

# Mapping Literature with GeoMapper

---

Undergraduate Honours Thesis

Faculty of Science (Computing Science)

University of Ontario Institute of Technology

Thao Truong

Supervisor: Dr. Christopher Collins

April 19, 2016

## Abstract

Literature often mentions real locations that are used to depict the setting and environment of the content. However, many pieces of literature are comprised of multiple locations and progression of a text may introduce a number of distinct cities throughout the world. A visualization of these locations would allow readers to have a better understanding of the expansive settings. As the use of web-based resources becomes ever frequent, it is the ideal place for a visualization to be accessible to readers. GeoMapper is a web-based literary analysis tool that uses this data to plot locations in order to provide users with a better understanding of the text.

In order to enhance the existing application, additional features were implemented. A number of usability heuristics were considered to improve interactions between the system and the users. The modification of aesthetics was used to separate and link various elements of the interface. To simplify interactions with the system, features such as marker traversing and subset line functionality were added to allow the data to be viewed in a logical order with reduced clutter. In addition, a permalink feature was included to allow users to access previous visualizations. In this thesis, we will also examine geo ambiguity and how we can use the context of a location word, along with other variables to be able to disambiguate them.

## Table of Contents

<b>1. Introduction .....</b>	<b>6</b>
1.1 Problem Statement .....	8
1.2 Goals .....	9
1.3 Related Works.....	9
1.3 Overview .....	12
<b>2. Design Approach .....</b>	<b>12</b>
2.1 Code Refactoring .....	13
2.2 Usability Improvements .....	14
2.2.1 Subset lines .....	14
2.2.2 Aesthetics .....	10
2.3 Interaction .....	16
2.3.1 Marker Traversing .....	19
2.3.2 PermaLink feature .....	20
2.3.2.1 Accessibility .....	20
2.3.2.2 Elimination of location tags .....	21

<b>3. Ambiguity Resolution</b> .....	<b>22</b>
<b>3.1 Ambiguity – How it currently works</b> .....	<b>22</b>
<b>3.2 Resolving ambiguity</b> .....	<b>23</b>
3.3.1 Identifying ambiguities .....	23
3.3.2 Grouping locations .....	25
3.3.3 K-means clustering .....	27
3.3.4 Ranking multiple matches .....	30
3.3.5 Results .....	32
<b>4. Conclusion</b> .....	<b>33</b>
<b>5. Future Work</b> .....	<b>34</b>
<b>6. References</b> .....	<b>37</b>

## List of Figures

<b>Figure 1:</b> GeoMapper project preview.....	6
<b>Figure 2:</b> Preview of Atlas Obscura’s Guide to Literary Road Trips.....	10
<b>Figure 3:</b> LOTR Movie Narrative Chart.....	11
<b>Figure 4:</b> Complex paths .....	14
<b>Figure 5:</b> Subset Functionality .....	15
<b>Figure 6:</b> Colour scheme modification.....	16
<b>Figure 7:</b> Text explorer.....	17
<b>Figure 8:</b> Marker Traversal.....	19
<b>Figure 9:</b> Tooltip with location removal option.....	21
<b>Figure 10:</b> Ambiguity resolution data flow.....	23
<b>Figure 11:</b> Location grouping.....	25
<b>Figure 12:</b> Alternating locations case.....	27
<b>Figure 13:</b> Location outlier case.....	27
<b>Figure 14:</b> K-means randomized centroids.....	28
<b>Figure 15:</b> K-mean first iteration.....	28
<b>Figure 16:</b> Implemented K-mean centroid initialization.....	29
<b>Figure 17:</b> Ranking variables.....	30
<b>Figure 18:</b> Ranking calculation.....	31
<b>Figure 19:</b> Highest population result.....	32
<b>Figure 20:</b> Ambiguity resolution result.....	32

# 1. Introduction

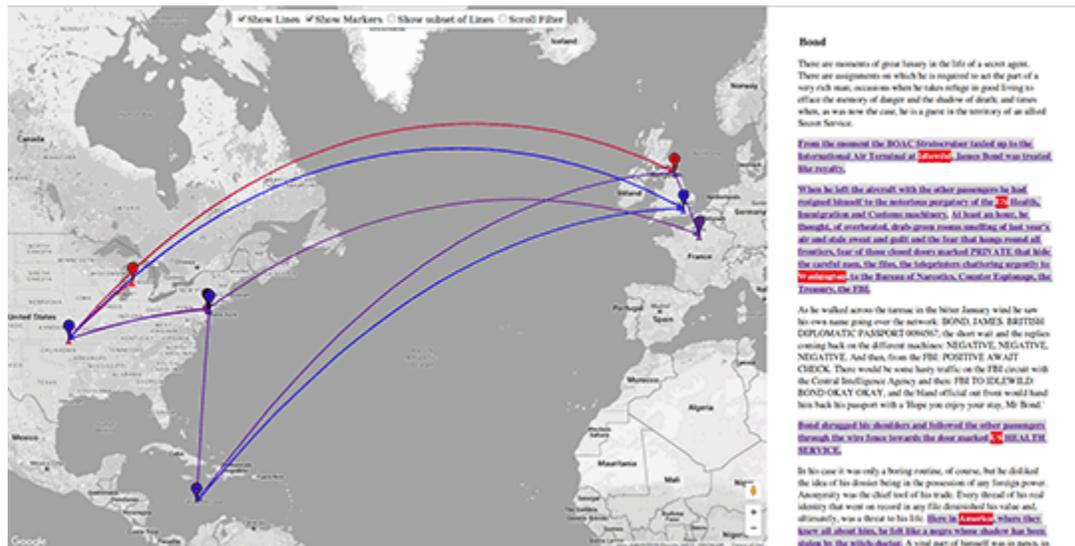


Figure 1: A preview of the GeoMapper project

Many books and pieces of literature have settings that take place in non-fictional locations. An even larger number of stories have setting locations that change over the progression of the plot. Although these expansive geographical locations are meant to help depict a story's theme and environment, their effectiveness could be enhanced. Despite mentioning geographical locations, readers may not be fully knowledgeable geographically. Thus, the mentioned locations may not effectively communicate the scope of the story. Geographic awareness for those who do not have a focus in it, is heavily influenced by travel experience, but since this approach is not exactly feasible, reading literature is the next closest thing. There lies the problem of being unable to fully relate to a story's geographical setting. GeoMapper, (Sidhu) is a web based application, originally created by Ammar Sidhu that provides users with a visualization

of a text's advancement in geographic location. It does this by plotting locations as they are mentioned within a text. This application is designed to provide users with an informative visualization, allowing them to upload a text document that the application uses to generate the visualization. From there, GeoMapper allows the user to view the visualization, which consists of the locations plotted on a map, with a path that is created from the context of the piece of literature. The text is also displayed within a text explorer that links to each location marker. By mapping the developments in a literature's setting, and linking it to the text, a deeper comprehension of what each text encompasses can be achieved.

Based on this prototype, there are numerous possibilities for enhancements to GeoMapper. Since GeoMapper is an application designed to visualize character movement in literature, the accuracy of the location markers is important. A deeper understanding of a text's scope cannot be achieved to its highest degree, if the visualization does not accurately represent the data. One such inaccuracy occurs in the geo ambiguity within each text. Geo ambiguity is the uncertainty of a location name due to the name being able to refer to multiple different locations in the world. London, for instance, could refer to London, England, or London, Ontario. Another example is Sydney, Australia, and Sydney Nova Scotia. The existing application's design could also be modified to provide higher usability.

The objective of this thesis is to improve upon the design of GeoMapper to increase its usability, and the accuracy of the visualization. The product of these changes is shown in Figure 1. The central contribution of this thesis is the development and implementation of an ambiguity resolution into the GeoMapper project. Other contributions include code refactoring, increasing usability with aesthetic improvements and subset lines functionality, and improving user interactions with marker traversing and a permalink feature.

## 1.1 Problem Statement

Resolving geo ambiguities was not a focus point of the original implementation of GeoMapper. As such, cases where geo ambiguity appeared, were resolved with a simple filter of results. This implementation, however, reduces the accuracy of the visualization. By considering alternative approaches, and taking into account accessible data, geo ambiguity can be resolved in a way that preserves its accuracy and context. In addition to the accuracy of the visualization, a visualization application must also have high usability and be easy to navigate. Uploading larger text files sometimes leads to an increasingly cluttered map, and difficult to read text explorer. With an excess of clutter, understanding the data becomes challenging. In order for GeoMapper to be an effective visualization tool, it must be both easy to use, and functionally accurate.

## 1.2 Goals

The goal of this thesis is to supplement the existing GeoMapper application with improvements that will enhance its usability and accuracy as a visualization.

Additional features that can improve the usability and provide a more intuitive interaction with the system include: displaying subset lines, marker traversal, a permalink feature, and customizability. The accuracy of the visualization can be improved by modifying the way that the application disambiguates geo ambiguities. Ideally, these modifications would take into account the data available, and literature context.

## 1.3 Related Works

“Graphs, Maps, Trees” (Franco Moretti), discusses alternative ways to express literature in place of texts. In this book, it examines how literature can be communicated by means of manually created graphs of data, maps of locations, and trees to break down information. The idea of using maps to articulate information that is not easily said with words inspired the GeoMapper project.

With the popularity of select fictional adventure novels, there has been an emergence of interactive map applications. Two novels that have had mapping applications developed for them include Game of Thrones (Martin) and the Lord of the

Rings series (Tolkien). The “Interactive Game of Thrones Map with Spoilers Control” (“Interactive Game of Thrones”) website allows users to view a map of the world and select character paths to observe. This approach however, does not display the context in which these paths are determined from. The “LOTR Project” (Johansson) takes a similar approach, but also allows users to select locations and view brief descriptions of the events that take place. Though the descriptions are not detailed, it still provides a beneficial bit of context for each location. Both maps however, have very limited interactivity and do not provide much functionality in terms of exploring the paths. Another mapping application developed is “A Journey through Middle-Earth” (“A Journey”) which explores locations, and follows characters through their journey. This application has much more interaction, as it follows characters and moves through locations with the characters and shows clips from the movie to give context. Though you can explore the map, it is limited in allowing users to explore the paths themselves, and thus is more of an interactive experience than an analysis.

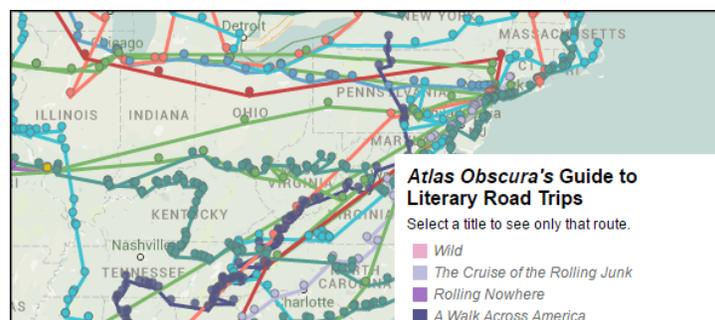


Figure 2: A section of Atlas Obscura’s Guide to Literary Road Trips

A similar approach to this problem is “Atlas Obscura’s Guide to Literary Road Trips” (Melendez) which is a map of the routes in 12 American road trip stories, shown in Figure 2. This approach was done by manually locating and plotting each point in each text on a map. The result differs, since it allows users to view the paths taken in each text on a single map.

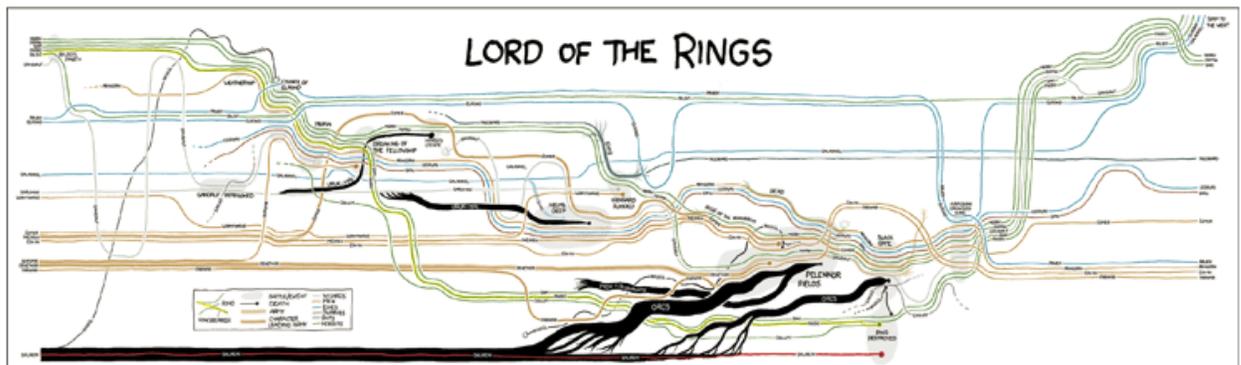


Figure 3: Lord of the Rings chart from the xkcd web comic “Movie Narrative Charts”

Another approach to modelling literature rather than plotting the locations on a map is to graph character interactions as demonstrated in the xkcd web comic “Movie Narrative Charts” (Munroe). Instead of tracking the location of movements in each story, these charts indicate when characters travel together or meet one another, as previewed in the Lord of the Rings chart in Figure 3.

Given how current web technology allows anyone with web access to be able to use a search engine to find the information they are looking for, the results may not be as accurate as we may think. The paper “Extracting Focused Locations for Web Pages”

(Zhang) discusses *GeoRank*, an alternate solution to geo/non-geo ambiguities. *GeoRank* finds solutions using a score system that determines how confident each solution is to be the correct solution. The scores are computed based on relative text distance and inverse geographical distance. The closer each word is to one another in the text, the stronger their relationship. Also, the larger the inverse geographic distance, the higher the score.

## 1.4 Overview

In section two, we will be looking at design changes that have been made to the GeoMapper project to increase its usability. This will cover changes to non-functional attributes, the appearance of GeoMapper, and user interactions. Section three will examine the accuracy of the visualization, and how it can be improved. An overview of how the current implementation handles the situation will be provided, along with possible solutions. Specifically, how the visualizations accuracy can be improved by approaching ambiguity a different way.

## 2. Design Approach

Since the GeoMapper Project is an analysis tool, it is important that the data being communicated is straightforward and easy to interpret. As a project that has a

large number of potential expansions, its code should be clean, and easy to work with. In order to ensure these, a number of changes to the non-functional attributes, the appearance of GeoMapper and its interactions were made.

## 2.1 Code Refactoring

As a project that has such a large network of opportunities for future work, it is important that the foundation is implemented with clean code. While efficiency of code is often the focus of applications, its cleanliness should not be ignored, as it is equally as important. The vast number of opportunities for future works, guarantees that someone else may have to inherit the project at some point. If the code is not readable nor easy to comprehend, the reuse of existing code is much more difficult, requiring that many components be re-implemented. This results in valuable time being lost from trying to understand existing code, or from reworking the solution.

To improve the legibility of the code, each class, and its methods were named in a way that reflects their purpose. For example, the modification of classes and its methods and variables that were named various “tests” to “LocationParser” and “geonameWriter”. This way, locating the required methods is simplified to glancing at the name of each method, rather than determining what each structure intended to do based on its statements. In addition to this, documentation and comments were added

to all structures to give an overview of their functionality. Similar sections of code were isolated to a single method, which reduces redundancy and the chance of error from minor differences. These changes, along with a few others makes the code more easily continued for future developers.

In addition to changes to non-functional attributes, pre-existing errors were fixed. These errors included the file upload not correctly working, location markers not automatically scrolling to the correct anchor in the text explorer, along with a few others.

## 2.2 Usability Improvements

As an analysis tool, the information should be easily interpreted and the interface should allow the user to examine the data clearly.

### 2.2.1 Subset Lines

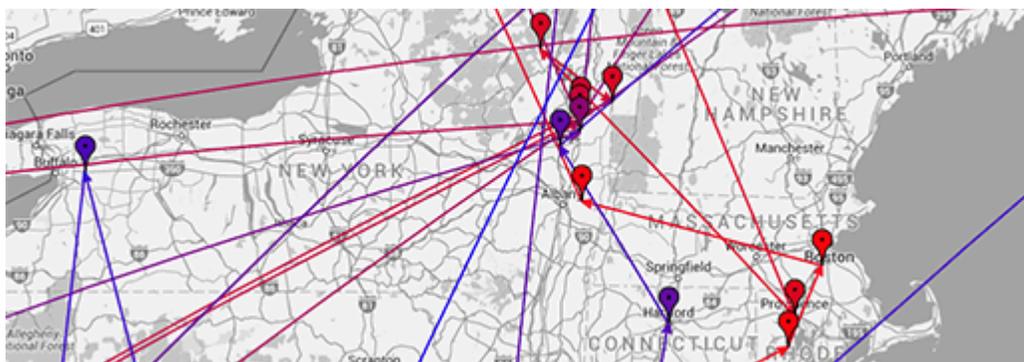


Figure 4: Example of the clutter that results from a text with many locations

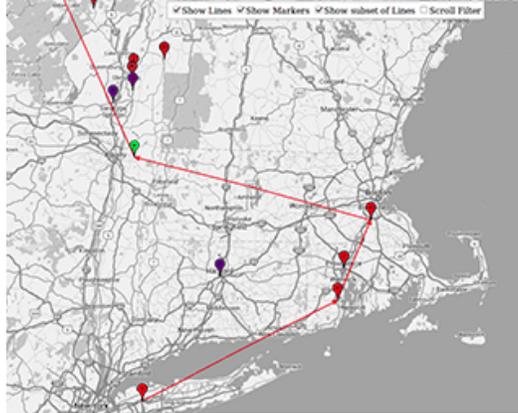


Figure 5: Subset line functionality enabled, which displays the two paths before and after the selected location

At a glance, Figure 4 might be difficult to view, the markers in the visualization are amassed in a condensed area, and the paths overlap repeatedly. The amount of data being displayed appears cluttered and results in GeoMapper being difficult to read. This is especially true when large texts are uploaded. To reduce the strain of trying to identify where each path goes, the show subset line functionality allows users to view the two paths before, and after the selected marker. By doing this, the legibility of the visualization improves significantly, as we can see in Figure 5.

## 2.2.2 Aesthetics

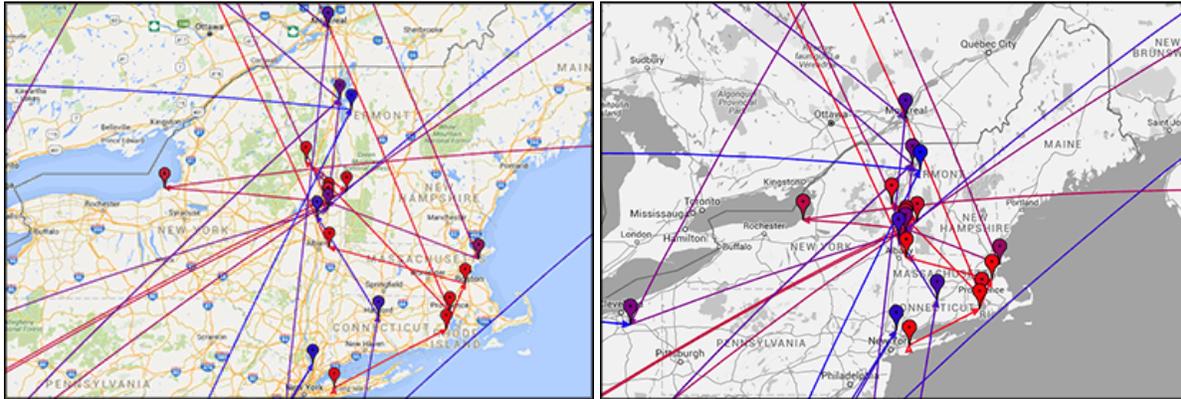


Figure 6: The left image shows the original colour scheme and the right image displays the new changes.

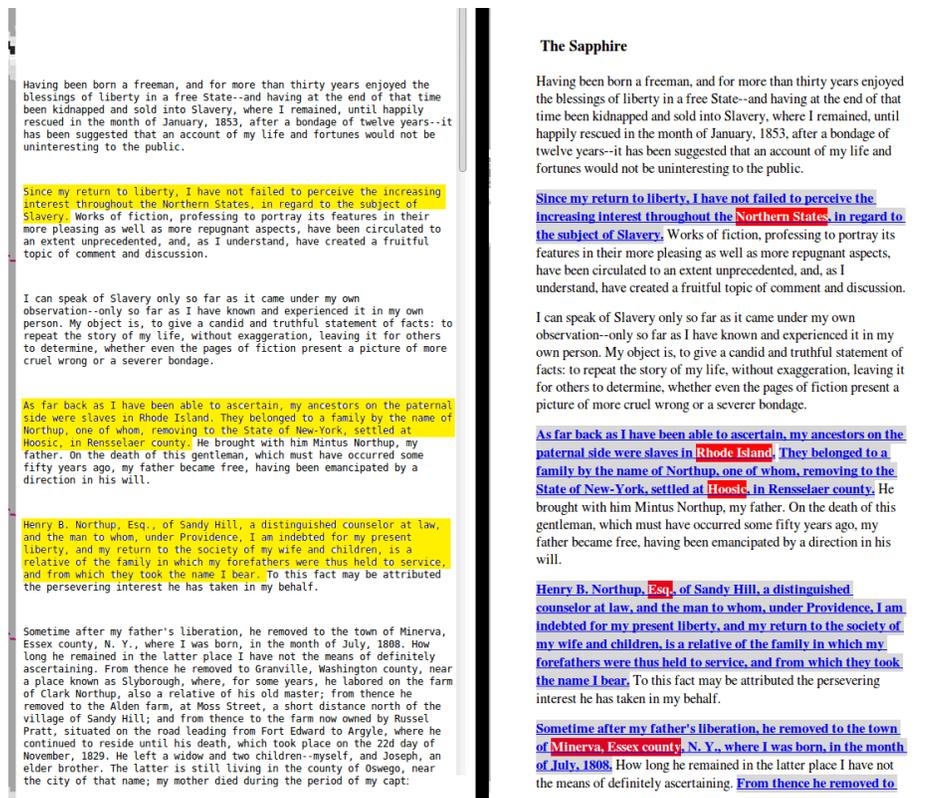


Figure 7: To the left, the original text explorer. To the right, modified explorer.

Prior to implementing any additional features or developing and implementing ambiguity resolution, the user interface and aesthetic design of the existing application was modified. There were many aesthetical changes made to GeoMapper to increase its visual appeal. First, the location markers were altered to generate icons using Google charts Dynamic Icons with colours matching those of its neighbouring path. In addition to this, the location tags identified in the text explorer is also coloured similarly. The benefit of this is making a visible correlation between the markers, paths and its context in the piece of literature. The map theme used by Google Maps was altered to use a modified grayscale, shown in Figure 6. This change results in a much easier to read map. The colours previously were too intense, thus making it difficult to focus on the key features of the visualization with all the clutter. From there, a grayscale map theme was converted into grayscale, but while this helped by reducing the range of visible colours, the map was still difficult to read. The resulting modified grayscale is lightened so that the map, and differentiation between map features is still visible, but its reduce intensity allows the markers and paths to stand out and be features that draws the users attention. Since this visualization is based on the context of a story, it is important that the text explorer be visually appealing. Figure 7 shows the modifications made to the text explorer. Increased whitespace along margins, reduced inter-paragraph spacing to crate spacing that differentiates elements on the page that improves its legibility while also making easier to pay attention to. The change of font

to a serif font that makes reading the text explorer easier to read, and skim through. The change in highlighting colour makes it less alerting and intrusive, while also bringing attention to the location tag within each excerpt. By altering the aesthetics of the application, not only is the visual appeal enhanced, but so is the ability to understand the visualization. It's important that visualizations present data clearly enough that users can take away valuable information. In this project, colour is used to link meaning to the data. For instance, a specific city name is given a colour, and the same colour is used for its marker and within the text explorer. This is better than using a legend to identify each location because would be inefficient and forces users to have to look in multiple places to be able to connect data.

### 2.3 Interaction

Displaying data isn't always enough in a visualization to communicate all the necessary information. This is especially true if it is a publically available web-based application because there are so many technologies available that allow various forms of interaction. As such, if a web application has similarities to any other applications, users may expect to be able to interact with in the same way. In addition to these expectations, if given a visualization, users may want to be able to examine that data more closely. To provide the highest detail of data, GeoMapper should allow intuitive navigation about the visualization.

### 2.3.1 Marker Traversing

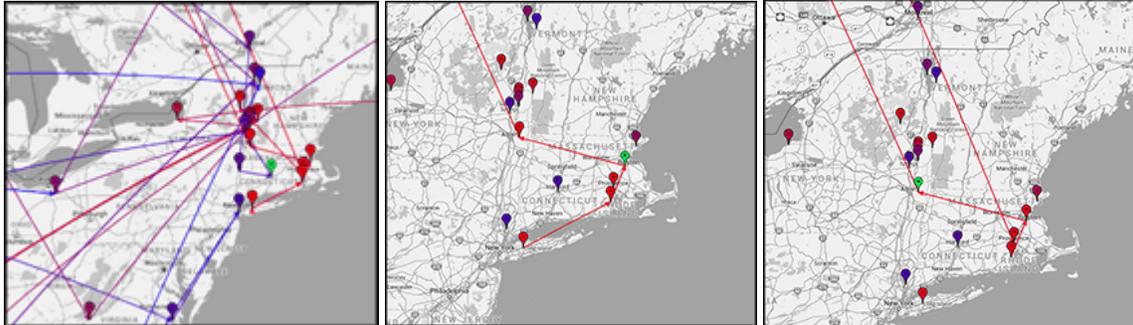


Figure 8: From left to right: Default map display, map with subset functionality enabled, and map with subset functionality enabled after a forward traversal

When given a visualization that involves a specific ordering, one such intuitive navigation interaction would be to be able to analyze the path and ordering depicted. Marker traversing is a functionality that would allow intuitive interaction. In GeoMapper, this functionality traverses through each location marker highlighting the current marker as it is selected. Traversing back and forth between locations makes it possible to trace routes as the literature context defines. In Figure 8, we can see the default visualization with a selected location. The second image has the same location selected, but has subset functionality turned on. Using marker traversal to move forward one location selects the next marker, and adjusts the subset being displayed, resulting in the image to the right.

## 2.3.2 Permalink feature

In the original implementation of GeoMapper, the lifespan of each instance was limited to the windows life. Upon upload, the application generated a series of files needed to create the visualization. In the case that a new file was uploaded, the previous files would be overwritten, and thus lost. Ideally, users should be able to access their previous uploads, thus reducing the strain on the system with having to process new files, especially as the number of users increase and the system needs to scale.

### 2.3.2.1 Accessibility

The new design stores the data to a database upon upload, rather than a file. A feature has been added that allows access to each of the entries using a URL that is uniquely generated by the system on upload. To identify which file is being accessed, there is an additional option to set a name for the piece of literature uploaded. This is a benefit to both the user, and the system. Users are no longer forced to repeat the initial process for the same file multiple times. There is however a trade-off for the system since the load is reduced from the number of time it needs to process, and the amount of memory needed to store each upload.

### 2.3.2.2 Elimination of locations tags

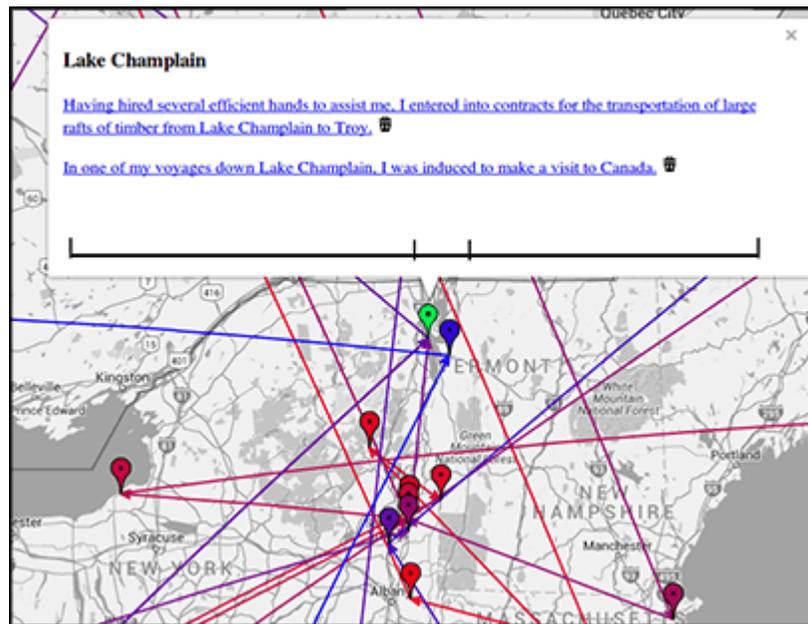


Figure 9: Location marker tooltip with the option to eliminate the individual location tags

While the application generates the visualization, the occasional location mentioned may not actually have been visited by the characters in the story. The user may notice this upon inspection of the analysis, and thus want to remove it. Each of the locations identified by GeoMapper is referred to as a location tag. As a complement to the permalink feature, customizability of each visualization allows users to improve its accuracy manually. While the application processes the data and displays it, there are often flaws that cannot be caught by the system. To reduce the impact of these flaws, the elimination of location tags allows users to be able to change the resulting visualization as they deem appropriate. Figure 9 shows an opened tooltip for a selected marker, along with the icon to remove the location from the analysis. Changes to the visualization are automatically saved to the database mentioned in section 2.3.2.1, thus allowing users to access their updated analysis at a future time.

### 3. Ambiguity Resolution

Accuracy of the location tags is important, as it is the central focus of the visualization. A problem in the accuracy of the visualization relates to ambiguities. One such ambiguity is geo ambiguity: ambiguity amongst locations, where many different locations have the same name.

#### 3.1 Ambiguity – How it currently works

How do we identify the correct location? This wasn't a focus of the original implementation. To solve this problem, initial implementation just selected the location with the highest population. Given that there are multiple options, the largest population is an insufficient choice as it doesn't take into consideration the various variables available. For example. A story that is centrally set in Toronto, Ontario. Say this story mentions that the characters have decided to go on a weekend road trip to London. In this case, the resulting solution would be to select London, England, as it has population of approximately 8 million. This wouldn't be the ideal choice due to the fact that it's on an entirely different continent. While it might still be possible, it isn't the most practical choice. By using this option, it might give a false representation of the text, since the appropriate location may not be geographically close.

### 3.2 Resolving ambiguity

A more ideal solution for geo ambiguity should take into consideration a larger number of variables if they are available. As a visualization that is examining literature, the context in which each location is mentioned could be a valuable detail to consider. An optimal solution for resolving ambiguity would involve examining context, all the possible solutions and balancing each variable in order to derive the most probable solution. The proposed solution to this problem discussed in this thesis involves five steps: identifying ambiguity, grouping locations, k-means clustering, ranking solutions, and storing the results.

#### 3.2.1 Identifying ambiguities

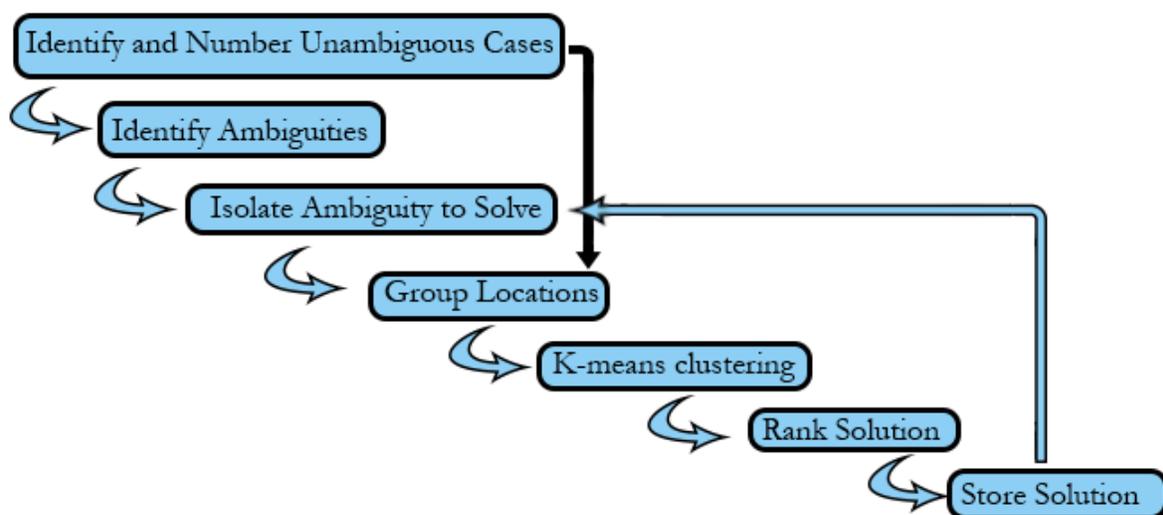


Figure 10: The flow of the ambiguity resolution process solves ambiguities one at a time

A preliminary step to resolving ambiguity is to identify the variables that need to be solved. To do this, the algorithm first queries the database to find all possible matches to locations mentioned in the text. The results of the query are examined, and all unambiguous cases are identified. Each unambiguous result is numbered according to its location order within the text. From there, we can deduce which locations are ambiguous. In this algorithm, each ambiguity identified is resolved one at a time. By doing this, each resolution can contribute to the calculation of the next case, since it can then be considered an unambiguous case. Figure 10 illustrates the steps used in this ambiguity resolution, and how the steps work together. This is especially helpful when a text contains a large number of ambiguities, which would make it difficult if all the surrounding locations are ambiguous. For each ambiguity, a query is used to retrieve all the possible solutions to the problem. To ensure all cases are handled by the end, a failsafe is implemented in this calculation. This failsafe involves sorting the possible matches in descending population order. As a result, if a suitable result cannot be found, the location with the highest population is used to catch outliers.



of these groupings is that it allows a smaller area to be examined in the case the locations are fairly spread apart. Grouping location using the surrounding locations keeps the context within the story in consideration. Mentioned locations usually aren't completely random. In most of the literature examined, context is important, because mentioned locations often form patterns.

By using this grouping method, we have increased the accuracy by reducing the area being examined and keeping the context in consideration. However, there remains a flaw in whether or not all these locations are applicable to our calculation. What if any of these points are also ambiguous? In the case that a few of the surrounding points are ambiguous, the algorithm simply discards them from the group. If this applies to all the points in the group, the algorithm will expand until it is able to find at least one unambiguous case. In the outlier case where all the locations in the text are ambiguous, the algorithm would fall back until the failsafe mentioned in the previous section, the highest population match. These decisions allow the grouping of locations to be done without failing.

### 3.2.3 K-means clustering

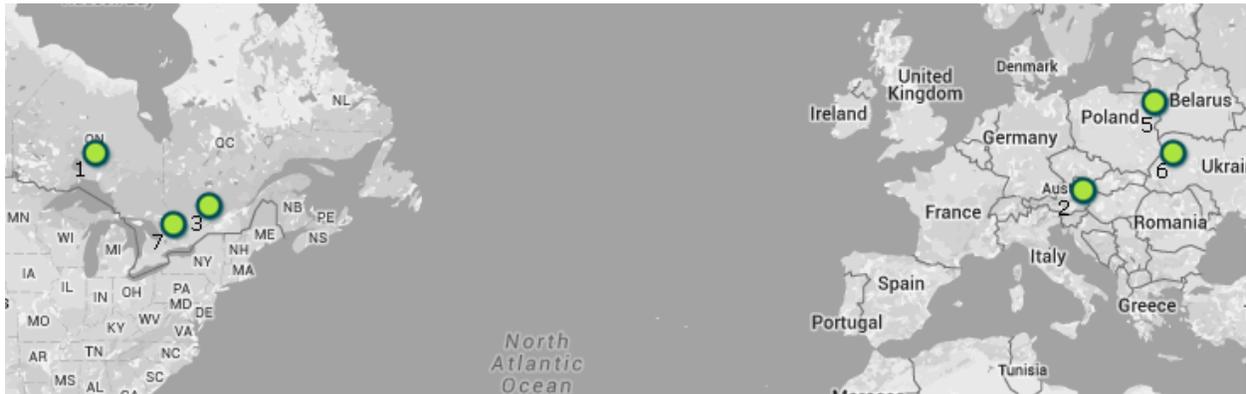


Figure 12: Illustration of a locations in a story that switch between areas



Figure 13: Illustration of location #5, the outlier location

Despite location grouping by context to reduce the span of area being examined, there are still cases where the span can still be relatively large. Many of these cases could be broke up into two categories. The first case, in Figure 12, is where the text would move between areas. Figure 13 depicts the second case, where there would be the outlier case, where there are one or two locations in the group that do not fit in with

all the others. These cases would cause the centroid to be misplaced and radius of area being observed to be much larger than it should actually be. The solution to this was to use a clustering algorithm to separate the locations that weren't in the same region. The clustering algorithm used in this thesis was k-means clustering.



Figure 14: K-means clustering algorithm initializes k=2 random clusters as indicated by the "X"s, C<sub>1</sub> and C<sub>2</sub>



Figure 15: K-means clustering after one iteration

The k-means clustering algorithm (MacQueen, 1967) divides a set of data into  $k$  groups. The clustering algorithm initializes  $k$  random cluster centroids, as seen in Figure 14, and then uses an iterative approach to adjust the centroid. The iterative approach creates the clusters by identifying which centroid each data item is closest to. Once the whole data set has been assigned, the cluster centroids are updated to the center of each cluster, as shown in Figure 15. The iteration is complete when the centroid has no more changes. The result of the k-means algorithm is  $k$ -centroids for  $k$  clusters.



Figure 16: K-means clustering algorithm initializes  $k=2$  clusters using the furthest locations

In this work,  $k$  is set to two, since its being used to separate outliers or check if there are multiple areas being visited. Based on the test cases, two clusters is sufficient in describing each case, factoring in the limited location groups. Rather than generating  $k$  completely random locations, this implementation initializes the  $k$ -centroids to the two furthest data points, as shown in Figure 16. This ensures that the two clusters formed separate the furthest locations. In addition to returning the cluster centroids, the implementation in this work returns the radius of each cluster. The radius of each



$$L_1 = \left(\frac{r_1}{d_1}\right) \left(\frac{r_2}{d_2}\right)$$

Figure 18: Equation for calculating the score of the first location match

Since there are multiple matches at this point, the identified locations must fall within the radius of a cluster. From this point, the final solution is chosen as the match which is closest to a centroid, and within a more condensed cluster. The ranking of each solution takes into consideration,  $r$ , the radius of each cluster, and  $d$ , the distance from each match to the cluster's centroid, as shown in figure 17. The algorithm calculates the ratio of the cluster radius by the distance of the match from the centroid. This value is multiplied by the ratio of the radius of the furthest cluster, by the radius of the cluster it lays within, as shown in Figure 18. The location match that has the highest score is chosen as the solution to the ambiguity. The result of this scoring is that higher priority is given to more condensed clusters, and matches that are closer to the cluster centroid.

### 3.2.5 Results

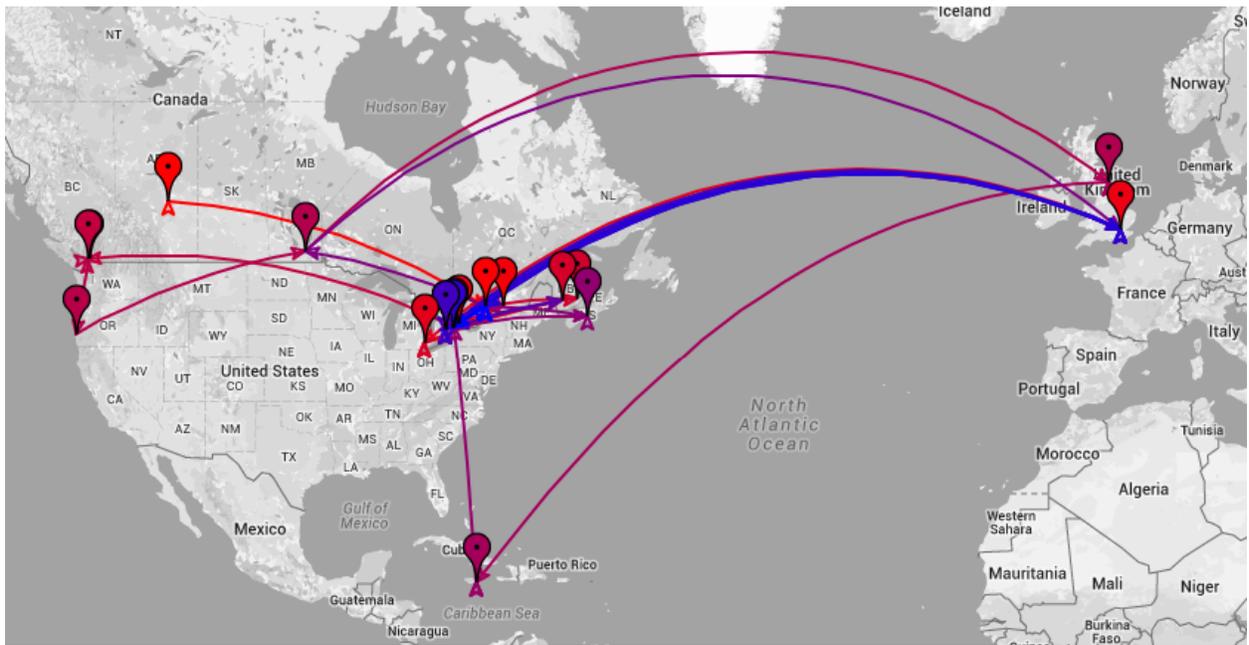


Figure 19: Canadian Travel Itinerary using original implementation of highest population

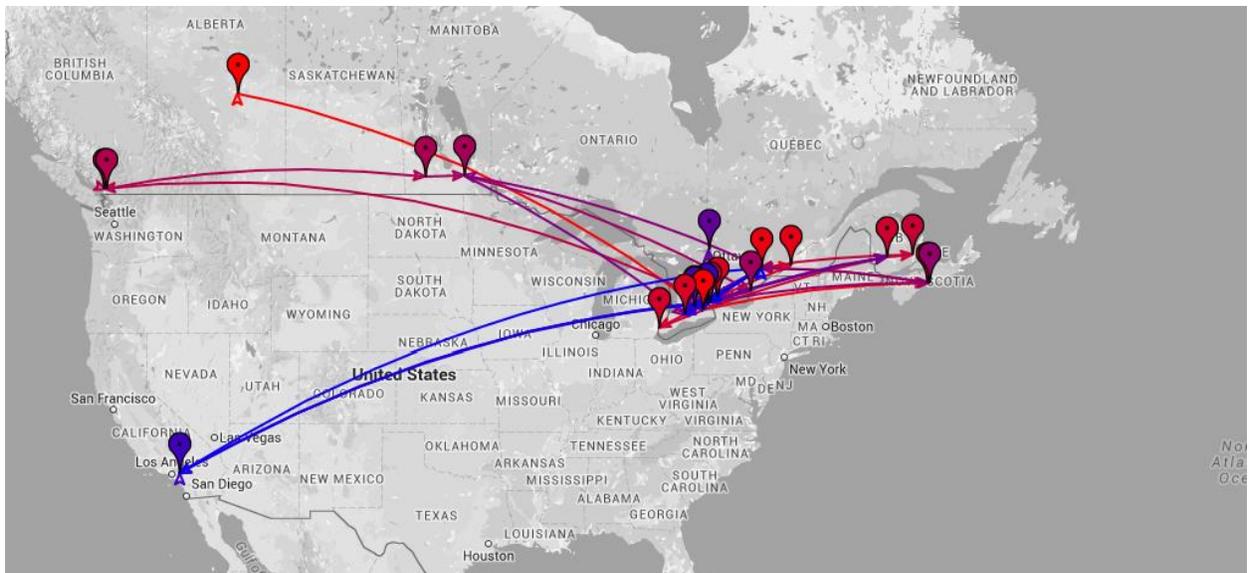


Figure 20: Canadian Travel Itinerary using proposed approach to ambiguity resolution

This approach to solving geo ambiguous cases takes into consideration the locations mentioned around the case within the text. This allows it to narrow down the most likely location to be visited based on locations we are sure of. In Figure 19, a Canadian travel itinerary was mapped using the highest populated city for each ambiguous case. As a result, a few of the cities were misplaced, specifically: Kingston, Washington, and London. Using the new approach to disambiguating each geo ambiguity resulted in Figure 20. For the most part in this case, the locations were identified as within Canada, based on the other identified locations. Since this text was a Canadian travel itinerary, the new solution has increased the analysis' accuracy.

#### 4. Conclusion

The GeoMapper project was originally developed to allow users to generate a map of character movement from their choice of literature. However, with the addition of various features and improvements in accuracy, GeoMapper can be enhanced to be a valuable literary analysis tool. The addition of usability features improves the intuitive interactions, thus enabling users to be able to acquire further data from the visualization. The modification of how ambiguity is resolved improves GeoMappers accuracy in create a visualization, thus giving users a better representation of the data. In addition to these changes, there is a large area available for future works and improvements on the system.

## 5. Future Work

While solving previous questions and problems, the work and research discussed in this thesis opens up a number of possibilities for future work. In its current implementation, GeoMapper is able to solve geo ambiguities using the discussed algorithm. Although it is able to identify a more likely solution to the ambiguity, it could be extended to allow more customizability from the user. This customizability could come in the form of allowing the user to specify a continent or country that could take priority in the ambiguity resolution upon file upload. Upon upload, the system would filter the location matches to reduce the number of matches available prior to the ambiguity resolution. By allowing this decision, the results of the visualization can be fine-tuned to be more accurate by the user. Such a change to the solution would give the user the ability to influence the resolution, while also providing a more accurate fallback to outliers that haven't been solved by the algorithm effectively.

In its current implementation, GeoMapper visualizes the progression of a text by plotting its mentioned locations based on its context. If more work were to be put into GeoMapper, it could incorporate option for other similar visualizations to allow for an even deeper understanding of the piece of literature being examined. Similar visualizations that would take advantage of GeoMapper's map based structure is

visualizing frequency of locations visited, and common routes taken to reveal patterns in literature. Expanding on this concept, GeoMapper could be modified to upload multiple sets of texts that are related, and visualize the multiple data sets simultaneously. Mapping multiple datasets could be used to compare pieces of literature, as well as reveal trends that appear in multiple texts.

There are many additions to GeoMapper that would refine and enhance its functionality and effectiveness as a visualization that can be added in future implementations. The design changes and the ambiguity resolution that were discussed in this thesis are amongst the many possibilities that increase its usability, as well as its functionality. A next step to ensure the effectiveness of GeoMapper would be to conduct user evaluations. Due to time constraints, a usability evaluation was not possible, though it would have been the next step in the software development process. As a visualization, its effectiveness is influenced by the user's subjective opinions on its usability. Due to this, a user evaluation would be highly beneficial to examine how users interact with the system, and determine how they would expect it to work. In a regulated environment, the participants would be given some questions to answer using GeoMapper. Observers would then observe and take notes on the participant's interactions. Notes would include whether existing features were intuitively used, or if they expected specific responses that were not given. These notes would later be used

to determine whether or not existing features are sufficient, or additional functionalities need to be added to be usable.

## References

- "A Journey Through Middle-Earth." *The Hobbit: The Battle of the Five Armies*. Warner Bros, 2014. Web. 19 Apr. 2016.
- "Interactive Game of Thrones Map with Spoiler Control." *Interactive Game of Thrones Map with Spoiler Control*. Web. 19 Apr. 2016.
- Johansson, Emil. "Interactive Map of Middle-Earth - LotrProject." *Interactive Map of Middle-Earth - LotrProject*. Jan. 2012. Web. 19 Apr. 2016.
- Martin, George R. R. *A Game of Thrones*. New York: Bantam, 1996. Print.
- Melendez, Steven. "Atlas Obscura's Guide to Literary Road Trips." *Atlas Obscura*. Atlas Obscura, 20 July 2015. Web. 19 Apr. 2016.
- Munroe, Randall P. "Movie Narrative Charts." *Xkcd*. Web. 19 Apr. 2016.
- Owen, David, and Alison Ryan. *Teaching Geography*. Great Britain: CPI, Bath, 2001. Print.
- MacQueen, James. "Some methods for classification and analysis of multivariate observations." *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. No. 14. 1967.

Sidhu, Ammar. "PathVis: An Online Software for Visualising Locations and Character Movements in Literature." Thesis. University of Ontario Institute of Technology, 2015. Print.

Tolkien, J. R. R. *The Lord of the Rings*. Boston: Houghton Mifflin, 1967. Print.

Zhang, Qingqing, et al. "Extracting focused locations for web pages." *Web-Age Information Management*. Springer Berlin Heidelberg, 2011. 76-89.