14], CRIS knowledge bases will likely consist of large collections of route descriptions. The route descriptions cover routes that are entirely indoors, entirely outdoors, or a mixture of both. As the collection of route descriptions for a given area grows, the volunteer community creates and subsequently maintains route-based maps. All descriptions are written or, as the quality of online speech recognition improves, spoken in natural language (NL) and converted into free text. Information extraction (IE) is used to extract landmarks so that route descriptions are subsequently transformed into directed graphs. The graphs are searched for new paths. The new paths are used to generate new NL route descriptions that are added to the database for subsequent collaborative human editing.

A CRIS environment is represented as two data structures: a landmark taxonomy and a directed graph of route descriptions. A *landmark* is any location, object or sensation in a route description that can be detected through travelers' sensorimotor means. Landmark can be rooms, doors, hallway intersections, buildings, sidewalks, streets, smells (e.g., "when you smell bread baking"), and textures (e.g., "where you feel the carpet change to tile"). Landmarks are assigned unique integer IDs and may have alternative linguistic references. For example, the landmark "ROOM 414" representing room 414 in the Old Main building on the Utah State University (USU) campus could have the alternative names "COMPUTER SCIENCE FRONT OFFICE," "CS FRONT OFFICE," or "CS DEPARTMENT HEAD'S OFFICE." Landmarks in the taxonomy are organized in terms of the standard *part-of* relationship. Users can add and remove landmarks or change a landmark's place in the hierarchy.

Landmarks can be viewed as tags or metadata that give meaning to other data. Flickr (www.flickr.com) and YouTube (www.youtube.com) have user-created tags for image and video content, e.g., "ORIGAMI." One problem with the Flickr and the YouTube tags is their inconsistent naming structure and their flat namespaces. The tag "DOORS," for example, could be associated with a photo containing multiple wooden doors in a hall as well as a photo of the members of the classic rock band. For more information on our landmark hierarchy interested readers may refer to [17, 23].

The second data structure is a directed graph that encodes the current set of NL route descriptions. Route descriptions and statements have unique IDs. Each route description has start and end locations and a NL description that guides from the start location to the end location. The start and end locations reference landmarks in the hierarchy. The description is stored as a list of route statements. A route statement is a sentence from the user's route description. Users may tag route statements with landmarks. For example, one user may add a route description with the following route statement: "Turn right when you detect the intersection with the main hall." Another user may tag this route statement with the tag "WATER FOUNTAIN."

3. LANDMARK AUTOTAGGING

One set of techniques used to extract information from unstructured text into structured formats is Information Extraction (IE) [24, 25]. Since IE has been researched for over a decade, there are now IE frameworks that allow researchers to develop IE applications and tools for specific domains. In this section, only a brief description is given of how IE is used for landmark autotagging in our system. A detailed presentation and performance analysis are given in [23].

Landmark autotagging falls in the category of named-entity recognition (NER) [45]. NER modules identify text chunks that refer to specific categories such as names of persons, organizations, and locations. NER can be done with rule-based and statistical models [26, 27]. We did not use statistical NER, because it requires large annotated corpora. Our rules were developed manually on a small corpus of indoor and outdoor route descriptions written by VI travelers in an online survey [23]. The survey solicited two route descriptions from each respondent. The respondents were first asked to describe an outdoor route from the entrance of one building to the entrance of another building and then asked to describe an indoor route from one room in a building to another room in the same building. The instructions required that respondents describe real-world routes with which they were familiar. Respondents were also instructed to write the route descriptions as if they were describing the route to a fellow traveler with the same visual impairments and the same traveling experience and skills.

We received 52 responses, providing 104 English language route descriptions: 52 indoor route descriptions and 52 outdoor route descriptions. Two thirds of the route descriptions were placed in the training set for the purpose of analysis and IE rule development. One third was placed in an evaluation set to test the effectiveness of the IE rules. The outdoor routes contained approximately twice as many sentences and words as the indoor route descriptions: 18.5 *vs* 9.9 and 277.3 *vs* 148.7, respectively. The average number of words per sentence was consistent across all descriptions: 14.98 for indoor *vs* 14.99 for outdoor. The number of landmarks mentioned in outdoor route descriptions was approximately double that number in indoor descriptions: 37.1 *vs* 19.5. The number of landmarks remained consistent on a per sentence basis with 1.96 per indoor sentence *vs* 2.00 per outdoor sentence.

If our corpus had been larger, serious consideration could have been given to machine learning techniques for

automated rule creation [28]. However, since there were only 104 route descriptions, we decided to use the General Architecture for Text Engineering (GATE) system from the University of Sheffield [24, 25] for defining landmark autotagging rules. GATE's IE component is ANNIE (A Nearly-New IE system). ANNIE uses a Java-based pattern matching language (JAPE) [24] for writing patterns to extract matching information from text. JAPE is based on the Common Pattern Specification Language (CPSL) [29]. Interested readers are referred to [25] for details on GATE, ANNIE, and JAPE.

ANNIE consists of several processing resources (PR): 1) Tokenizer; 2) Gazetteer; 3) Sentence Splitter; 4) Part-of-speech (POS) Tagger; 5) Named Entity Transducer; and 6) Orthomatcher. The Tokenizer splits a text into basic tokens such as numbers, words, and punctuation. The tokenized text is passed to the Gazetteer to find well-known entities, e.g., the names of all the employees in a company. The text is then passed to the Sentence Splitter for sentence location. The POS Tagger assigns parts of speech to each token in each located sentence and passes the text to the NE Transducer to run JAPE rules for identifying named entities. Finally, the Orthomatcher is run to find co-references.

In our system, landmark autotagging is treated as in the NE subtask. Known entities are the landmarks that have been identified through the process of tagging or are included in the landmark hierarchy. Finding known entities is done in the Gazetteer. A file is created listing each possible string for a given entity type. Unknown entities, including unknown landmarks, are entities that cannot be derived from look-up lists. These also include words that are misspelled. If “*Old Main*” is in the Gazetteer look-up list but a user spells it as “*Old Mian*,” then the user's text is not annotated. Autotagging unknown entities was done by extending the JAPE rules for the NE Transducer. The NE Transducer relies on rules written in the JAPE language. Matches can be performed by specifying specific strings, previously assigned annotations, or tests on annotation attributes. Although sufficient in many cases, the JAPE syntax cannot handle complex annotation manipulations. For such more complex manipulations, arbitrary Java code can be written in a specific macro. For example, when processing the string “*John Angus Nicholson*” for names, ANNIE's Gazetteer creates three annotations: “*John Angus Nicholson*,” “*Angus*,” and “*Nicholson*.” This default behavior is modified to return the longest single match for these cases, i.e., “*John Angus Nicholson*.”

4. PATH INFERENCE EXAMPLE

A VI traveler may travel some routes without realizing that they share common landmarks. Suppose a VI USU student who is familiar with the campus describes a route to a new VI student. The route description describes a route (**R-A** in Fig. 1) that begins at the Animal Science building (**L-A**) and ends at the Ray B. West building (**L-C**). The route passes through the Quad, a large grassy area with two sidewalks (**L-B**). At a later date, the new student is given another route description (**R-B** in Fig. 2). This route starts at the Old Main building (**L-D**) and ends at the Distance Learning Center (**L-E**) in the Eccles Conference Center. Both **R-A** and **R-B** pass through the Quad's center at **L-B**.

The first part of **R-B** can be combined with the second part of **R-A** to form the new route **R-C** from Old Main to Ray B. West shown Fig. (3). Path inference is the process of inferring new, previously unknown routes from sets of previously known routes. **R-C** is inferred from the known routes **R-A** and **R-B** due to the landmark (**L-B**) shared by **R-A** and **R-B**. Path inference is a time saving utility, because the manual writing of route descriptions takes time, and an area such as a university campus requires many route descriptions for the coverage to support independent blind travel.

R-A Description: Exit the Animal Science building doors on the south side. Walk straight until you find the sidewalk entrance to the Quad's sidewalk. Pass the main sidewalk intersection. Walk south until you detect a road and then carefully cross the street. Continue to walk south until you find the doors to the Ray B. West building.

R-B Description: Exit Old Main walking east. You will walk through the Quad, passing the intersection. Keep walking straight until you run into grass and then turn left, walking north. Walk until you detect the bike racks on your right and then turn right. Walk east until you find the stairs leading to the entrance to the distance learning center.

5. TRANSFORMING TAGGED DESCRIPTIONS INTO DIGRAPHS

Path inference starts after landmarks are labeled in route statements though autotagging. Tagged route descriptions are transformed into a digraph. Given a starting landmark and an end landmark, the digraph is searched for a path between them. If a path is found, a route description is built and given to the community for collaborative editing. If no path is found, its absence may signify that more route descriptions may be needed.

Digraphs consist of *statement nodes* and *landmark nodes*. Digraphs are used because landmarks and statements in a description of a route from **A** to **B** may not be identical to the landmarks from in a description of a route from **B** to **A**. For example, when walking down a hall from **A** to **B**, a traveler may trail the right hand wall and encounter a water cooler and a soda machine. When walking from **B** to **A** and trailing the wall on the other side of the hall, the traveler may encounter only three office doors.

Each route statement is represented by a statement node. When a statement node **SN1** is connected to another statement node **SN2** in the graph by a *precedence* edge, it means that the action described in the statement that corresponds to **SN1** must be executed before the action described in the statement that corresponds to **SN2**. If a route statement is associated with a set of landmarks, each landmark becomes a landmark node. A statement node and landmark nodes are connected *via association* edges. A precedence edge can also connect a landmark node and a statement node. This redundancy is useful in handling collaborative route editions. For example, if **SN1** precedes **SN2** and **SN1** is deleted, the landmarks associated with **SN1** are not automatically deleted as they may be associated with other statement nodes. These landmarks still precede **SN2** in the sense that the traveler will experience them before performing the action specified in **SN2**.

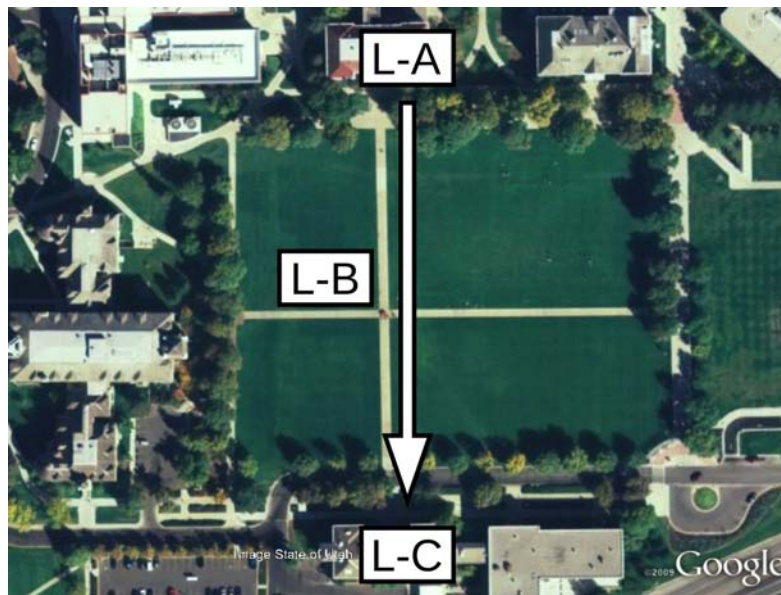


Fig. (1). Example Route **R-A**.

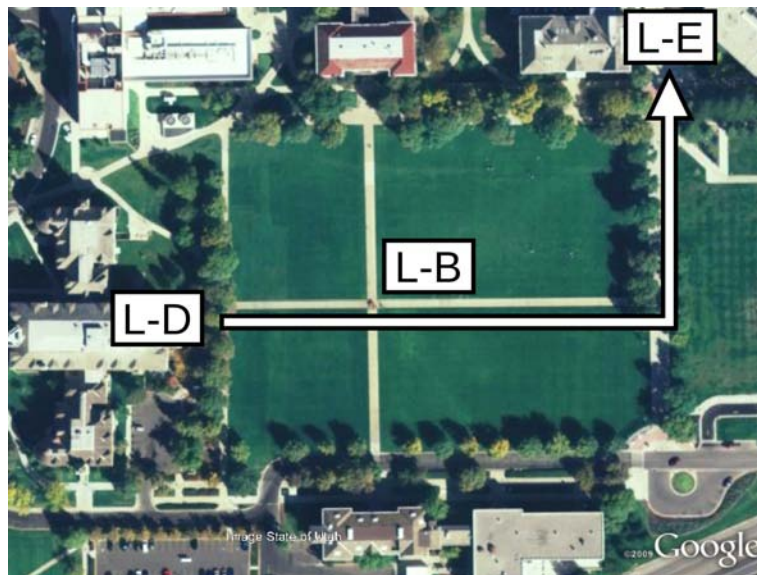


Fig. (2). Example route **R-B**.

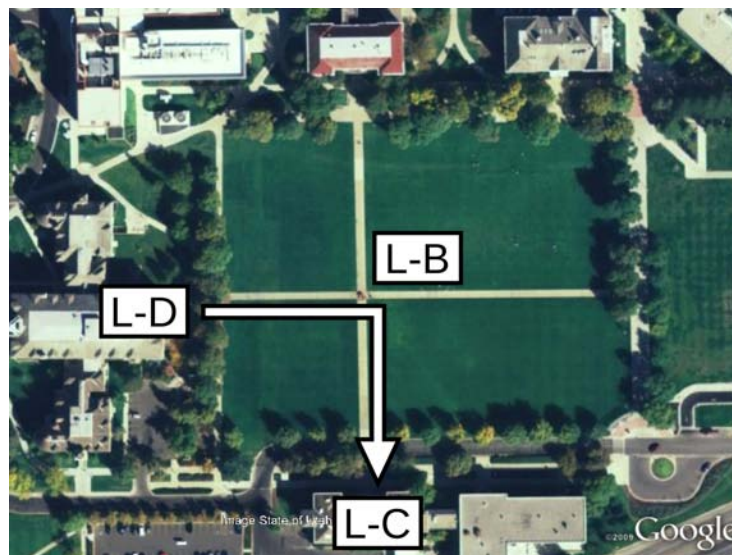


Fig. (3). Route **R-C** inferred from **R-A** and **R-B**.

Route ID: R-3

Start Landmark: Landmark(name="QUICK STOP", id="L-500")

End Landmark: Landmark(name="THE HUB", id="L-789")

Statement List

1. Statement

id: "S-3-1"

text: "You are standing with your back to the south entrance to the Quick Stop."

landmarks: Landmark(name="SOUTH ENTRANCE", id="L-501")

2. Statement

id: "S-3-2"

text: "Turn left so you are walking east."

landmarks: NIL

3. Statement

id: "S-3-3"

text: "ATM machines which make distinctive sounds, and the campus post office and mailbox."

landmarks:

Landmark(name="ATM", id="L-550"),

Landmark(name="POST OFFICE", id="L-551"),

Landmark(name="MAILBOXES", id="L-552")

4. Statement

id: "S-3-4"

...

Fig. (4). Partial tagged route description.

Consider the following route description written by a VI USU undergraduate student. It describes an indoor route from the Quick Stop, a small convenience store, to the Hub, one of the main areas on campus for buying meals. The route occurs entirely in the Taggart Student Center. The beginning of the tagged route description is shown in Fig. (4). The resulting digraph of this tagged description is shown in Fig. (5). The figure contains a pentagon node (**R-3**) representing the start of the route and a connection from that node to the route's start landmark (**L-500**). The final digraph also contains edge weights (not displayed in Fig. 5) discussed in Section 6.1.

You are standing with your back to the south entrance to the Quick Stop. Turn left so you are walking east. On your left you will pass the ATM machines which make distinctive sounds, and the campus post office and mailbox. You will pass the entrance to the financial aid office on your right and several bulletin boards. Continue walking east and passing offices, the barber shop, and the copy center as you walk down this long hall. Towards the eastern end of the building, you will come to a wide open area on your left. Turn left and walk a little north. Pass Taco Time on your left, and look for a small opening on your left. This opening will have a cashier counter on your right. Turn left and enter the world of the Hub. You will find a wide variety of food stations around a semicircle.

The algorithm that transforms a set of tagged route descriptions to a digraph is shown in Fig. (6). The graph's set of vertices, **V**, contains statement and landmark nodes,

each of which has a unique ID. The loop at line 17 connects a route statement to all its landmark tags. If the route statement does not have landmark tags, it is connected to the next route statement or end landmark if it is the last route statement. The check at line 19 ensures that an edge from the first route statement to the start landmark is not created. The start landmark is set as the first node for a route in line 5 and is always connected to the route's first statement in the for-loop at line 10. The final check at line 30 ensures that the end landmark is added. If it is added, this signifies that the end landmark is one of the landmarks in the set of tags for the route's final route statement node. If the end landmark is not one of the final route statement's tags, this check adds a connection between the statement node and the end landmark.

The function **BUILD_GRAPH()** uses the function **CONNECT()** (see in Fig. 7) to create the weighted edges. Whenever there is a precedence edge that connects the start landmark of a route to the first route statement, the edge's weight is set to **START_ROUTE_COST**. This weight signifies the start of the route and is higher than the other weights. Fig. (8) shows the digraph created by **BUILD_GRAPH()** and **CONNECT()** from the tagged description in Fig. (5).

6. PATH INFERENCE

The constructed digraph is used to find a path from one landmark node to another landmark node. If a path is found, the statement nodes along the path are combined into a new route description. Fig. (9) shows a digraph constructed from Tables 1 and 2. Table 1 gives the landmark IDs and names; Table 2 associates each route statement with its route ID and tagged landmarks. Table 2 is the output of the auto-tagging algorithm. Route description **R-1** describes the route from

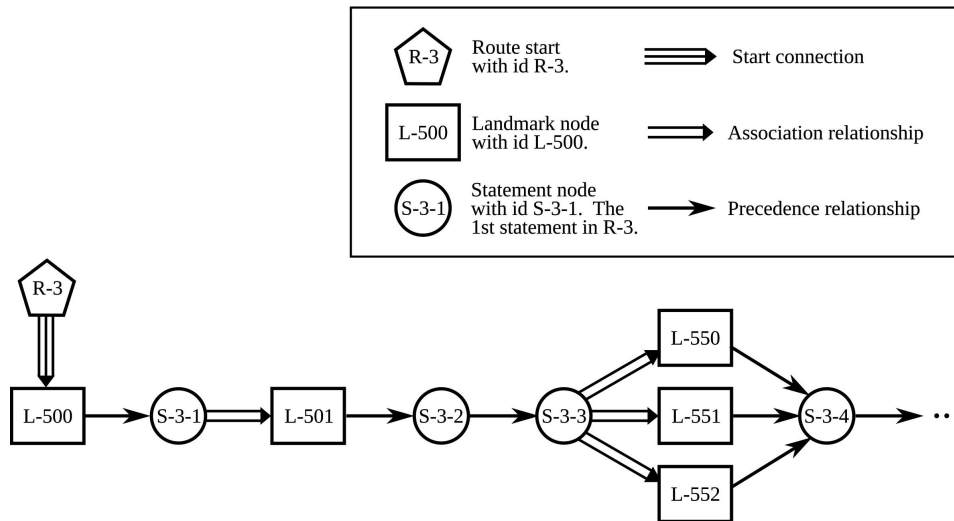


Fig. (5). Partial transformation of tagged description into digraph.

Function BUILD_GRAPH(*TaggedRouteSet*)

```

1  foreach route ∈ TaggedRouteSet do
2  | startLandmark ← route's start landmark;
3  | endLandmark ← route's end landmark;
4  | statements ← route's list of route statements;
5  | V[G] ← V[G] ∪ {startLandmark};
6  | previousNodes ← {startLandmark};
7  | for statementId = 1 to length(statements) do
8  | | statement ← statements[statementId];
9  | | if statementId = 1 then
10 | | | CONNECT(E[G], previousNodes, statement, true);
11 | | | else
12 | | | | CONNECT(E[G], previousNodes, statement, false);
13 | | | end
14 | | | V[G] ← V[G] ∪ {statement};
15 | | | previousNodes ← {statement};
16 | | | nextNodes ← ∅;
17 | | | foreach landmark ∈ statement's set of landmarks do
18 | | | | V[G] ← V[G] ∪ {landmark};
19 | | | | if statementId = 1 and landmark = startLandmark then
20 | | | | | nextNodes ← nextNodes ∪ {statement};
21 | | | | else
22 | | | | | nextNodes ← nextNodes ∪ {landmark};
23 | | | | | CONNECT(E[G], previousNodes, landmark, false);
24 | | | | end
25 | | | end
26 | | | if nextNodes ≠ ∅ then
27 | | | | previousNodes ← nextNodes;
28 | | | end
29 | | end
30 | | if endLandmark ∉ statement's set of landmark tags then
31 | | | V[G] ← V[G] ∪ {endLandmark};
32 | | | previousNodes ← {statement};
33 | | | CONNECT(E[G], previousNodes, endLandmark, false);
34 | | end
35 | end
36 return G;

```

Fig. (6). Algorithm for transforming tagged route description into digraph.

Old Main (L-1) to the Distance Learning Center (L-2), and route description R-2 describes the route from Animal Science (L-3) to Ray B. West (L-4). If no route description

exists in the database from Old Main to Ray B. West, path inference finds the path in the digraph from L-1 to L-4 via L-99, as shown in Fig. (3).

```

Function CONNECT(E, previousNodes, toNode, isFirstStatement)
1 foreach fromNode ∈ previousNodes do
2   if fromNode is a Statement and toNode is a Statement then
3      $E \leftarrow E \cup$ 
       PrecedenceEdge(fromNode, toNode, TO_STATEMENT_COST);
4   else if fromNode is a Landmark and toNode is a Statement then
5     if isFirstStatement then
6        $E \leftarrow E \cup$ 
         PrecedenceEdge(fromNode, toNode, START_ROUTE_COST);
7     else
8        $E \leftarrow E \cup$ 
         PrecedenceEdge(fromNode, toNode, TO_STATEMENT_COST);
9     end
10  else if fromNode is a Statement and toNode is a Landmark then
11     $E \leftarrow E \cup$ 
      AssociationEdge(fromNode, toNode, TO_LANDMARK_COST);
12  end
13 end
    
```

Fig. (7). Algorithm for creating weighted edges in digraph.

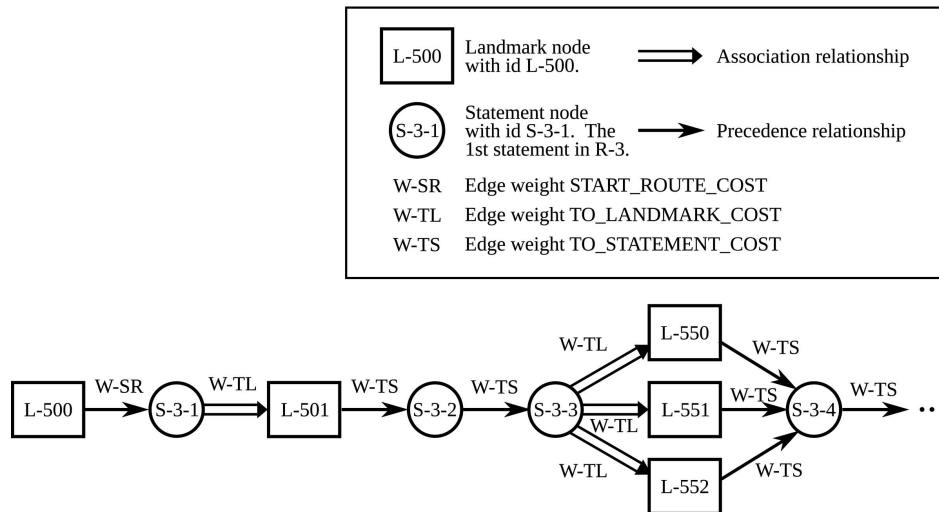


Fig. (8). Partial result of transforming tagged description into digraph.

To build a new route description, the statements along the path are joined to create a new route statement sequence. The original route statements are cloned, including their tags, and the new description and its route statements are given new IDs. Statement cloning is done so that the new route description can be edited without affecting the original route descriptions. In the above example, the new route description’s route statement sequence:

- S-3-1 (cloned from S-1-1): Exit Old Main walking east.
- S-3-2 (cloned from S-1-2): You will walk through the Quad, passing the intersection.
- S-3-3 (cloned from S-2-4): Walk south until you detect a road and then carefully cross the street.
- S-3-4 (cloned from S-2-5): Continue to walk south until you find the doors to the Ray B. West building.

Table 1. Example Landmark IDs for R-A and R-B

ID	Landmark Name
L-1	Old Main
L-2	Distance Learning Center
L-3	Animal Science
L-4	Ray B. West
L-99	Quad Sidewalk Intersection

The newly constructed route description requires the traveler to make a right turn at the Quad’s sidewalk intersection. However, statement S-3-2 does not mention a turn, but instead appears to instruct a traveler to continue walking straight with the phrase “passing the intersection.” This is an example of action inconsistency when actions from two different route descriptions result in inconsistency when the statements are joined in a new route description.

The path inference algorithm uses several heuristics for handling action inconsistencies described in the next Section.

Table 2. Example Route Statements and their IDs for Descriptions A and B

Route ID	Statement ID	Statement	Tags
A (R-2)	S-2-1	Exit the Animal Science building doors on the south side.	L-3
A (R-2)	S-2-2	Walk straight until you find the sidewalk entrance to the Quad’s sidewalk.	
A (R-2)	S-2-3	Pass the main sidewalk intersection.	L-99
A (R-2)	S-2-4	Walk south until you detect a road and then carefully cross the street.	
A (R-2)	S-2-5	Continue to walk south until you find the doors to the Ray B. West building.	L-4
B (R-1)	S-1-1	Exit Old Main walking east.	L-1
B (R-1)	S-1-2	You will walk through the Quad, passing the intersection.	L-99
B (R-1)	S-1-3	Keep walking straight until you run into grass and then turn left, walking north.	
B (R-1)	S-1-4	Walk until you detect the bike racks on your right and then turn right.	
B (R-1)	S-1-5	Walk east until you find the stairs leading to the entrance to the distance learning center.	L-2

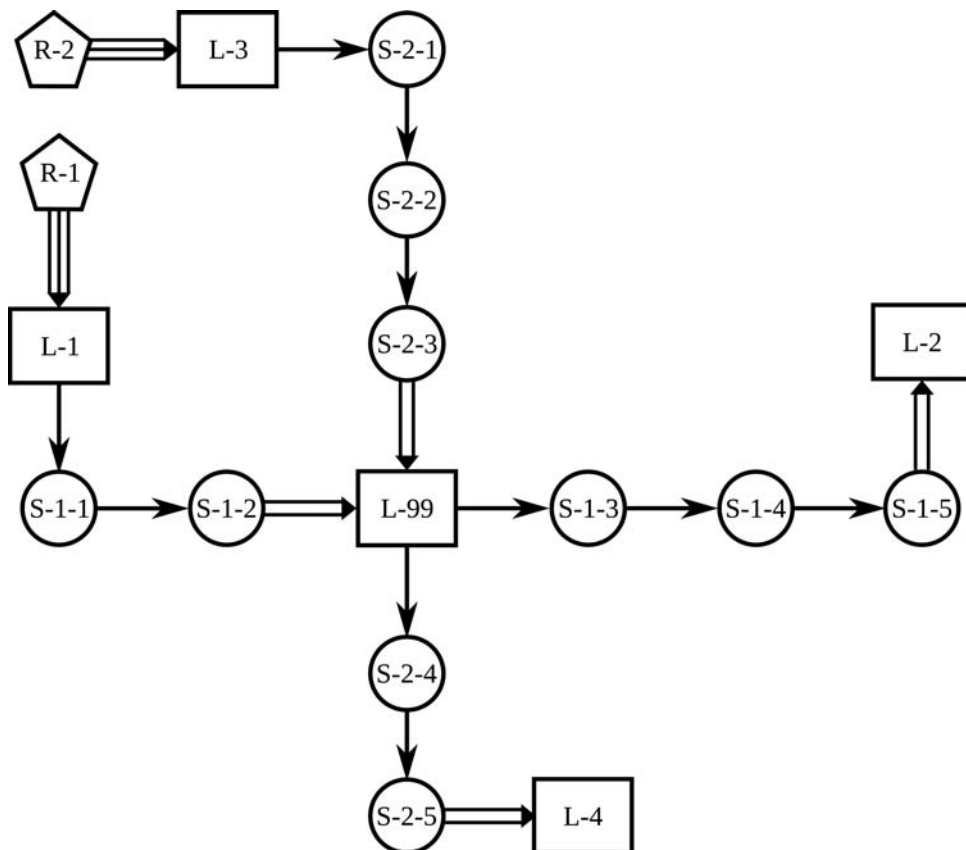


Fig. (9). Digraph transformation of descriptions IDs from Tables 1 and 2.

Since these heuristics do not guarantee action consistency, it is assumed that all new route descriptions constructed by the system will be passed to the community for subsequent collaborative editing to ensure route safety. Even if a stronger AI approach is taken to automate route editing, subsequent crowdsourcing *via* collaborative editing will likely be required. In the example above, knowledgeable community users may edit the route description by deleting the phrase “passing the intersection” and adding a new route statement “turn right at the sidewalk intersection” (S-3-3), which results in the following edited route description:

- S-3-1: Exit Old Main walking east.

- S-3-2: You will walk through the Quad.
- S-3-3: Turn right at the sidewalk intersection.
- S-3-4: Walk south until you detect a road and then carefully cross the street.
- S-3-5: Continue to walk south until you find the doors to the Ray B. West building.

6.1. Inference Heuristics

Choosing a path from which to build a new route description requires that choices be made regarding which path should be preferred over others. Three heuristics have

been designed to aid the system in choosing paths. Two of the heuristics address action inconsistency and the third uses the final length of the path as a metric.

The first action inconsistency heuristic, **H-1**, chooses paths put together from fewer route descriptions. The second heuristic, **H-2**, prefers to join route descriptions at their start and ending landmarks rather than at middle landmarks. Since action inconsistencies occur when a route statement from one route description is followed by a route statement from another route description, **H-1** minimizes the number of changes from one route to another when searching for a path in the digraph. Fig. (10) provides an example of how **H-1** works. The digraph in Fig. (10) has two possible paths for generating new routes from L-182 to L-428: the upper path from routes R-1, R-3, and R-4 that consist of route statements S-1-1, S-3-1, and S-4-1 and the lower path from

routes R-2 and R-4 that consists of route statements S-2-1, S-2-2, S-2-3, and S-4-1. The upper path has two potential locations for action inconsistencies: the transition from S-1-1 to S-3-1 and the transition from S-3-1 to S-4-1. The lower path has only one potential action inconsistency - the transition from S-2-3 to S-4-1. **H-1** chooses the lower path, because it uses paths from two routes, R-2 and R-4, not three as the upper path.

H-2 selects complete route descriptions over partial ones. Fig. (11) demonstrates how **H-2** is applied. The digraph in Fig. (11) has four routes R-11, R-12, R-13, and R-14. Suppose that a new route description from L-3 to L-429 is requested. There are two candidate paths. This first path starts with the first statement of R-12, S-12-1, and at L-99 switches to R-11's second statement S-11-2 and ends with S-11-3. This route description, when constructed, results in a

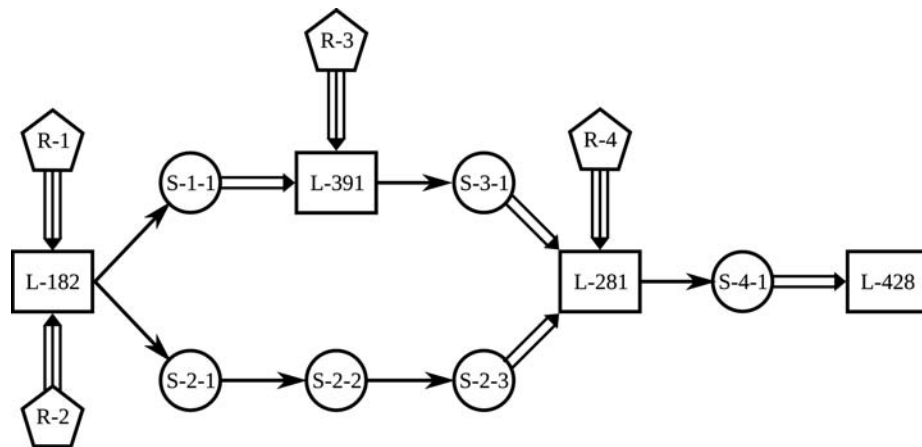


Fig. (10). Deciding the path from L-182 to L-428.

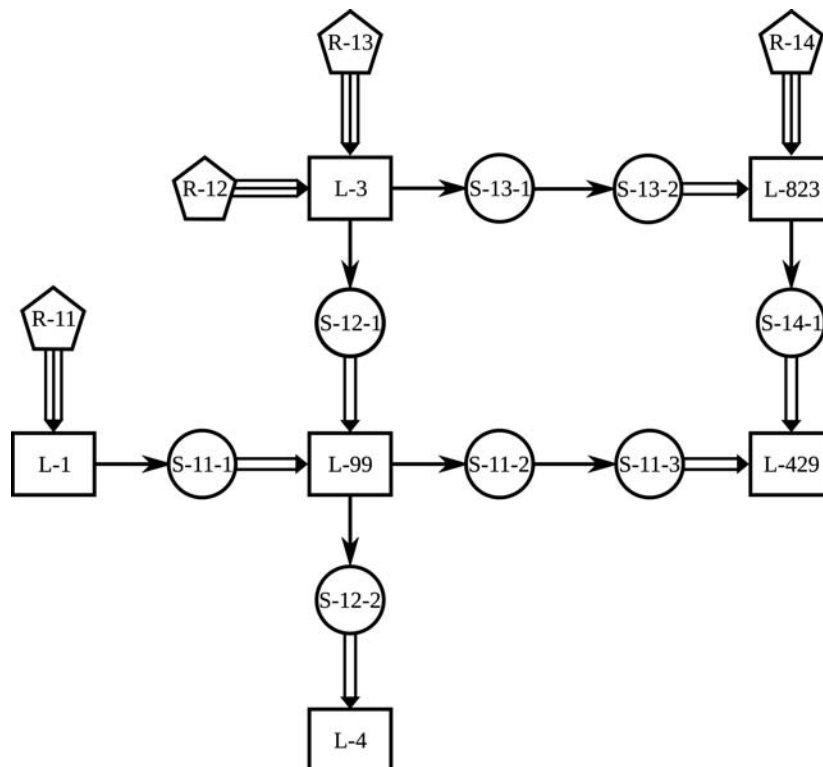


Fig. (11). Deciding the path from L-3 to L-429.

three statement description (S-12-1, S-11-2, and S-11-3), with one possible action inconsistency at L-99. The second candidate path passes through L-823 using routes R-13 and R-14 that results in another three statement description S-13-1, S-13-2, and S-14-1. This second route description also has one possible action inconsistency at L-823. Both paths generate route descriptions that have three route statements and one possible action inconsistency. The difference between the two is *where* the action inconsistency occurs. In the first path, action inconsistency occurs in the middle of a new route description that begins the first statement of R-12 (S-12-1), but at L-99 switches to the second statement of R-11 (S-11-2). Thus, the route change occurs in the middle of R-11 rather than at the end. On the other hand, in the second candidate path, the switch from R-13 to R-14 occurs at L-823, i.e., at the end of R-13 and at the beginning of R-14. Thus, the second path is chosen.

The third heuristic, **H-3**, prefers shorter route descriptions over longer ones, because longer route descriptions are harder to remember. **H-3** is used to build new route descriptions in situations where **H-1** and **H-2** do not apply. Fig. (12) shows how **H-3** is applied. The digraph is built from four routes: R-23, R-24, R-25, and R-26. A new route description is requested from L-172 to L-311. There are two candidate paths. The first one uses R-24 and R-25 to generate a new route description from three statements: S-24-1, S-24-2, S-25-1. The second candidate path consists of R-23 and R-26 with two statements: S-23-1, S-26-1. Both descriptions are created from two routes and have one action inconsistency each. Since **H-1** and **H-2**, **H-3** is applied. The heuristic selects the second path (S-23-1, S-26-1), because it contains fewer route statements: two instead of three.

The three heuristics assign different weights to edges in the digraph. There are four weights that can be assigned to

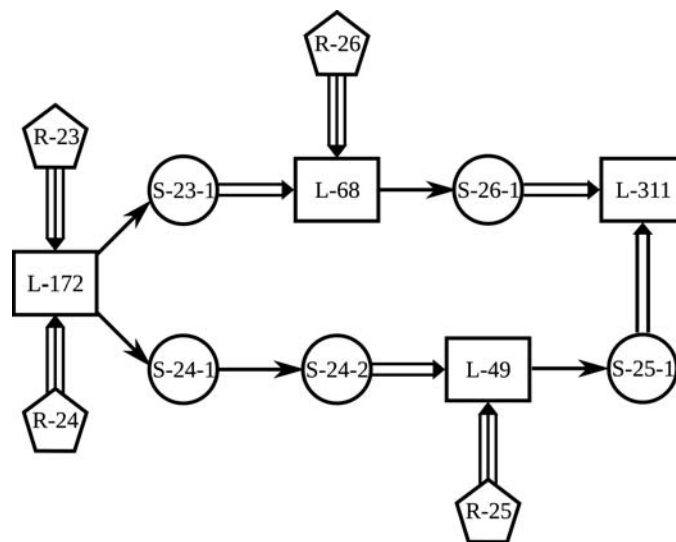


Fig. (12). H-3 Application.

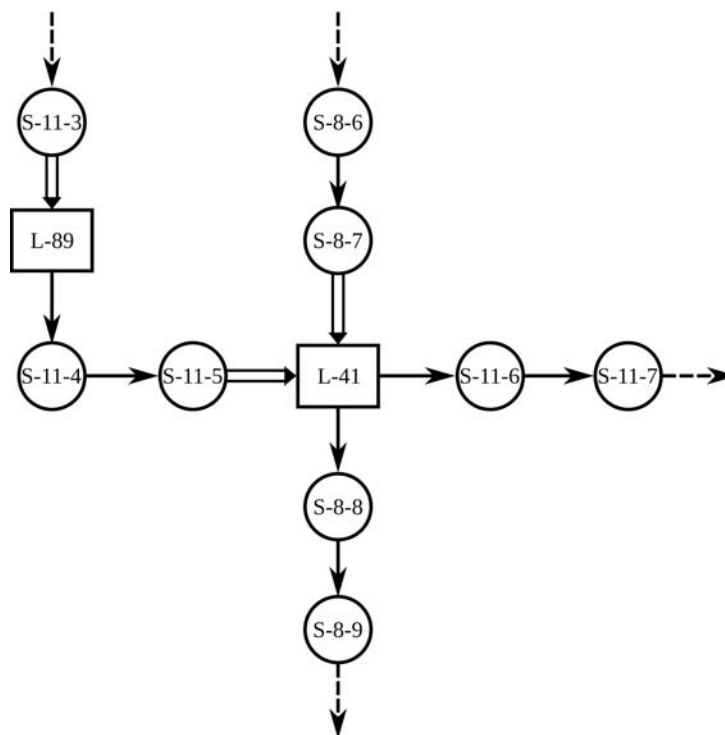


Fig. (13). W-RC assignment.

an edge: TO_LANDMARK_COST (W-TL), TO_STATEMENT_COST (W-TS), START_ROUTE_COST (W-SR), ROUTE_CHANGE_COST (W-RC). Edges are assigned these weights by **CONNECT()** shown in Fig. (7). W-TL is assigned to association edges, i.e., edges from statement nodes to landmark nodes. Landmark nodes and their association edges indicate that two or more routes share a landmark so that the traveler is not affected by the presence of the landmark because the landmark already exists. In our system, the value of W-TL is 0. This value is arbitrary and can be set to another real number so long as it is smaller than the other weights. W-TS represents the cost of following a route statement, and is set to 1. W-SR is assigned to precedence edges that connect a route's starting landmark and to the route's first statement. Since new paths should be built using as few original routes as possible, W-SR can be set to any value greater 1. When a route description is requested, the starting landmark of the requested route may not be a starting landmark of any existing route. Therefore at runtime, the weights of the edges leading from the requested route's starting landmark are set to W-SR. W-RC, used in **H-2**, is different from the other costs in that it is determined by the direction through which a node is reached. The value for W-RC is set to a value greater than W-SR.

Fig. (13) demonstrates a situation that requires W-RC to be calculated. Fig. (14) shows the same graph with the weights assigned by **CONNECT()** in the first pass through the graph in Fig. (13). Fig. (15) shows how the weighted graph in Fig. (14) is modified during the transition cost analysis. The graph in Fig. (15) is a clone of the graph in Fig. (14). Therefore, the landmark L-41 removed from the cloned graph is still retained in the original graph for subsequent access. Specifically, when the search algorithm processes S-

8-7, it needs to determine the cost of the transition from S-8-7 to S-8-8 and S-11-6 via L-41. Since S-8-7 and S-8-8 are in the same route, the transition cost from S-8-7 to S-8-8 is set to W-TS. The transition from S-8-7 to S-11-6 via L-41 creates a potential action inconsistency, because S-8-7 and S-11-6 are from different routes. Therefore, the weight of the edge connecting S-8-7 to S-11-6 is set to W-RC. Similarly, transitioning from S-11-5 to S-11-6 retains the value W-TS since both nodes are in the same route description. The transition from S-11-5 to S-8-8 is set to W-RC, because S-11-5 and S-8-8 are from different routes.

The function **MODIFY_DIGRAPH()** in Fig. (16) modifies the digraph as shown in Figs. (13-15). The modification ensures that there is higher cost for joining route descriptions constructed from different routes, as prescribed by **H-2**. In the modified digraph, *mG*, only two landmark nodes, *startLandmark* and *endLandmark*, are retained. It should be noted that, since *mG* is a clone of *G*, all landmarks are still available for subsequent processing in *G*. The edges of *G* are checked and, if they start or end at *startLandmark* or *endLandmark* (lines 8 and 12), or connect two statements (line 10), then the original edge and its original cost are retained in *mG*. Otherwise, as ensured at line 14, the end node of the edge is a landmark different from *startLandmark* and *endLandmark*. The nodes to which this landmark is connected and their route IDs are checked to see if they are different from the current vertex *v*. If they have the same route IDs, signifying two route statements in the same route description, the new edge uses the original cost of the edge from the landmark node to the second statement node. If two statement nodes come from different route descriptions, the new edge is assigned the weight of W-RC.

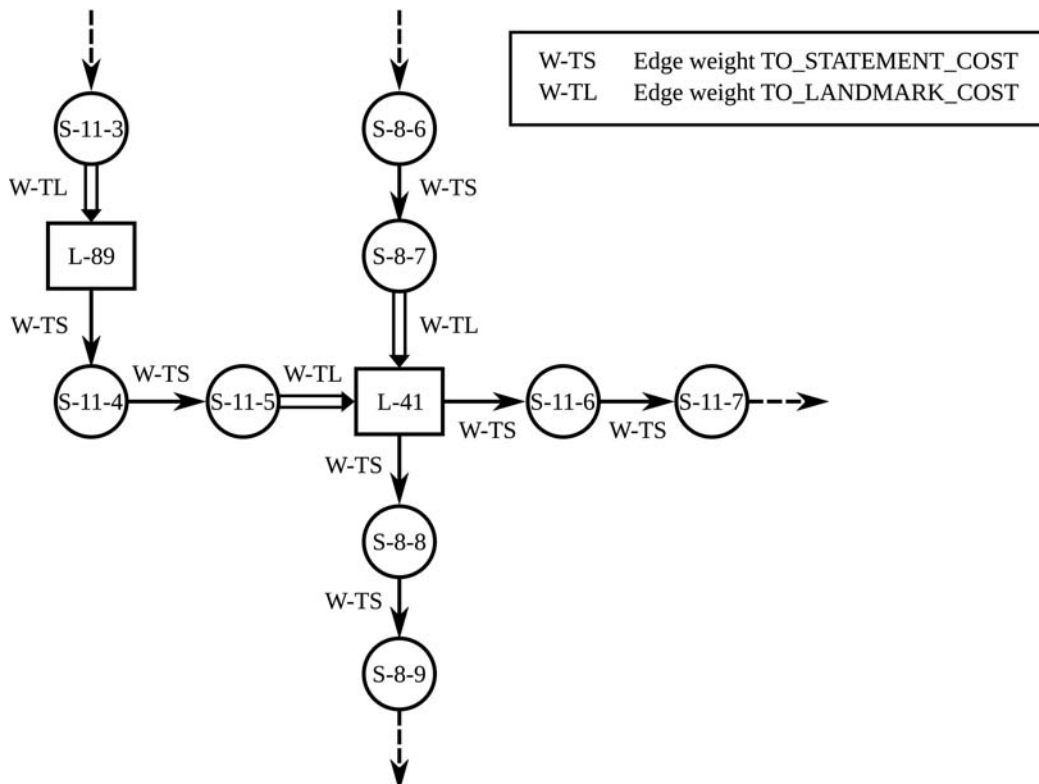


Fig. (14). Default weights set by **CONNECT()**.

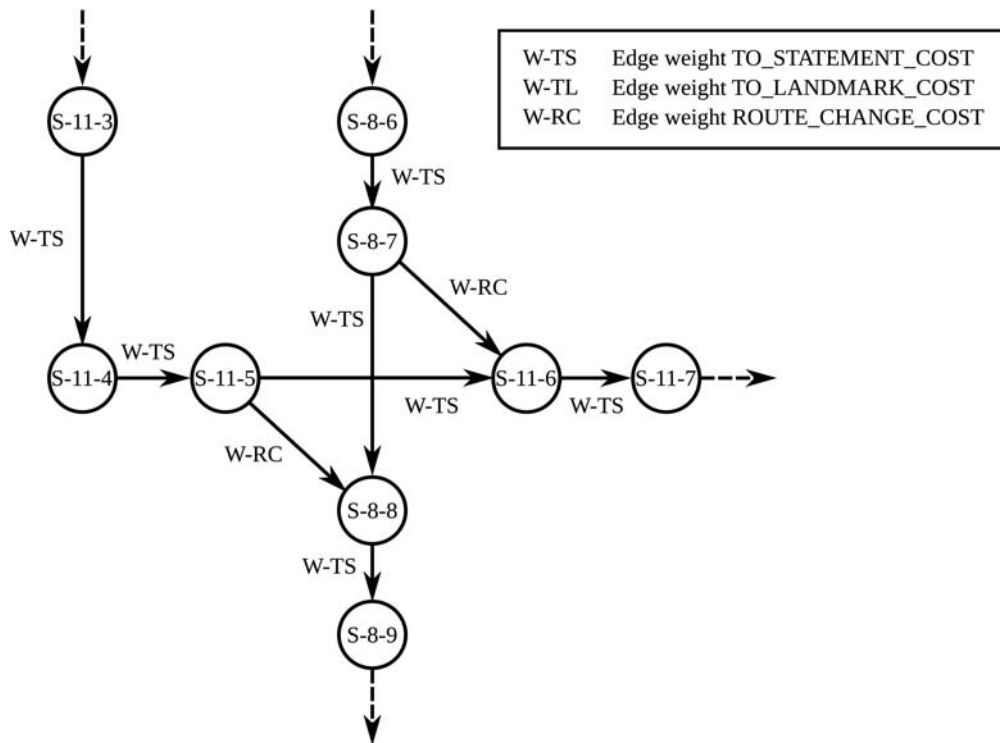


Fig. (15). Graph in Fig. (14) after Transition Cost Analysis.

Function MODIFY_DIGRAPH($G, startLandmark, endLandmark$)

```

1  foreach vertex  $v \in V[G]$  do
2    if  $v$  is a Statement or  $v = startLandmark$  or  $v = endLandmark$ 
3      then
4         $V[mG] \leftarrow V[mG] \cup \{v\}$ ;
5      end
6    end
7  foreach vertex  $v \in V[mG]$  do
8    foreach edge  $e \in E[G]$  starting with  $v$  do
9      if  $v = startLandmark$  or  $v = endLandmark$  then
10        $E[mG] \leftarrow E[mG] \cup \{e\}$ ;
11     else if  $v$  is a Statement and  $e.toNode$  is a Statement then
12        $E[mG] \leftarrow E[mG] \cup \{e\}$ ;
13     else if  $e.toNode = startLandmark$  or  $e.toNode$  is a
14       endLandmark then
15        $E[mG] \leftarrow E[mG] \cup \{e\}$ ;
16     else if  $e.toNode$  is a Landmark then
17       foreach edge  $e2 \in E[G]$  starting with  $e.toNode$  do
18          $startRouteId \leftarrow v.routeId$ ;
19          $endRouteId \leftarrow e2.toNode.routeId$ ;
20         if  $startRouteId = endRouteId$  then
21            $cost \leftarrow e2.cost$ ;
22         else
23            $cost \leftarrow ROUTE\_CHANGE\_COST$ ;
24         end
25          $newEdge \leftarrow PrecedenceEdge(v, e2.toNode, cost)$ ;
26          $E[mG] \leftarrow E[mG] \cup \{newEdge\}$ ;
27       end
28     end
29   end
30 end
31 return  $mG$ ;

```

Fig. (16). MODIFY_GRAPH applies H-2 heuristic to remove landmark nodes and adjust weights.

Function INFER_PATH($G, startLandmark, endLandmark$)

```

1  $mG \leftarrow$  MODIFY_DIGRAPH( $G, startLandmark, endLandmark$ );
2  $\pi \leftarrow$  DIJKSTRA( $mG, startLandmark$ );
3  $description \leftarrow$ 
  BUILD_DESCRIPTION( $\pi, startLandmark, endLandmark$ );
4 return  $description$ ;

```

Fig. (17). Algorithm for inferring new route descriptions.

6.2. Path Inference Algorithm

The path inference algorithm consists of three basic steps. First, the digraph is cloned to handle **H-2** modifications as described in the previous section. Second, the modified digraph is processed to find the shortest path between the start and end landmark nodes. Third, if a path is found, a new route description is generated from it. The algorithm is shown in Fig. (17). The function **INFER_PATH()** takes as inputs the digraph G constructed by **BUILD_GRAPH()** (see Fig. 6) from the set of known route descriptions and two landmarks for a route description. **INFER_PATH()** is called after the original set of route descriptions has been searched and a route between $startLandmark$ and $endLandmark$ has been found. The modified graph constructed in line 1 is passed to the Dijkstra algorithm that finds the least cost path from $startLandmark$ to all nodes in mG . The last step in the process, **BUILD_DESCRIPTION()** in Fig. (18), generates a route description using the path found by Dijkstra's algorithm. Lines 12 and 13 ensure that the first and last route statements are tagged with the appropriate landmarks. The new description is generated from the path from $startLandmark$ and $endLandmark$. The texts of the statements from different routes are not modified. If no path is found, then no description is generated as evident from line 16.

7. EXAMPLES OF INFERRED ROUTE DESCRIPTIONS

This section contains several examples of inferred route descriptions. The original route descriptions on which the inferences were made were taken from our previous study of the self-sufficiency of verbal route directions for blind navigation with prior exposure in which we manually wrote a set of route directions for two routes on the USU campus, machine-converted them into audio files, and placed these files on a Nokia E70 cellular phone [21]. Three completely blind USU students were then recruited to navigate the routes using nothing but the instructions provided on the cell phone. All three participants successfully navigated the routes. Interested readers are referred to [21] for the details of navigation experiments. For the path inference examples in this section, the original routes were tagged with the landmarks from the USU landmark hierarchy by our autotagging module described in Section 3. The inferred route descriptions were built from the lowest cost paths in the digraph.

7.1. Inferred Route Description 1

This example demonstrates the concatenation of two source route descriptions. The first route description covers a route, R-01, from the Animal Science building to the east

Function BUILD_DESCRIPTION($\pi, startLandmark, endLandmark$)

```

1 if  $endLandmark \in \pi$  then
2    $description.startLandmark \leftarrow startLandmark$ ;
3    $description.endLandmark \leftarrow endLandmark$ ;
4    $description.statements \leftarrow [ ]$ ;
5    $current \leftarrow endLandmark$ ;
6   while  $current \neq startLandmark$  do
7     if  $current$  is a Statement then
8       | Add  $current$  to front of  $description.statements$ ;
9     end
10     $current \leftarrow \pi[current]$ ;
11  end
12  Tag first route statement in  $description$  with  $startLandmark$ ;
13  Tag last route statement in  $description$  with  $endLandmark$ ;
14  return  $description$ ;
15 end
16 return NIL;

```

Fig. (18). Function BUILD_DESCRIPTION for building new route descriptions.

entrance to the Old Main on the USU campus.

ROUTE_ID: R-01**START LANDMARK:** Animal Science**END LANDMARK:** Old Main East Entrance**STATEMENTS:**

1. Exit Animal Science at the south entrance. [Tag: Animal Science]
2. Walk forward until you detect the entrance to the quad. [Tag: Quad]
3. Keep walking south until you detect the sidewalk intersection in the center of the quad. [Tag: Center of the Quad]
4. Turn right, walking west.
5. Keep walking straight until you detect bricks on the ground.
6. Walk straight until you detect a wall.
7. Turn left, and look for stairs on your right.
8. Go up the stairs to the entrance. [Tag: Old Main East Entrance]

The second source route description describes a route, R-02, from the entrance to Old Main to the entrance to the President's Office which is located inside the Old Main building.

ROUTE_ID: R-02**START LANDMARK:** Old Main East Entrance**END LANDMARK:** President's Office**STATEMENTS:**

1. Enter the building from the east entrance. [Tag: Old Main East Entrance]
2. Walk forward until you find a second set of doors.
3. Go through them and walk down the long hall.
4. Keep walking straight.
5. You will eventually run in a wall at a t intersection.
6. You should be able to detect some glass doors.
7. These are the doors to the President's Office. [Tag: President's Office]

The path inference algorithm is asked to find a route description for a route from Animal Science to the President's Office. Since R-01 ends at the landmark "Old Main East Entrance" and R-02 begins at the same landmark, the algorithm concatenates these two routes into a new route description for a new route R-03.

ROUTE_ID: R-03**START LANDMARK:** Animal Science**END LANDMARK:** President's Office**STATEMENTS:**

1. Exit Animal Science at the south entrance. [Tag: Animal Science]
2. Walk forward until you detect the entrance to the quad. [Tag: Quad]
3. Keep walking south until you detect the sidewalk intersection in the center of the quad. [Tag: Center of the Quad]
4. Turn right, walking west.
5. Keep walking straight until you detect bricks on the

ground.

6. Walk straight until you detect a wall.
7. Turn left, and look for stairs on your right.
8. Go up the stairs to the entrance. [Tag: Old Main East Entrance]
9. Enter the building from the east entrance. [Tag: Old Main East Entrance]
10. Walk forward until you find a second set of doors.
11. Go through them and walk down the long hall.
12. Keep walking straight.
13. You will eventually run in a wall at a t intersection.
14. You should be able to detect some glass doors.
15. These are the doors to the President's Office. [Tag: President's Office]

This route description does not contain any action inconsistencies, because the phrases "entrance" and "east entrance" in statements 8 and 9, respectively, refer to the same landmark - Old Main East Entrance. The statements where the two routes are joined, statements 8 and 9, form a logical action sequence. Therefore, this description requires little, if any, editing from the community during subsequent collaborative editing.

7.2. Inferred Route Description 2

This example demonstrates how action inconsistencies arise in combining route descriptions. As with the previous example, there are two source route descriptions. The first description describes a route R-04 from Room 405, also known as the CSATL Lab, to Room 414, also known as the Computer Science front office.

ROUTE_ID: R-04**START LANDMARK:** Room 405**END LANDMARK:** Room 414**STATEMENTS:**

1. Exit the CSATL Lab. [Tag: Room 405]
2. Turn left and walk down the hall, going east.
3. If you are trailing the right wall, you will pass an entrance to a computer lab. [Tag: Room 406]
4. After the alcove, you will detect another hall on your right.
5. Keep going straight.
6. You will pass a glass case on the right wall where they put announcements.
7. After the case is a large flat-screen TV hanging on the wall.
8. Immediately after the TV, you will find the door to the Computer Science front office on your right. [Tag: Room 414]

The second source route description describes a route R-05 from the CS front office to the restrooms.

ROUTE_ID: R-05**START LANDMARK:** Room 414**END LANDMARK:** 4th floor restrooms**STATEMENTS:**

1. When you exit the office, you are facing north. [Tag:

Room 414]
 2. Keep walking straight.
 3. Immediately across from the door is the entrance to a small hall.
 4. Walk down this hall.
 5. When you detect a T intersection, turn left.
 6. The men's bathroom is on the left and the women's restroom is on the right in a small alcove. [Tag: 4th floor restrooms]

The path inference algorithm constructs the following description for a route R-06.

ROUTE_ID: R-06
START LANDMARK: Room 405
END LANDMARK: 4th floor restrooms
STATEMENTS:
 1. Exit the CSATL Lab. [Tag: Room 405]
 2. Turn left and walk down the hall, going east.
 3. If you are trailing the right wall, you will pass an entrance to a computer lab. [Tag: Room 406]
 4. After the alcove, you will detect another hall on your right.
 5. Keep going straight.
 6. You will pass a glass case on the right wall where they put announcements.
 7. After the case is a large flat-screen TV hanging on the wall.
 8. Immediately after the TV, you will find the door to the Computer Science front office on your right. [Tag: Room 414]
 9. When you exit the office, you are facing north. [Tag: Room 414]
 10. Keep walking straight.
 11. Immediately across from the door is the entrance to a small hall.
 12. Walk down this hall.
 13. When you detect a T intersection, turn left.
 14. The men's bathroom is on the left and the women's restroom is on the right in a small alcove. [Tag: 4th floor restrooms]

This route contains an action inconsistency at lines 8 and 9 where the two routes were joined. Statement 9 from R-05 describes an initial action of leaving the office. However, in the context of the new route description, entering the front office is completely unnecessary, because it is not a destination but a waypoint where the traveler must make a turn. Therefore, during subsequent collaborative editing statement 9 will likely be changed by the community.

7.3. Inferred Route Description 3

This example shows how the number of routes is reduced when a new route description is generated. All routes in this example come from the first floor of the Taggart Student Center (TSC), a large building on the USU campus housing shops, restaurants, student services, and administrative offices. The first source route description, R-07, describes

the route from the Quick Stop, a small convenience store, to the campus post office.

ROUTE_ID: R-07
START LANDMARK: Quick Stop
END LANDMARK: Post Office
STATEMENTS:
 1. You are standing with your back to the south entrance to the Quick Stop. [Tag: Quick Stop]
 2. Turn left so you are walking east.
 3. On your left you will pass the ATM Machines which make distinctive sounds. [Tag: ATM Machines]
 4. On the left wall you will find a shelf or counter sticking out from the wall.
 5. This is the counter to the Post Office. [Tag: Post Office]

The second route description, R-08, describes how to get from the post office to Cafe Ibis, a small coffee shop inside the TSC.

ROUTE_ID: R-08
START LANDMARK: Post Office
END LANDMARK: Cafe Ibis
STATEMENTS:
 1. Turn to face east and start walking down the hall.
 2. Continue walking east and passing the barber shop, and the copy center as you walk down this long hall.
 3. Towards the eastern end of the building, you will come to a wide open area on your left.
 4. Turn left and walk a little north.
 5. Cafe Ibis is immediately on your left. [Tag: Cafe Ibis]

The third route description, R-09, covers the route from Cafe Ibis to the patio outside the TSC, which has tables and chairs where people can sit and relax.

ROUTE_ID: R-09
START LANDMARK: Cafe Ibis
END LANDMARK: TSC Outdoor Patio
STATEMENTS:
 1. After you get your coffee at Cafe Ibis, turn to face the large hall, that is south. [Tag: Cafe Ibis]
 2. Walk across until you hit the hall's south wall which is all windows.
 3. Turn left and walk east until you find doors on your right.
 4. Exit the door and go through a second door.
 5. Walk forward watching for the big concrete pillars.
 6. This is the TSC Outdoor Patio and there tables where you can sit and relax. [Tag: TSC Outdoor Patio]

At this point, a route description is requested from the Quick Stop (the start landmark of R-07) to the TSC Outdoor Patio (the end landmark of R-09). The result description is a concatenation of the three above routes. The path consists of a description from the Quick Stop to the Post Office and from Cafe Ibis to the TSC Outdoor Patio.

ROUTE ID: R-10**START LANDMARK:** Quick Stop**END LANDMARK:** TSC Outdoor Patio**STATEMENTS:**

1. You are standing with your back to the south entrance to the Quick Stop. [Tag: Quick Stop]
2. Turn left so you are walking east.
3. On your left you will pass the ATM Machines which make distinctive sounds. [Tag: ATM Machines]
4. On the left wall you will find a shelf or counter sticking out from the wall.
5. This is the counter to the Post Office. [Tag: Post Office]
6. Turn to face east and start walking down the hall.
7. Continue walking east and passing the barber shop, and the copy center as you walk down this long hall.
8. Towards the eastern end of the building, you will come to a wide open area on your left.
9. Turn left and walk a little north.
10. Cafe Ibis is immediately on your left. [Tag: Cafe Ibis]
11. After you get your coffee at Cafe Ibis, turn to face the large hall, that is south. [Tag: Cafe Ibis]
12. Walk across until you hit the hall's south wall which is all windows.
13. Turn left and walk east until you find doors on your right.
14. Exit the door and go through a second door.
15. Walk forward watching out for the big concrete pillars.
16. This is the TSC Outdoor Patio and there tables where you can sit and relax. [Tag: TSC Outdoor Patio]

8. DISCUSSION

Our present investigation contributes to the large body of research on verbal route directions for blind and VI travelers. Gaunet [30] proposed several verbal guidance rules for localized wayfinding aids that can be used by blind pedestrians in urban areas. Tuti *et al.* [31] investigated the relationship between lack of visual experience and the ability to create survey-type spatial inferential representations. Their investigation concluded that visual experience may not be necessary for the creation of such representations. Franklin [32] argued that, in the absence of perceptual experience, language is used as a means of constructing and conveying cognitive maps. She proposed several mechanisms for the linguistic production of cognitive maps such as selection of important communication elements, temporal structuring of the elements, reference frame and perspective selection, and verbal regularization of spatial relations. Couclelis [33] proposed a model of the cognitive mechanism underlie verbal direction giving. The fundamental hypothesis was that that both cognitive maps and direction giving discourses are generated by one unifying mental model, which is itself organized by more primitive kinesthetic image schemas and basic-level categories. Dennis [34] presented a cognitive approach to spatial discourse production and distinguishes two types of product categories: action prescriptions and landmark references. Tversky and Lee [35] studied how space structures language and propose the Schematization Similarity Conjecture which states language is successful in

conveying space to the extent that space is schematized similarly in language and cognition. To gain evidence to support their conjecture, they analyzed schematization of routes conveyed in sketch maps or directions. Allen [36] presented an experimental investigation of several principle-based practices in communicating route knowledge. His results showed that following route directions is facilitated by directions presented in correct temporal-spatial order where information is concentrated in statements describing choice points. Kuipers [37] presented the Spatial Semantic Hierarchy (SSH), a formal model of knowledge of large-scale space. The SSH consisted of multiple representations that interact with each other. The SSH can represent states of partial knowledge and enable human and robotic agents to handle navigation uncertainty.

The problem of path inference draws on a large body of research on formal inference models in AI. Many formal inference models rely on a combination of semantic networks and first-order predicate calculus (FOPC), which resulted in the development of semantic network languages such as KRL [38] and KL-ONE [39]. It was subsequently concluded that many such efforts were restricted variants of FOPC. Russell and Norvig [40] give an in-depth historical analysis of these representational efforts.

As the amount of electronic data began to increase rapidly after the advent of the Internet in the early 1990's, many NLP researchers and practitioners discovered that knowledge-intensive approaches do not scale as well as approaches based on statistical analysis, probability theory and shallow knowledge [41]. Shallow knowledge was a basic IE principle. The Fast Reading, Understanding and Memory (FRUMP) [42] system was one of the first IE systems that put to use the power of shallow knowledge. The system used simple rules, called *sketchy scripts*, to extract information from newswire stories, e.g. car accidents. Research and development on IE was supported through a series of Message Understanding Conferences (MUCs) that focused on various domains such as joint ventures and microelectronics and satellite launch reports [45]. The objective of this conference is to automate mundane tasks performed by human analysts reading large quantities of electronic texts. The proliferation of web documents underscored the need for IE solutions that are low cost, flexibility, and easy adaptation to new domains, which, as some researchers have argued (http://en.wikipedia.org/wiki/Information_extraction), the MUC failed to address.

The path inference algorithm presented in this article can also be categorized as a shallow knowledge algorithm. Our findings should be interpreted with caution, because we have not yet tested our path inference algorithm on a large corpus of route directions (due to its unavailability) from a VGI site. Nonetheless, our findings may serve as a sound basis for future research and development efforts in automated path inference. JAPE rules are used to autotag landmarks in free-text route descriptions so that route descriptions can be subsequently transformed into directed graphs. The generated graphs are searched for new paths not originally present in the original route descriptions.

Our approach differs from AI intensive approaches (e.g., [39]) or IE approaches of many MUC systems [41, 45] in that our objective is not to automate inference or knowledge

extraction. Since our long-term goal is to enhance travel independence for VI and blind travelers, travel safety on route is the top concern. Consequently, the results of autotagging and path inference are assumed to undergo Wikipedia-style collaborative human editing. There is a growing body of research that shows that crowdsourcing provides a reasonable alternative to automated annotation tools (e.g., [43, 44]). It is argued that automated annotation tools are expensive, because they require a great deal of skilled labor both in development and maintenance and are still error-prone. It is further argued that crowdsourcing annotation tasks on commercial services such as Amazon Mechanical Turk (<https://www.mturk.com>) can be used to reliably annotate large bodies of electronic data.

Two potential problems with using commercial annotation services for annotating landmarks in route directions for VI and blind travelers are volume traffic and payment. Volume traffic, an important condition for speed and reliability, may not be feasible for route auto-tagging for VI and blind travelers due to their small numbers. Payment may be problematic for the same reason. Volunteered geographic information (VGI) [19, 20] may present at least partial solutions to both problems. Since VGI is a bottom-up volunteer movement organized around specific web sites, local communities may be more interested and motivated to enhance travel accessibility of specific geographic areas for VI and blind travelers. Our path inference algorithm, presented in this article, is designed to be part of a VGI website's back-end software that helps the website's community process and assemble route directions.

ACKNOWLEDGEMENT

Declared none.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflicts of interest.

REFERENCES

- [1] Sendero Group, LLC. Accessible GPS Products [Online]. 2012 June 30, [cited 2012]. Available from: <http://www.senderogroup.com/products/GPS/allgps.htm>
- [2] Kulyukin V. On Overcoming longitudinal and latitudinal drift in GPS-based localization outdoors. In: Anson D, Ed. 28th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA 2005). 2005 June 23-27; Atlanta, Georgia. Arlington, Virginia: RESNA Press 2005.
- [3] Cheng YC, Chawathe Y, LaMarca A, Krumm J. Accuracy characterization for metropolitan-scale Wi-Fi localization. In: Shin KG, Kotz D, Noble BD, Eds. 3rd international conference on Mobile systems, applications, and services (MobiSys '05); 2005 June 6-8; Seattle, Washington. New York, New York: ACM 2005; pp. 233-45.
- [4] Seshadri V, Zaruba GV, Huber M. A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication. Third IEEE International Conference on Pervasive Computing and Communications (PerCom 2005); 2005 March 8-12; Kauai Island, Hawaii. Los Alamitos, California: IEEE Computer Society, 2005; pp. 75-84.
- [5] Crandall WF, Brabyn JA, Bentzen B, Myers L. Remote infrared signage evaluation for transit stations and intersections. *J Rehabil Res Dev* 1999; 36(4): 341-55.
- [6] Roussos G, Kostakos M. RFID in pervasive computing: state-of-the-art and outlook. *Pervasive Mob Comput* 2009; 5(1): 110-31.
- [7] Roussos G. Enabling RFID in retail. *Computer* 2006; 39(3): 25-30.
- [8] Langheinrich M. RFID and Privacy. In: Petkovic M, Jonker W, Eds. Security, privacy and trust in modern data management. Secaucus, NJ, USA: Springer-Verlag 2009; pp. 433-50.
- [9] Nicholson J, Kulyukin V, Coster D. ShopTalk: Independent blind shopping through verbal route directions and barcode scans. *Open Rehabil J* 2009; 2: 11-23.
- [10] McQuiston L. Rehabilitation engineering: ergonomics for one. *Ergon Des* 1993; 1(1): 9-10.
- [11] LaMarca A, Chawathe Y, Consolvo S, et al. Placelab: device positioning using radio beacons in the wild. In: Gellersen HW, Want R, Schmidt A, Eds. Third International Conference on Pervasive Computing (PERVASIVE 2005). 2005 May 8-13; Munich, Germany: Springer 2005; pp. 116-33.
- [12] Golledge RG. Geography and the disabled: A survey with special reference to vision impaired and blind populations. *Trans Inst Br Geogr* 1993; 18(1): 63-85.
- [13] Golledge RG. Human wayfinding and cognitive maps. In: Golledge RG, Eds. Wayfinding behavior: cognitive mapping and other spatial processes. Baltimore, Maryland: John Hopkins University Press 1999; pp. 5-45.
- [14] Golledge RG, Klatzky RL, Loomis JM. Cognitive mapping and wayfinding by adults without vision. In: Portugali J, Ed. The construction of cognitive maps. Dordrecht, Netherlands: Kluwer Academic Publishers 1996; pp. 215-46.
- [15] Nicholson J, Kulyukin V. ShopTalk: independent blind shopping = verbal route directions + barcode scans. Proceedings of 30th Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America (RESNA 2007); 2007 Jun 15-19; Phoenix, AZ. Arlington, Virginia: RESNA Press 2007.
- [16] Nicholson J, Kulyukin V, Marston J. Building route-based maps for the visually impaired from natural language route descriptions. Proceedings of the 24th International Cartographic Conference (ICC 2009); 2009 Nov 15-21; Santiago, Chile: International Cartographic Association 2009.
- [17] Nicholson J, Kulyukin V. CRISS: A collaborative route information sharing system for visually impaired travelers. Cruz-Cunha MM, Oliveira EF, Tavares AJ, Ferreira LG, Eds. Handbook of Research on Social Dimensions of Semantic Technologies and Web Services (Volume II). Hershey, PA, USA: IGI Global 2009; pp. 720-41.
- [18] Passini R, Proulx G. Wayfinding without vision: An experiment with congenitally totally blind people. *Environ Behav* 1998; 20(2): 227-52.
- [19] Goodchild MF. Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. *Int J Spat Data Infrastructures Res* 2007; 2: 24-32.
- [20] Haklay M, Weber P. OpenStreetMap: user-generated street maps. *IEEE Pervasive Comput* 2008; 7(4): 12-8.
- [21] Kutiyawala A, Kulyukin V, Bryce D. On self-sufficiency of verbal route directions for blind navigation with prior exposure. Proceedings of the 25th Annual International Technology and Persons with Disabilities Conference (CSUN 2010); 2010 Mar 22-27; San Diego, CA: California State University, Northridge Center on Disabilities 2010.
- [22] Gaunet F, Briffault X. Exploring the functional specifications of a localized wayfinding verbal aid for blind pedestrians: simple and structured urban areas. *Int J Hum Comput Interact* 2005; 20(3): 267-14.
- [23] Nicholson J. Generation and analysis of verbal route directions for blind navigation. PhD thesis. Logan, UT, USA: Utah State University 2010.
- [24] Cunningham H, Maynard D, Bontcheva K, Tablan V. GATE: A framework and graphical development environment for robust NLP tools and applications. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02); 2002 Jul 7-12; Philadelphia, PA; Stroudsburg, PA, USA: Association for Computational Linguistics 2002; pp. 168-75.
- [25] Cunningham H, Maynard D, Bontcheva K, et al. The GATE User Guide [Internet]. Sheffield, England: University of Sheffield, Department of Computer Science; 2007 [updated 2012 Jul 4; cited 2012 Jun 30]. Available from: <http://gate.ac.uk/sale/tao>
- [26] Finkel JR, Grenager T, Manning C. Incorporating non-local information into information extraction systems by gibbs sampling. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005); 2005 Jun 25-30; Ann

- Arbor, MI; Stroudsburg, PA, USA: Association for Computational Linguistics 2005; pp. 363-70.
- [27] Ratinov L, Roth D. Design challenges and misconceptsc in named entity recognition. Proceedings of the 13th Conference on Computational Natural Language Learning; 2009 Jun 4-5; Boulder, CO; Stroudsburg, PA, USA: Association for Computational Linguistics 2009; pp. 147-55.
- [28] Li Y, Bontcheva K, Cunningham H. SVM-based learning system for information extraction. In: Winkler J, Lawrence N, Niranjan M, Eds. Deterministic and Statistical Methods in Machine Learning. Heidelberg, Berlin: Springer 2002; pp. 319-39.
- [29] Appelt DE, Onyshkevych B. The Common Pattern Specification Language. In: Perose D, Alan M, Eds. Proceedings of a workshop on held at Baltimore, Maryland (TIPSTER '98); 1998 Oct 13-15; Baltimore, MD; Stroudsburg, PA, USA: Association for Computational Linguistics 1998; pp. 23-30.
- [30] Gaunet F. Verbal Guidance Rules for a Localized Wayfinding Aid Intended for Blind Pedestrians in Urban Areas. *Univers Access Inf Soc* 2006; 4(4): 338-53.
- [31] Tinti C, Adenzato M, Tamietto M, Cornoldi C. Visual experience is not necessary for efficient survey spatial cognition: evidence from blindness. *Q J Exp Psychol (Hove)* 2006; 59(7): 1306-28.
- [32] Franklin N. Language as a means of constructing and conveying cognitive maps. In: Portugali J, Ed. *The construction of cognitive maps*. Dordrecht, Netherlands: Kluwer Academic Publishers 1996; pp. 275-95.
- [33] Couclelis H. Verbal directions for wayfinding: space, cognitive, and Language. In: Portugali J, Ed. *The construction of cognitive maps*. Dordrecht, Netherlands: Kluwer Academic Publishers 1996; pp. 133-53.
- [34] Denis M. The Description of Routes: A cognitive approach to the production of spatial discourse. *Curr Psychol Cogn* 1997; 16(4): 409-58.
- [35] Tversky B, Lee PU. How space structures language. In: Freksa C, Habel C, Wender KF, Eds. *Spatial cognition: an interdisciplinary approach to representation and processing of spatial knowledge*. Heidelberg, and Berlin: Springer 1998; pp. 157-75.
- [36] Allen G. Principles and practices for communicating route knowledge. *Appl Cogn Psychol* 2000; 14(4): 333-59.
- [37] Kuipers B. The spatial semantic hierarchy. *Artif Intell* 2000; 119(1-2): 191-233.
- [38] Bobrow DG, Winograd T. An overview of KRL, a knowledge representation. *Cogn Sci* 1977; 1(1): 3-46.
- [39] Brachman RJ, Schmolze JG. An overview of the KL-ONE knowledge representation system. *Cogn Sci* 1985; 9(2): 171-216.
- [40] Russell S, Norvig P. *artificial intelligence: a modern approach*. Saddle River, NJ, USA: Prentice-Hall 1995.
- [41] Cowie J, Lehnert W. Information Extraction. *Commun ACM* 1996; 39(1): 80-91.
- [42] de Jong G. Frump... Frump... Frump... *ACM SIGART Bull* 1997; (61): 54-5.
- [43] Sorokin A, Forsyth, D. Utility data annotation with amazon mechanical turk. In: *Computer vision and pattern recognition workshops, 2008. CVPRW '08. IEEE computer society conference on computer vision and pattern recognition; 2008 Jun 23-28; Anchorage, AL: IEEE 2008; pp. 1-8.*
- [44] Hsueh P, Melville P, Sindhvani V. Data quality from crowdsourcing: a study of annotation selection criteria. In: *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing; 2009 Jun 5; Boulder, CO, Stroudsburg, PA: Association for Computational Linguistics 2009; pp. 27-35.*
- [45] Grishman R, Sundheim B. Message understanding conference - 6: A Brief History. *COLING '96: Proceedings of the 16th conference on Computational linguistics - Volume 1; 1996 Aug 5-9; Copenhagen, Denmark; Stroudsburg, PA: Association for Computational Linguistics 1996; pp. 466-71.*

Received: April 25, 2012

Revised: June 5, 2012

Accepted: June 26, 2012

© Kulyukin and Nicholson; Licensee *Bentham Open*.This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.