Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings

Volume 15 *Seoul, South Korea* Article 49

2015

Generating Geospatial Footprints For Geoparsed Text From Crowdsourced Platial Data

Ahmad O. Aburizaiza
Department of Geography & Geoinformation Science, George Mason University

Matthew T. Rice
Department of Geography & Geoinformation Science, George Mason University

Michael F. Goodchild
Department of Geography, University of California, Santa Barbara

AT1-02

GENERATING GEOSPATIAL FOOTPRINTS FOR GEOPARSED TEXT FROM CROWDSOURCED PLATIAL DATA

Ahmad O. Aburizaiza[1,2], Matthew T. Rice[1], and Michael F. Goodchild[3]

[1]Department of Geography & Geoinformation Science, George Mason University
Suite 2400, Exploratory Hall, Fairfax Campus, 4400 University Dr. Fairfax, VA 22030, USA
Email: aaburiza@gmu.edu
[2]Department of Geomatics, King Abdul Aziz University
P.O. Box 80210, Faculty of Environmental Designs, Jeddah 21589, Saudi Arabia
[3]Department of Geography, University of California, Santa Barbara
1832 Ellison Hall, UC Santa Barbara, Santa Barbara, CA 93106, USA

ABSTRACT

The research paper reports on the generation of geospatial footprints from geoparsed text associated with geocrowdsourced platial data collected and stored in the George Mason University Geocrowdsourcing Testbed (GMU-GcT). The GMU-GcT facilitates study of social dynamics, quality assessment, data contribution patterns, and position validation for geocrowdsourced geo data, with a primary purpose of mapping transient obstacles and navigation hazards in a dynamic urban environment. This paper reports on the automated generation of spatial footprints using open-source software, and discusses the role of automated spatial footprints in quality assessment for automated position validation. A detailed, local gazetteer is used to store placenames and placename variants including abbreviated, slang, former, and jargon-based instances. Obstacle reports containing location descriptions are geoparsed and processed with the help of the GMU-GcT gazetteer to generate geospatial footprints, which are used in a quality assessment process to validate the position of obstacle reports. Continuing research with the GMU-GcT has produced fifteen characteristic footprints types, which are generated and grouped into simple, complex, and ambiguous categories. The open-source tools used for generating these footprints are MapBox, MapBox.js, TURF.js, jQuery, and Bootstrap.

1. INTRODUCTION

George Mason University, is the largest public university in the Commonwealth of Virginia, hosts 33,000 students and several thousands faculty and staff, in a dynamic urban environment outside Washington, D.C. The campus, adjacent to the City of Fairfax, is the site of near-constant construction and expansion. This changing urban environment presents difficulty for students, faculty, and staff who are visually- or mobility-impaired and depend on familiar navigation pathways to get to and from work or home. Construction barricades, sidewalk obstructions, and detours are commonplace. The temporary, unplanned nature of these disruptions make them nearly impossible to capture and map using traditional GIS workflows, and a crowdsourced approach is one of the only practical ways to provide the information in a timely manner. Geocrowdsourcing and volunteered geographic information (VGI), introduced by Goodchild (2007), and reviewed by Elwood (2008), Haklay (2010) and others, represents an opportunity to extend traditional mapping through open-source software and novel geocrowdsourcing workflows. The GMU Geocrowdsourcing Testbed (GMU-GcT), developed by Rice *et al.* (2012, 2013, 2014), introduces a comprehensive geocrowdsourcing approach for collecting and quality-assessing transient obstacle data. Related work by Laakso

(2013), and Karimi *et al.* (2014) provides a useful look at how the accessibility domain benefits from novel open-source mapping applications and data modeling. The GMU-GcT is based on the early work in geoparsing, gazetteers, and geocrowdsourcing (Aburizaiza 2011, Rice at al. 2011, Rice 2012a) and extended to include quality assessment, routing, and visualization (Rice *et al*. 2013a, 2013b, 2014).

Contributors to the GMU-GcT include students, faculty, staff, and members of the public, who submit obstacle reports containing the location, the basic characteristics, and images of the obstacles (Figure 1). This information is processed in a preliminary quality assessment procedure and displayed on a map as a provisional obstacle report. Student moderators field check the reports and provide a comprehensive quality assessment through ground truth. The valid reports are maintained in the system and displayed to the public as confirmed reports.

Figure 1: A GMU-GcT obstacle report with image

2. POSITION VALIDATION AND LOCATION DESCRIPTION TEXT IN THE GMU-GcT

Qin *et al.* (2015) present the quality assessment procedures in the GMU-GcT, including assessments of location, time, and attribute, the three primary facets of the atomic view of geographic information noted by Longley *et al.* (2011). Modeled after the comprehensive

quality assessment of geocrowdsourced data by Girres *et al.* (2010) and Haklay (2010), Qin *et al.* develop assessments of position and categorical attribute agreement, and use this quality assessment procedure to create a composite score for geocrowdsourced data.

Rice *et al.* (2015) discuss the details of the positional validation procedures in the GMU-GcT, by introducing a concept of multi-position validation in the GMU-GcT. Reports contributed by the public are checked for position through three comparisons. First, they are compared to moderated ground truth, where the mapped position of a report contributed by the end-user is compared with the location established through a moderated field check. Second, the images contributed by the end-user are processed to extract embedded geotags and orientation data, which can be combined with geotags and orientation data from multiple reports to create an image geotag-based footprint. Finally, the location description provided by the contributed is processed as a geospatial footprint using extracted placenames, prepositions, distances, directions, landmarks, addresses, and other feature names.

This geospatial footprint developed from geoparsed placenames and other information is a valuable tool for position validation. When the user-contributed map location and the moderator location established through field check do not coincide with the features named in the location description, the inconsistency can be used to flag the report for closer inspection. Earlier work by the authors (Aburizaiza *et al.* 2011, Rice *et al.* 2011, Rice *et al.* 2012a) used this general technique, but used the simplest cases based on proximity to named polygonal features. The work was extended in Rice *et al.* (2015) to include intersection points from two named linear features, a convex hull formed by two named polygonal features, and a linear segment cut by two named linear features. These general cases were developed and implemented so that any crowdsourced contribution to the GMU-GcT with a location description containing named features with this configuration could have a geospatial footprint developed automatically. The ongoing recent work in this paper extends this general technique to include several more cases of general named or platial feature layout, as contained in the obstacle description text in the GMU-GcT.

Location text descriptions in the GMU-GcT contain references to one, two, three, or more places, and these place references can be in various forms including abbreviations, slang, former (old) names, and colloquial variants of standard names. Many of these name variants are contained in the GMU-GcT gazetteer. In additional to placenames or variations of placenames, the GMU-GcT frequently contains locations descriptions with directions, distances, and spatial prepositions. Moreover, a description can contain features that have a distinct group identity and associated name, or a named feature contained within another named feature, or an unnamed places, e.g. walkways, with a geospatial relationship to another named feature.

The combinations of features in the GMU-GcT's location description text is understandably complex. This paper focuses on extending the geoparsing capabilities based on simple and complex placename or platial orientations to generate more cases of geospatial footprints, and instantiating a reference library of the footprints for future research. Several cases are identified and more are being discovered. In the methodology section below, the cases are explained in three categories: simple, complex, and ambiguous. The GMU-GcT gazetteer used in this work is actively updated to reflect naming changes and instances of abbreviation, slang, and colloquial place references. The individual entries for the gazetteer are stored as a JSON array value stored in the GeoJSON properties.

3. METHODOLOGY

A web application was developed to run and test the geoparsing algorithms for the distinctive cases. The application was designed as a mobile web application to permit mobile and tablet users to utilize it through mobile browsers. HTML5, jQuery, and Twitter's Bootstrap were used to build the interface and adjust the site components according to the screen size.

MapBox is a well-known and powerful web mapping technology with a fast map tiling capability. MapBox has a JavaScript Library called MapBox.js which permits programmers to build web mapping applications for geospatial data visualization. It also allows programmers to customize its map object and navigational tools.

MapBox.js uses the TURF.js JavaScript Library, which is a geospatial analysis library that can run on client side, server side, or both. It uses the GeoJSON format as the input and the output. It is a very rich library and runs efficiently since geospatial analysis can run on the client side without connecting to server.

MapBox.js with TURF.js were used to build and visualize the geospatial footprints after extracting placenames, prepositions, and bearing words using jQuery and AJAX. Currently, campus data are stored in a GeoJSON file residing in the server. The plan in the future is to store the campus data in a MongoDB database since MongoDB stores data in JSON format rather than relational database table format.

The geospatial footprints developed are categorized into three categories: simple, complex, and ambiguous. Each category is explained in details. Some cases are similar in concept but are different in their algorithm structure. Such cases are explained in sync while their differences in the algorithm are covered after.

3.1 Simple geospatial footprints

The first simple case (Figure 2) is only one point type or one polygonal place mentioned in a text message with no preposition or bearing. Some landmarks are digitized as either point or polygon features. Other polygonal places are buildings and group of buildings. The point or the polygon is buffered and then both the place itself and its buffer are plotted on the map. The only difference between point and polygon cases is the zoom level set after the geoparsing process is finished.

The system also creates a Bootstrap modal object, similar to a JavaScript's alert, informing the user that a name of one landmark, or building, etc.. was found in the message. Also, the modal informs the user that there were no preposition or bearing words in the message. In other words, the algorithm's detailed steps are explained to the user. This explanatory Bootstrap modal is given in each footprint case. It also notifies the user if no text was entered, or if placenames are not found in the message. Figure 2 displays an example of the first simple case with its explanatory modal.

Figure 2: The first simple case of one polygonal or point type place with no prepositions nor bearing words

The next simple case (Figure 3) is finding only two names of intersected linear places. The algorithm searches the message for terms such as "intersection" or "corner of" as well. Even if two linear places are specified without intersectional terms, the algorithm would find the intersection of the two linear places. The explanation modal will clarify if intersection terms are found or not, in addition to the two linear places names. The intersection point is buffered and then the two linear places, the intersection point, and the intersection's buffer are all plotted on the map. Rarely, two linear places have two intersections instead of one. The algorithm is also taking care of this and the web application will zoom to the center point between the two intersections with an appropriate zoom level. Figure 3 illustrates two examples of this algorithm of one intersection and two intersections.

Figure 3: The left diagram shows one intersection between two roads. The right one demonstrates two intersections because one of the roads has three separate segments

3.3 Complex geospatial footprints

The first complex case (Figure 4) is extracting either one point type place and one polygonal place, or two polygons. A polygon can be either a placename of a single feature or a group of placenames. The vertices of the polygonal places are extracted to an array, using a Turf function named "explode". In the case of a point and a polygon, the point is added to the exploded array of points afterward. A convex hull is created based on the array points and then buffered negatively. The reason of buffering negatively is to avoid areas that are not exactly in between the two places. In the case of one point and one polygon, The buffer is clipped by the polygon since the location would be between the polygon and the point and not inside the polygonal feature itself. As for the case of two polygons, The buffer is clipped twice with the two polygons. After clipping, the result is a multipolygon GeoJSON object. Only one polygon is extracted from the multipolygon which intersects with a linestring connecting the polygon's centroid and the point feature or the two polygons' centroids in case of two polygons. This polygon is referred to as the in-between polygon. It covers the possible location indicated by the user. Figure 4 explains the details of selecting the in-between polygon. The explanatory modal, similarly to all cases, describes the details of the algorithms.

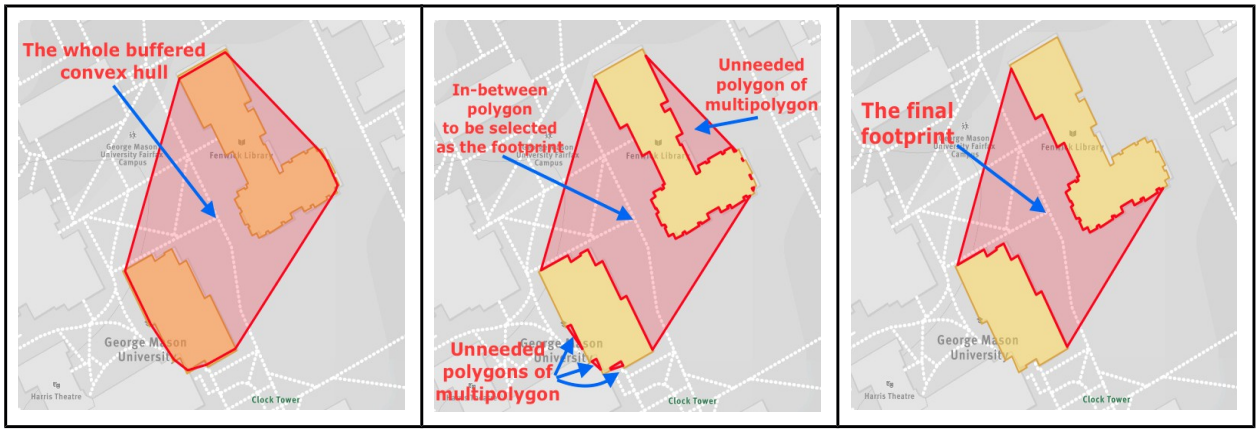

Figure 4: The final transitions in the algorithm of highlighting a location in-between two polygonal places

There are unnamed places such as walkways used to describe a location in relation to named places (Figure 5). The algorithm to select the unnamed places begins similarly to the previous case. Volunteers reported several comments about obstacles on walkways between two polygonal places. The points of the two places are exported to an array. Then a convex hull is created and negatively buffered. The buffer is also clipped by the two polygonal places. The result is a multipolygon and the in between polygon is selected using the linestring connecting the two places' centroids. The unnamed walkways are stored in a separate GeoJSON file on the server. Using jQuery and AJAX, the code iterates the walkways file and uses turf geospatial functions to collect the walkways within the in-between polygon. Figure 5 describes an example of selecting walkways between two places with its explanatory modal. One quick note is that the official building names are David King Hall and Robinson Hall A. The gazetteer has other possible names such as slang placenames, jargon-based placenames, and colloquial or information names. Rob A is a jargon-based name used by students in GMU. King Hall is a common name not the official name.

Figure 5: The footprint of walkways between Robinson Hall A and David King Hall

The fourth complex case (Figure 6) is explaining a location along a linear place with proximity to a polygonal place. The algorithm starts with finding the nearest point on the linear places to the centroid of the nearby polygon. Currently the algorithm will select the previous two segments and the next two segments starting from the nearest point selected before. After that, the points of both the polygonal place and the extracted linear segments are extracted to generate the convex hull. The convex hull is clipped by the polygonal place. The clipped convex hull, the polygonal place, and the extracted segments are all plotted on the map. Figure 6 demonstrates the result of the following volunteer's comment: "Directly in front of GMU commerce building, university Dr, chipped pavement along the sidewalk".

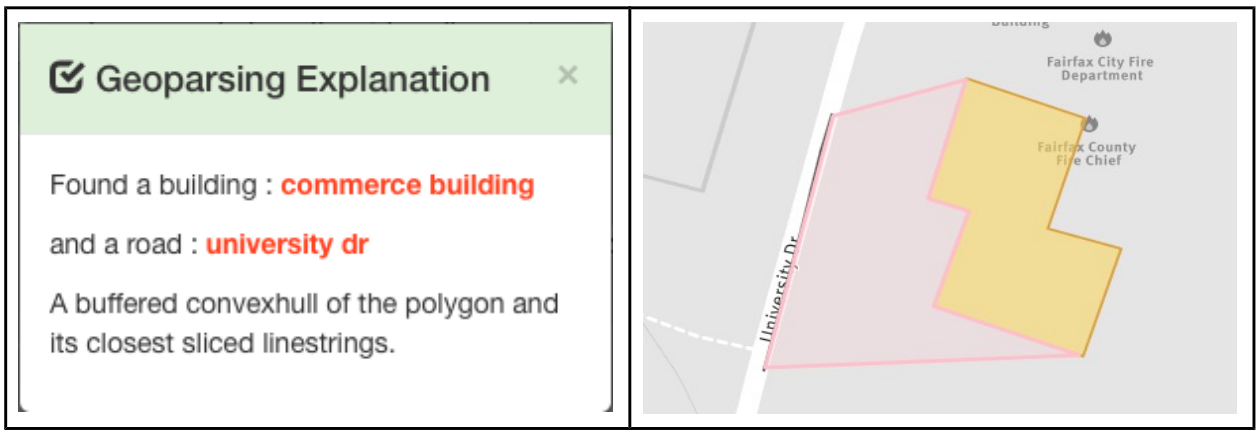

Figure 6: Creating a footprint between a linear place and a polygonal place

Creating a footprint of the case of a bearing word such as north, northwest, west in relation to a polygonal place is challenging (Figure 7). The algorithm begins by creating a rectangular envelope around the polygon. Then the width and length of the envelope are computed to calculate the diagonal of the envelope. Again this is all done through TURF.js geospatial functions. A point is created in the direction of the bearing word with half the diagonal as the distance. The new created point is then buffered, but the buffer could intersect with the polygonal place. If so, the algorithm will clip the polygonal place's part out to give the final geospatial footprint. Figure 7 demonstrates the algorithm's results for three directions: north, northwest, and west, for comparison. The polygonal place part is clipped out in all three cases. A fourth example is also illustrated showing the use of bearing word in relation to a point type place.

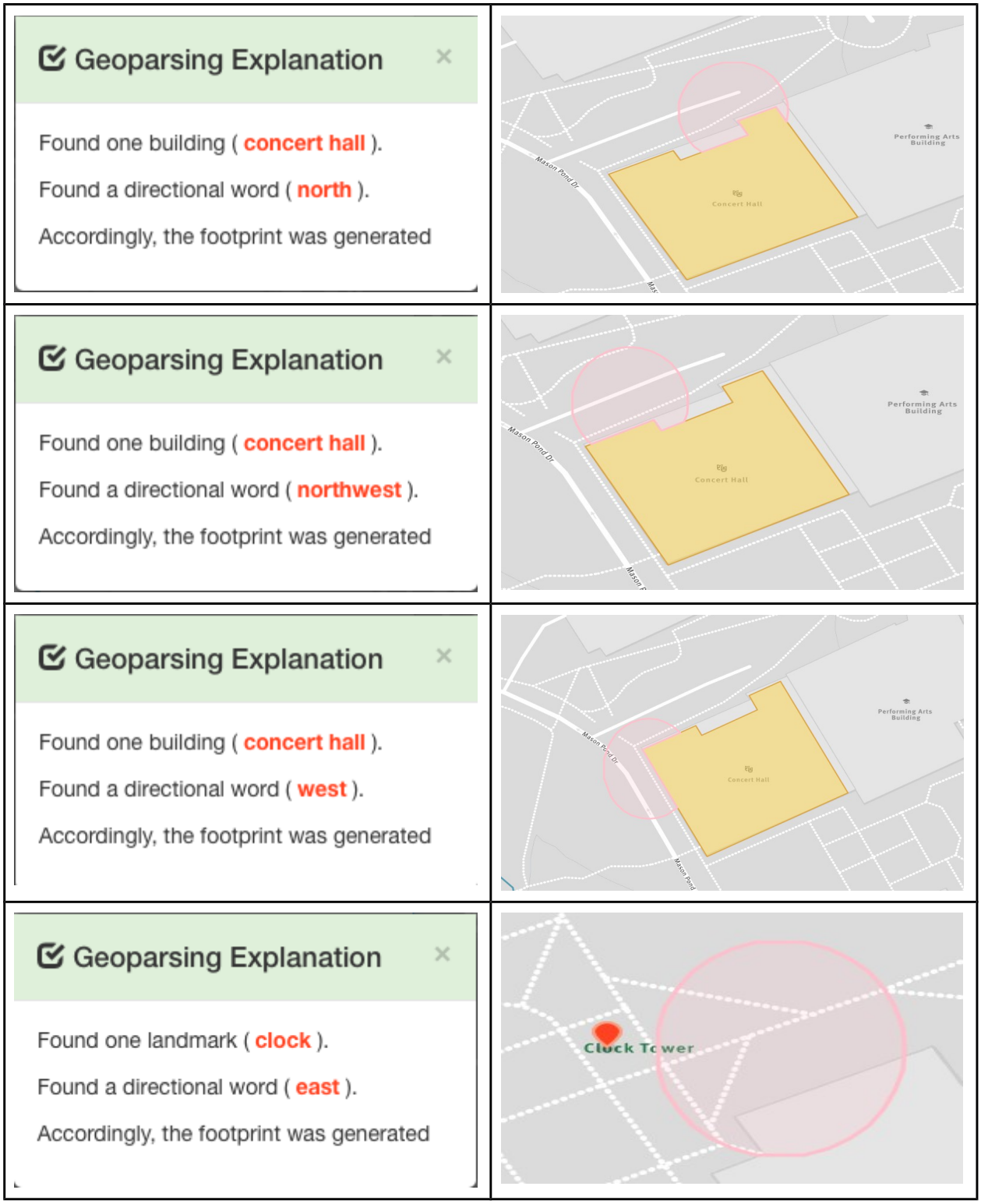

Figure 7: The bearing geospatial footprints in based on polygonal and point type places

The final complex case (Figure 8) is describing a location along a linear place between two intersections with two other linear places. A common example message would be “An X obstacle on Main Street between 1st Street and 2nd Street”. The algorithm first determines which linear place should be highlighted. According the previous example, a volunteer could

say instead "Between 1st Street and 2nd Street, I was driving on Main Street and saw X obstacle". The order of the linear places is different between the two messages. Different examples were tested and the algorithm was capable of determining the correct order. After the two intersections are created, the segments between them are selected and buffered as the footprint. The algorithm is still missing two scenarios, if Main Street is intersecting with 1st Street more than once, and if Main Street has a major curve between 1st Street and 2nd Street. Both cases are being implemented but not working yet. Figure 8 illustrates an example of this case. The original volunteer's comment is "Sager between University and East Street, fractured concrete covered by plywood and orange cones…".

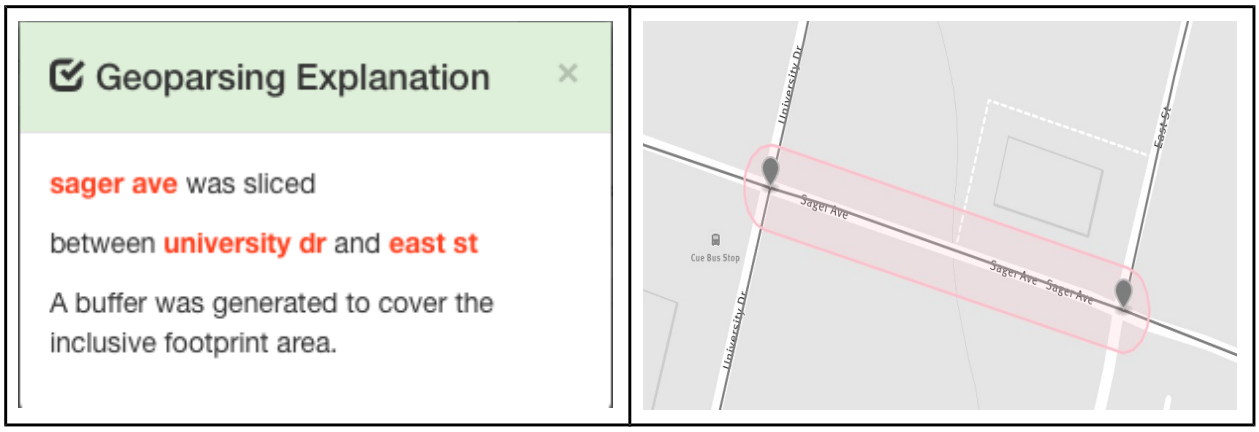

Figure 8: A footprint of a sliced linear place between two linear intersections

3.4 Ambiguous geospatial footprints

There are many examples on the GMU-GcT where the location text and geoparsing code yield ambiguous results. Finding a preposition with one point or one polygonal place is an ambiguous case (Figure 9). Currently the algorithm searches for prepositions in text and then accesses a JSON object that stores buffer distances based on the preposition found. The prepositions are categorized in proximity ranges. For instance, the preposition "next to" has a lower proximity than the preposition "close to". Further research on spatial prepositions and natural language processing is needed in order to determine what processing steps should be undertaken in this case, and specifically, what buffer distances should be used to buffer a place. A surveying of GMU-GcT contributors may be able to help determine what proximity is intended with certain spatial prepositions and some patterns may emerge. Figure 9 emphasizes examples based on different prepositions. Rice *et al*. (2011) note that buffer distances associated with spatial prepositions may be related to factors such as visibility and lighting that are not captured in the GMU-GcT.

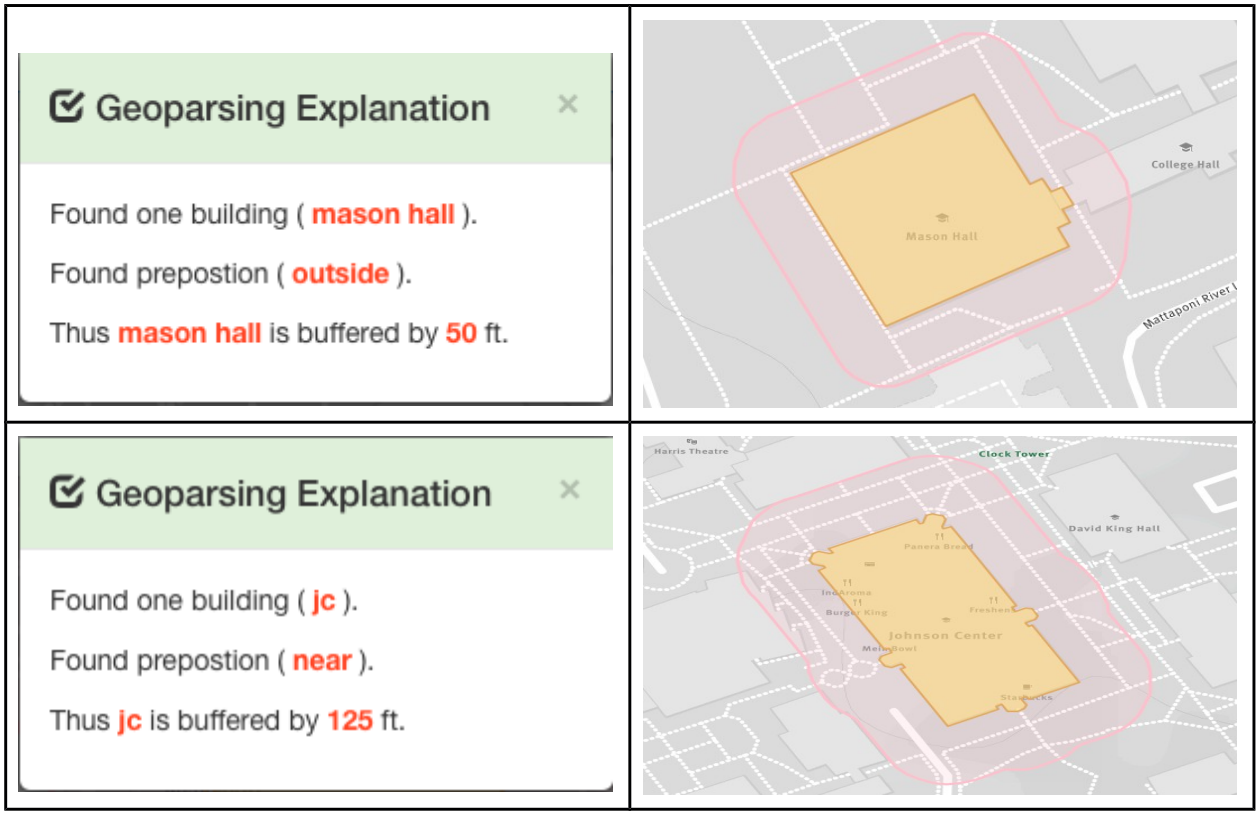

Figure 9: Examples of prepositions found in messages explaining proximity to places

The other ambiguous case is explaining the location by only one linear place(Figure 10). For instance, a volunteer reporting an obstacle on Main St. Main St in Fairfax VA is roughly 3 miles long. The location of the obstacle cannot be determined unless the user gives another placename to clarify the location. The code will inform the user about this ambiguity in a warning modal without generating a geospatial footprint. Figure 10 shows an example of such warning modal. This warning modal may be unnecessary or may be suppressed if additional positional information (such as an image geotag or user-asserted map position) is present. In these instances, the ambiguous geospatial footprint provides only general confirmation that the obstacle is close to or associated in some way with the linear place.

Figure 10: The warning modal if only one linear place is indicated in a message. Patriot Circle is highlighted on the right image

4. DISCUSSION AND CONCLUSIONS

The GMU Geocrowdsourcing Testbed (GMU-GcT) was developed to provide a mechanism to map transient navigation obstacles in real time. A comprehensive quality assessment system has been developed which uses validation of position through multiple sources. A useful way to validate position, and one based on natural human expression, is look at the descriptive location text entered by a contributor. Humans use placenames, prepositions, and directional words to organize and describe the location of objects. The text-based descriptions of obstacle locations in the GMU-GcT provide a way of automatically checking the asserted, map-based positioning of crowdsourced obstacle contributions and the position of obstacles as determined through image geotags, which by nature have a relatively high level of imprecision (Rice 2015). In order to use the text-based location descriptions contributed to the GMU-GcT, gazetteer-based geoparsing and processing have been developed to create geospatial footprints for geocrowdsourced obstacle reports, which are used to validate position. Hundreds of obstacle reports to the GMU-GcT have been processed and analyzed to develop the presented general cases. These general cases allow for automated position validation in a variety of cases. Some cases remain ambiguous and are the source of present and future work, which will involve linguists and other language experts to develop additional general cases.

5. REFERENCES

Aburizaiza, A.O., and M.T. Rice, 2011. "VGI and Geotechnology for Supporting Blind and Vision-Impaired People using a Localized Gazetteer." *FOSS4G: Free and Open Source Software for Geospatial, 2011*. Denver, Colorado.

Elwood, S. (2008). Volunteered geographic information: key questions, concepts and methods to guide emerging research and practice. *GeoJournal*, 72(3-4), 133–135. http://doi.org/10.1007/s10708-008-9187-z

Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal*, 69(4), 211–221.

Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning. B, Planning & Design*, 37(4), 682.

Karimi, H. A., Zhang, L., & Benner, J. G. (2014). Personalized accessibility map (PAM): a novel assisted wayfinding approach for people with disabilities. *Annals of GIS*, 20(2), 99–108. http://doi.org/10.1080/19475683.2014.904438

Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2011). *Geographic Information Systems and Science* (3rd edition). Hoboken, New Jersey: John WIley & Sons.

Qin, H., Rice, R., Fuhrmann, S., Rice, M., Curtin, K., & Ong, E. (2015). Geocrowdsourcing and accessibility for dynamic environments. *GeoJournal*, 1–18. http://doi.org/10.1007/s10708-015-9659-x

Rice, M. T., Aburizaiza, A. O., Jacobson, R. D., Shore, B. M., & Paez, F. I. (2012a). Supporting Accessibility for Blind and Vision-impaired People With a Localized Gazetteer and Open Source Geotechnology. *Transactions in GIS*, 16(2), 177–190. http://doi.org/10.1111/j.1467-9671.2012.01318.x

Rice, M. T., Curtin, K. M., Paez, F. I., Seitz, C. R., & Qin, H. (2013a). Crowdsourcing to Support Navigation for the Disabled: A Report on the Motivations, Design, Creation and Assessment of a Testbed Environment for Accessibility (US Army Corps of Engineers, Engineer Research and Development Center, US Army Topographic Engineering Center Technical Report, Data Level Enterprise Tools Workgroup No. BAA: #AA10-4733, Contract: # W9132V-11-P-0011) (pp. 1–62). Fairfax, VA: George Mason University. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA588474

Rice, M. T., Hammill, W. C., Aburizaiza, A. O., Schwarz, S., & Jacobson, R. D. (2011). Integrating User-contributed Geospatial Data with assistive Geotechnology Using a localized Gazetteer. In A. Ruas (Ed.), *Advances in Cartography and GIScience*. Volume 1 (pp. 279–291). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-19143-5_16

Rice, M. T., Jacobson, R. D., Caldwell, D. R., McDermott, S. D., Paez, F. I., Aburizaiza, A. O., … Qin, H. (2013b). Crowdsourcing techniques for augmenting traditional accessibility maps with transitory obstacle information. *Cartography and Geographic Information Science*, 40(3), 210–219. http://doi.org/10.1080/15230406.2013.799737

Rice, M. T., Paez, F. I., Mulhollen, A. P., Shore, B. M., & Caldwell, D. R. (2012b). Crowdsourced Geospatial Data: A report on the emerging phenomena of crowdsourced and user-generated geospatial data (Annual No. AA10-4733). Fairfax, VA: George Mason University. Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a576607.pdf

Rice, M. T., Paez, F. I., Rice, R. M., Ong, E. W., Qin, H., Seitz, C. R., … Medina, R. M. (2014). Quality Assessment and Accessibility Applications of Crowdsourced Geospatial Data: A report on the development and extension of the George Mason University Geocrowdsourcing Testbed (Annual No. BAA: #AA10-4733, Contract: # W9132V-11-P-0011) (p. 91). Fairfax, VA: George Mason University.

Rice, Rebecca M. (2015).Validating VGI Data Quality in Local Crowdsourced Accessibility Mapping Applications: A George Mason University Case Study (Master of Science Thesis, July 2015). George Mason University, Fairfax, VA.

Rice, R.M., A.O. Aburizaiza, M.T. Rice, and H. Qin. (2015) “Position Validation in Crowdsourced Accessibility Mapping”, *Cartograpica* (in press).