Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Human-Computer Interaction

A Visual Analysis Tool for Geolocation Inference Methods

Master's Thesis

Hiyeon Kim
born 2st May 1991 in Daegu

Matriculation Number 118654

First Referee: Prof. Dr. Bernd Fröhlich
Second Referee: Prof. Dr. Benno Stein

Submission date: 24th July 2020

Declaration of Academic Honesty

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, 24th July 2020

HIYEON KIM

Abstract

Location inference methods for social media data are growing in popularity and importance for disaster detection and business research. While inference methods can be evaluated by quantitative metrics such as an error distance or accuracy, visual analysis can provide deeper insights into how an algorithm yields its result. In this thesis, a visual analytics tool is developed to allow researchers to interactively explore their datasets through an iterative process of visualization, comparison, and creation of sets.

A *set*, a collection of tweets with inferred locations, is a key concept on which user interaction is based. Sets can be visualized and compared by being added to visualization widgets in a dashboard layout. New sets can be created by selecting subsets through inter-set operation in a venn diagram or subset selection in a widget. The *Map* and *N-Gram Frequency* serve as the main visualizations along with 4 other complementary widgets. Rose-pie charts in the *Map* widget offer a decluttered representation of error patterns in location inferences by applying aggregation and interactive filtering. The *N-Gram Frequency* widget shows both the term frequency and term frequency-inverse document frequency of n-grams, and can dynamically be updated by adjusting sorting option and filtering.

The expert review shows that rose-pie charts can be more effective in detecting patterns from pairs of locations than the commonly used flow maps, and that meaningful insights on term locality can be gained by the combined use of subset selection and the *N-Gram Frequency* widget.

Table of Contents

Declaration of Academic Honesty II

Abstract III

1 Introduction 1

2 Related Work 5
- 2.1 Geolocation Inference Methods . . . 5
- 2.2 Evaluation Metrics . . . 7
- 2.3 Origin-Destination Visualization . . . 8

3 Structural Design 11
- 3.1 Tweet Inferences . . . 11
- 3.2 Sets . . . 13
- 3.3 UI components . . . 15
- 3.4 Set Actions . . . 18

4 Visualization Widget 22
- 4.1 Map . . . 22
  - 4.1.1 Rose-Pie Chart . . . 22
  - 4.1.2 Details on Demand . . . 27
  - 4.1.3 Additional Layers . . . 34
- 4.2 N-Gram Frequency . . . 38
  - 4.2.1 N-Grams . . . 38
  - 4.2.2 Frequency Measures . . . 39
  - 4.2.3 Percentile Filtering . . . 41
  - 4.2.4 Sorting . . . 43

4.3 Countries 44
4.4 Text Length 46
4.5 Error Distance 48
4.6 Single Metrics 50

5 Evaluation 52

6 Conclusion and Future Work 55

Bibliography 59

1 Introduction

Social media has become ever more prevalent in our daily lives. with the number of users standing at nearly half of the global population [1], social media serves as an inexhaustible source of *big data* for various research areas. Among the different attributes that come with a social media post, location information is particularly valuable for geospatial analysis. Analysis of social data accompanied with geolocations offers deep insight into the geological context of formation, dissemination and distribution of mass opinion and events.

Geospatial analysis on social data has proved valuable in a wide range of fields such as economics, sociology, and emergency management. Using located data on social media, businesses can comprehend characteristics and preferences of potential customers faster and in a much bigger scale than traditional survey-based approaches [2, 3]. Geo-tagged social data can also be exploited to observe political orientation and polarization of users upon major political events such as elections. [4, 5] Real-time analysis of social data has actively been applied by government agencies to enhance situational awareness in emergency management which can ultimately lead to detection and prevention of disasters. [6, 7]

One substantial challenge in tasks that make use of geo-spatial attributes is the acquisition of such data. With increasing privacy concerns, it is a growing trend for users not to attach geolocations to posts or profiles. On Twitter, one of the biggest social media platform, less than 1-2% of tweets are geo-tagged [8]. On top of that, most location information, even when available, tends to lack reliability or precision. On Twitter, for example, users can choose arbitrary place names such as *Earth* or *Inside Your Head*. This is when geolocation inference, the process of inferring geographic locations, becomes a valuable tool.

Due to the challenge of handling such a large scale text corpus that is noisy, informal, and short by nature, most approaches are adopting machine learning techniques, such as Support Vector Machines or Convolutional Neural Networks, to extract geolocations. The tasks of geolocation inference vary in terms of the type of location that is being extracted and the type of information used for inference. There are three types of locations that can be inferred from social media data: (i) a posting location; (ii) a user's home location; and (iii) mentioned locations in a post [9]. To extract locations, various information can be used, e.g. post content, user network, or other metadata. In this thesis, only the tweet-level machine learning based method using text contents, was considered. This is because the dataset was already available from the research team of the Webis Group at Bauhaus-University Weimar, with whom the tool was developed in consultation.

Conventionally, location extraction algorithms are evaluated using a set of metrics. The most commonly adopted evaluation metrics are mean and median error distances between an extracted location and a ground-truth location. Another widely-used metric is accuracy, the ratio of (tolerably) correct inferences within the corpus. Similarly, precision, recall, and their harmonic mean f1-score are also applied to measure the accuracy of inference methods where some inferences fail to yield inferred locations.

While allowing quick and simple comparisons of *how well* algorithms work, these numerical metrics fail to provide deeper insight into *how* they work. In addition to merely knowing if one's method works better than the others, a researcher would like to know the patterns in which the errors occur in a system and how to improve it. The difficulty in doing that stems not only from the vast amount of data but also from the black box nature of machine learning algorithms. Growing complexity of models, e.g. deep neural networks, along with their performances, makes it challenging for researchers to gain constructive insight into how their models arrive at their results.

Interpreting results means getting answers to questions such as follows: what are the ratios between correct and wrong inferences and how are they different among geographic regions; in which direction do errors occur; how likely is a post from one country to be located in another country; how are the error distances distributed; what are the most unique common terms used in a country compared to the others; how do frequently used terms in two countries overlap; how do sets of different ground-truth locations differ in terms of syntactic statistics such as average number of words or type-to-token ratio;

In this thesis, we devised and implemented a visual analytics tool for researchers to better understand the results of their location inference methods. To support the exploration of data, various types of visualizations were included as resizable widgets in a dashboard layout. The core mechanism of the tool is interactive set management. A set comprises one or more instances of an inference—namely, a post with its true and extracted location. By adding these sets to widgets, users can compare how one set is different from the other(s). Most importantly, a new set can be dynamically created by interactive subset selection in a widget, or by leveraging set operations—merging, exclusion, and intersection. This allows users to flexibly explore the data by repeating the cycle of visualizing, comparing and creating sets.

Among six types of visualizations, the *Map* widget serves as the main component. To address the challenge of visualizing a huge amount of inference flows formed by pairs of true and inferred locations, several versions were developed iteratively with different visualization and interaction techniques. The final version of the widget incorporates a rose-pie chart where the pie represents the (tolerably) correctly located inferences and each petal of the rose represents the wrongly located ones in the corresponding direction. To convey the flow of errors, each petal shows the heatmap of their pairs' location, and can also be selected as a subset on click.

As another main visualization, the *N-Gram* widget supports text analysis by describing frequencies for n-grams in two measures (term frequency and term frequency-inverse document frequency) in a scatter plot, overlaid with a bar chart representing the value for the combined sets. Efficient exploration of datasets can be achieved by adjusting percentile filtering value and sorting option. More details on the concept of each widget are described in Chapter 4.

The expert review conducted at the end of the implementation process indicates the usefulness of the application design. Especially, a rose-pie chart with its interactive filtering and aggregation was shown to offer more intuitive understanding when plotting data with pairs of locations than conventional origin-destination visualizations such as flow maps. It was also found that meaningful insights on word locality can be gained by combining subset selection and interactive n-gram analysis into the workflow.

Here is the overview of the following chapters: In Chapter 2, existing works related to the thesis topic is discussed. In Chapter 3, the overall structure of the application is

described. In Chapter 4, detailed explanations on visual concepts on each visualization widget is given. In Chapter 5, the efficacy of the tool is discussed based on the review given by an expert. Finally Chapter 6 concludes the thesis with some suggestions on further developments.

2 Related Work

In the process of designing, understanding of the following research topics was required: differences among various geolocation inference methods, characteristics of conventional evaluation metrics, visualization techniques for origin-destination data as a similar task to depicting inference errors.

2.1 Geolocation Inference Methods

Among various location-based social media platforms, Twitter has been a popular choice of data source for geolocation inference due to its large user base and data availability. In a survey on location inference on Twitter, Zheng et al. [9] classify inference methods based on the type of output locations: home location, tweet location, and mentioned location. Similarly, the shared task on Noisy User-generated Text (W-NUT) proposed by Han et al. [10] was carried out on two levels, tweet- and user-level.

The tweet-level inference refers to inferring the location of a tweet where a tweet was posted. For ground truth locations, geo-tags of tweets are generally used [11, 12, 13]. **The user-level location inference**, or inferring users' home locations, refers to extracting the location where a user primarily resides in. Its ground truth locations are most likely to be gathered from users' profile locations [14, 15, 16]. Since profile locations tend to be noisy and inaccurate at times, some studies extract ground truth locations by aggregating geo-tags within users' tweets [13, 17]. In general, locations inferred on tweet-level have finer granularities, such as GPS coordinates or Point-of-Interests (POIs), compared to coarser administrative regions extracted on user-level inference [9].

Another way of classifying inference tasks is by the type of input information. The methods relying mainly on tweet contents can be classified as **text-based approaches**

[11, 12, 18, 17]. **Network-based approaches**, on the other hand, utilize users' network on Twitter, e.g. followed users, followers, and mentions [14, 15]. This is based on the assumption that a close relationship in Twitter implies closeness in real life. Some other contextual information obtained from meta-data can also be taken into consideration, such as posting times or user profile attributes [13]. Modern researches combine approaches for better results [13, 16, 17].

Network-based approaches are mostly adopted by user-level inference methods. Davis Jr et al. [14] infer home locations through recursive expansion of the network, tracing the following-follower relationships of already-located users. Jurgens et al. [15] applied a similar iterative algorithm, as known as spatial label propagation, which infers home locations using relationships revealed by bidirectional mentions. Along with users' social networks, tweet content is an equally important source of information for network-based approaches. Han et al. [17] identifies location-indicative words in tweet texts for locating users using Inverse City Frequency, a measure proposed in the same study. Ebrahimi et al. [16] proposes a hybrid approach where locality is indicated via the concept of celebrities, namely highly-mentioned users, while using text-based methods as a back-up strategy.

When it comes to tweet-level inference, tweet content is the primary source of information. Similar to Han et al. [17], Priedhorsky et al. [11] and Flatow et al. [12] employ the spatial distributions of n-grams in tweet texts using a Gaussian model for location inference. Along with tweet content, Dredze et al. [13] focus on the impact of time. They extract cities from tweets by training a classifier not only with unigrams and bigrams, but also with posting time and timezone. Huang et al. [18] yield competitive results by applying a deep learning approach to identify differences in language usage from tweets, representing texts using a multi-head self-attention model.

Previous works on location inference methods show that approaches are largely affected by the type of location they aim to infer, the type of ground truth locations and the location granularity of them. Especially, tweet-level inference methods mainly depend on tweet contents, which involves processing and analyzing n-grams, sequences of n adjacent words extracted from texts, for locality detection. In addition, inferring tweet locations is usually done on finer granularities like GPS coordinates. Based on these aspects, the visual analytics application implemented in this thesis adopts the use of GPS coordinates for all locations and n-gram analysis.

2.2 Evaluation Metrics

Mourad et al. [19] classify evaluation metrics into three groups: continuous, discrete, and mixed evaluation. In continuous evaluation, errors are estimated based on distances between geographical coordinates. In contrast, discrete evaluation is employed for resolved locations like country, city, or POI. The metrics used for discrete evaluation include accuracy, precision, recall, and f1-score. Mixed evaluation refers to accuracy within a certain distance, which is a combination of the continuous and discrete metrics.

Error Distance When pairs of GPS coordinates are given, their mean and median error distances can be used for measuring errors. Eisenstein et al. [20] first applied the metrics for evaluation by calculating the error distance in 2D euclidean distance. However, it is recommended to use the great-circle distance instead to properly convey the Earth's surface. While median error distances can be more robust to outliers [21, 22], mean error distances are often preferred since median error distances are insensitive to significantly inaccurate results with non-normal distributions [9, 23]. These metric values need to be interpreted with care in terms of consistency when coordinates are resolved from locations of coarser granularities, as the suggested by Mourad et al. [19].

Accuracy Given pairs of two discrete locations, accuracy is defined as the proportion of all pairs where two locations coincide with each other. When a ranked list of inferred locations is given for one ground truth location, accuracy@k can be used by counting the pairs where the ground truth location is found in top-k results in its list. Accuracy can also yield different results depending on the granularity of locations [19]. For instance, the locations *Weimar, Germany* and *Hamburg, Germany* are deemed identical on the country level, but unidentical on the city level. For flexible comparison, the tool implemented in this thesis offers both options. Accuracy also suffers from bias towards densely populated areas [24].

Accuracy@d Accuracy@d combines two types of evaluation. Here d refers to a threshold for an error distance. An inference is considered correct when the error distance between two locations does not exceed the threshold. Likewise, results are susceptible to the

threshold value [19]. The widely accepted threshold is 161 km, or 100 miles. Then the metric value is calculated in the same way as accuracy. Mourad et al. [19] suggests that accuracy@d, along with Mean and Median Error Distances, is *more consistent and highly correlated across different granularities* compared to accuracy, as they are based on earth representation.

PRF In some cases, inferences can fail to infer locations. These failed inferences have no location values to be deemed correct or incorrect, compared to the successful ones. Precision, recall, and f1-score are metrics used in such cases. Precision is then defined as the ratio of correct inferences to all successful inferences, while recall being the ratio of correct to all inferences, including the failed ones. Given precision and recall, f1-score can be calculated as their harmonic mean. When there are no failed inferences, all three metrics are the same as accuracy.

The metrics described above are all commonly used for evaluating location inference methods. As mentioned, each metric has its limitations and relative advantages. It is recommend by Mourad et al. [19] to combine multiple measures for evaluation. Therefore, all the above types of metrics were considered in the development of the visualization presented in this thesis, allowing users to adjust the parameters and compare metric values among selected sets.

2.3 Origin-Destination Visualization

Origin-destination (OD) data represents the movement of objects from one place to another, such as migration [25] or traffic flows [26]. Tweet data with inferred locations is similar to OD data, in that both types of data contain pairs of locations. Visualizing OD data reveals the patterns among these pairs, which is also the primary goal of the thesis application.

A popular way to visualize OD data is to use **flow maps**. First introduced by Henry Drury Harness [27], flow maps represent the movement on a map with lines encoded with magnitudes and directions. The advantage of using flow maps is that they show geospatial characteristics of the flows. However, flow maps are susceptible to visual clutter due to overlapped lines. In order to reduce clutter, researchers have suggested solutions such as

edge bundling [28, 29], curving lines [30] or smoothing flows [31]. Despite the solutions, clutter still remains a significant issue in flow maps. In addition, bundled edges often make interpretation of flows difficult [32].

Instead of using 2D maps, flows can be visualized in 3-dimensional space. **3D flow maps** generally have less clutter as intersections among lines can be avoided [33]. In particular, information on flows can be more effectively communicated by encoding some attributes as the height of an edge. For instance, Eick [33] encoded the flow magnitude as the height of the line. In flow maps developed by Vrotsou et al. [34], not only the height but also the opacity of each edge conveys the distance between two locations. They also implemented interactive filtering by direction to reduce clutter [34]. The effectiveness of these different 3D encoding techniques were compared in Yang et al. [35] in a Virtual Reality environment. Their results suggest that 3D globes with raised flows allow faster and more accurate understanding in the immersive environment [35].

Alternatively, flows can also be visualized without using geospatial attributes. A simple visualization technique of such type is **OD matrices** [36], where locations are encoded as the row and the column keys in a matrix and a connection between the two as each cell. Zeng et al. [37] adopted the **Sankey flow diagram** to show path-related OD patterns in the field of transportation network. **Chord diagrams** are another good alternative visualization for OD data. For example, Sander et al. [38] arranged origins and destinations in a circular layout and encoded the volume of movement and the direction of each flow as its width and color respectively, in their circular migration plots. They found that the circular layout is easier to read and better reveals the flow patterns than flow maps.

While the non-geospatial visualizations are simpler to read, geographic aspects of OD data are still missing. Hence, some researchers have proposed **hybrid approaches** to combine both spatial and non-spatial representations. Suggested by Wood et al. [39], *OD Maps* preserve spatial layouts of both locations with a two-level spatial treemap [40]; origins are represented in each larger cell, within which its destination density grid map is contained. Stephen and Jenny [41] utilized a circular layout where state nodes are surrounding and connected to the selected state in the middle via force-directed method. While both layouts improve readability in visualizing US county-to-county migration data, its applicability to the world scale is questionable.

Another simple hybrid approach was showcased in *Flowstrates* [40], where two maps separately display each location with the heatmap in the middle showing the flow magnitude over time. Similarly, *MapTrix* [32] connects each location in two maps via an adjacency matrix in the middle, in which each cell denotes a connection. These techniques work well with limited amount of location entries, e.g. countries, but would not scale well with the long list of arbitrary place entries as found in Twitter.

To overcome the inherent clutter problem of OD visualization, **multiple views** can be provided to show different aspects of data or Level of detail (LoD). For visualizing large multivaraite network data, den Elzen and van Wijk [42] developed a prototype user interface which contains two views with different LoD; a map view with finer details and an aggregated graph view with large-scale overview. In their work, especially, the effectiveness of their proposed interaction flow, as known as DOSA (Detail to Overview via Selections and Aggregations), is emphasized. Cuenca et al. [43], in their work *EvoFlows*, present a stream graph view for showing temporal changes in flows alongside a flow map. Dobraja and Kraak [44] explore the concept of dashboards specifically in the context of visualizing OD data. They proposed and implemented an adaptable dashboard as a solution to the lack of flexibility of traditional dashboards; each visualization adapts its level of details (overview, focus, detail) to selections of subsets and elements.

3 Structural Design

The visual analytics tool designed in this thesis was implemented as a web application using a dashboard layout and various types of visualizations for each widget. Before discussing visualizations in each widget, this chapter explains structural concepts employed in the application. The concepts are introduced as follows: the structure and processing of data items; the concept of a *set* as an interactional dataset; an overview of UI components; and the interaction design based on sets.

3.1 Tweet Inferences

The basic data item in the application is *a tweet inference*, namely a tweet with its inferred location. An original tweet is described in JavaScript Object Notation (JSON) format, consisting of key-value pairs as attributes. Some of the core attributes include a unique tweet ID, a posted message, a timestamp, and some information about an author. A tweet may also have additional information such as geospatial metadata or media, which can add up to over 150 attributes [45].

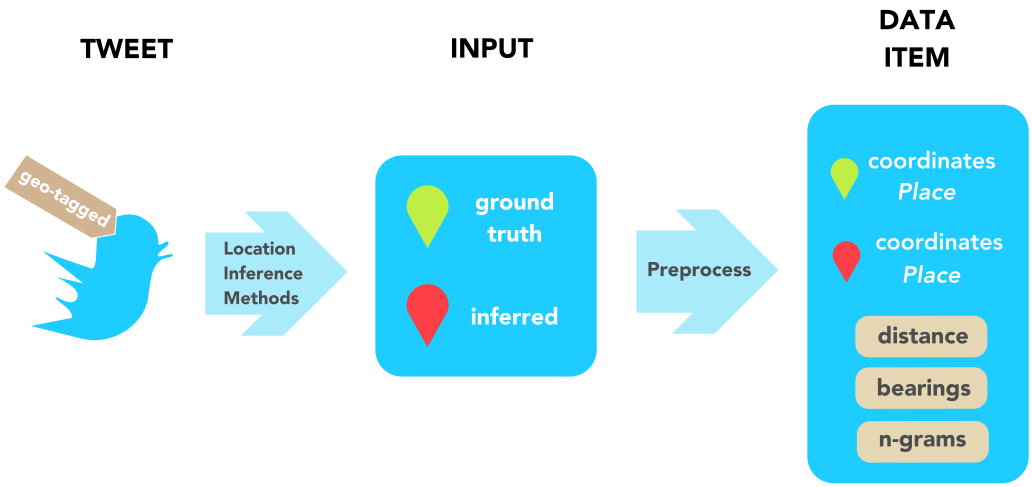


Figure 3.1: Data flow; from an original tweet to a data item.

Tweets are then processed by a location inference method to have inferred locations. Innately, input tweets for location inference methods would be geo-tagged to have ground truth locations. Geo-tagged tweets can have two types of geospatial metadata, an exact location and a *Twitter Place*. Exact locations are shown in Point coordinates of specific longitude-latitude pairs. On the other hand, a *Twitter Place* defines a location as a general area. Unlike exact locations, *Twitter Place* conveys contextual information on a location through additional fields such as a place name (e.g. 'Weimar'), type (e.g. neighborhood, city), a country name, and a country code [8]. Additionally, four pairs of latitude-longitude coordinates are provided as a bounding box. Similar to ground truth locations, inferred locations can also have different granularity (e.g. coordinates, city, state, country, point of interest) depending on the utilized method [9].

Tweets with the pairs of ground truth and inferred locations can then be used as an input to the system. Each widget requires specific attributes for its visualization. To be used properly and efficiently, i.e. to reduce the loading delay of the dataset in each widget, each input tweet is pre-processed to have a coherent set of attributes.

As mentioned above, each tweet can have different granularities for both ground truth and inferred locations. To measure and compare the inference errors consistently, all tweets are made sure to have both types of locations, namely **exact coordinates** and

contextual place information. Exact locations can easily be extracted with geocoding, namely by calculating the centroid of a bounding box which is usually included in the place information. Conversely, place information can be obtained with the help of reverse geocoding API.

Given that all tweets have coordinates of ground truth and inferred locations, some other geospatial attributes are added for each tweet. First of all, a **distance** between a ground truth and inferred location is calculated in kilometers. The map visualization additionally requires bearings and directions . A **bearing** is the angle measured from the north direction in a clockwise direction. Each bearing can then be classified into one of the eight compass **directions** (i.e. N, NE, E, SE, S, SW, W, NW). Bearings and directions always have two values, from a ground truth to an inferred location and vice versa. Aside from these geospatial attributes, **unigrams, bigrams, and trigrams** are extracted from each tweet text for *N-Gram Frequency* widget.

3.2 Sets

In the system, the data is handled through **sets**. A set refers to one or more tweet inferences that are bundled together. In other words, the tweets that belong to a set are the members of the set. Every tweet should be a member to at least one set to be used in widget interactions. In addition to a membership, the core attributes that define a set is a name and a color. While two sets can have the same composition of tweet inferences as their members, their names and colors should be unique from each other.

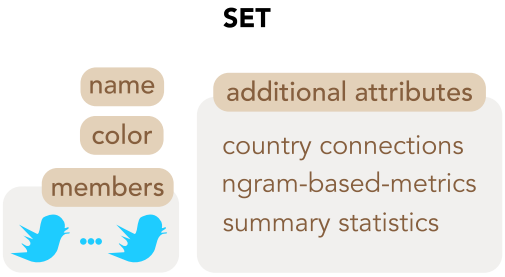


Figure 3.2: Set Attributes

When the application receives its initial input tweet inferences, two sets are automatically created; the *all* set and the set for that input instance. The *all* set represents the set that semantically contains all input inferences (Figure 3.3). The latter gets the name that had been specified along with the input dataset. From an initial input set, users can create new sets by selecting subsets in a widget or by set operations. With more sets created, the *all* set grows bigger to contain all existing sets.

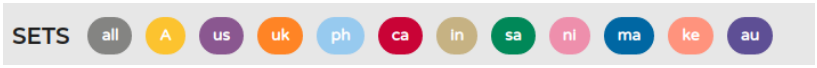


Figure 3.3: Sets. The *all* set is the union of all other sets.

The color scheme for sets is important as colors are what makes sets distinguishable from each other. However, it is tricky to find a large palette of contrasting colors for categorical data. Paul Green-armytage [46] suggested 26 as the provisional limit to the number of distinguishable colors. In this application, the 22 colors of maximum contrast (Figure 3.4) proposed by Kenneth Kelly was used [47]. Although the list of colors dates back to 1965, it is still considered effective compared to the more recent alternatives, as discussed in [46]. The Kelly's list is particularly useful in that it was designed to convey the maximum contrast between colors when selected in order. Since the first two colors are used for the background (white) and widget titles (black), only remaining 20 colors are used to identify sets. In addition, the grey color is occupied for the *all* set.

Colour Serial or selection number	Colour sample matched visually to ISCC-NBS centroid colour	General colour name	ISCC-NBS centroid number	ISCC-NBS colour name (abbreviation)	Munsell renotation of ISCC-NBS Centroid Colour
1		white	263	white	2.5PB 9.5 / 0.2
2		black	267	black	N 0.8 /
3		yellow	82	v.Y	3.3Y 8.0 / 14.3
4		purple	218	s.P	6.5P 4.3 / 9.2
5		orange	48	v.O	4.1YR 6.5 / 15.0
6		light blue	180	v.l.B	2.7PB 7.9 / 6.0
7		red	11	v.R	5.0R 3.9 / 15.4
8		buff	90	gy.Y	4.4Y 7.2 / 3.8
9		grey	265	med.Gy	3.3GY 5.4 / 0.1
10		green	139	v.G	3.2G 4.9 / 11.1
11		purplish pink	247	s.pPk	5.6RP 6.8 / 9.0
12		blue	178	s.B	2.9PB 4.1 / 10.4
13		yellowish pink	26	s.yPk	8.4R 7.0 / 9.5
14		violet	207	s.V	0.2P 3.7 / 10.1
15		orange yellow	66	v.OY	8.6YR 7.3 / 15.2
16		purplish red	255	s.pR	7.3RP 4.4 / 11.4
17		greenish yellow	97	v.gY	9.1Y 8.2 / 12.0
18		reddish brown	40	s.rBr	0.3YR 3.1 / 9.9
19		yellow green	115	v.YG	5.4GY 6.8 / 11.2
20		yellowish brown	75	deep yBr	8.8YR 3.1 / 5.0
21		reddish orange	34	v.rO	9.8R 5.4 / 14.5
22		olive green	126	d.OlG	8.0GY 2.2 / 3.6

Figure 3.4: Kelly's 22 colours of maximum contrast [46]. For coloring sets in the thesis application, the last 20 colors are used.

Just like additional attributes per each tweet, there are attributes that are pre-calculated on a set level for visualizations. There are mainly three types; country connections, n-gram-based metrics, and summary statistics.

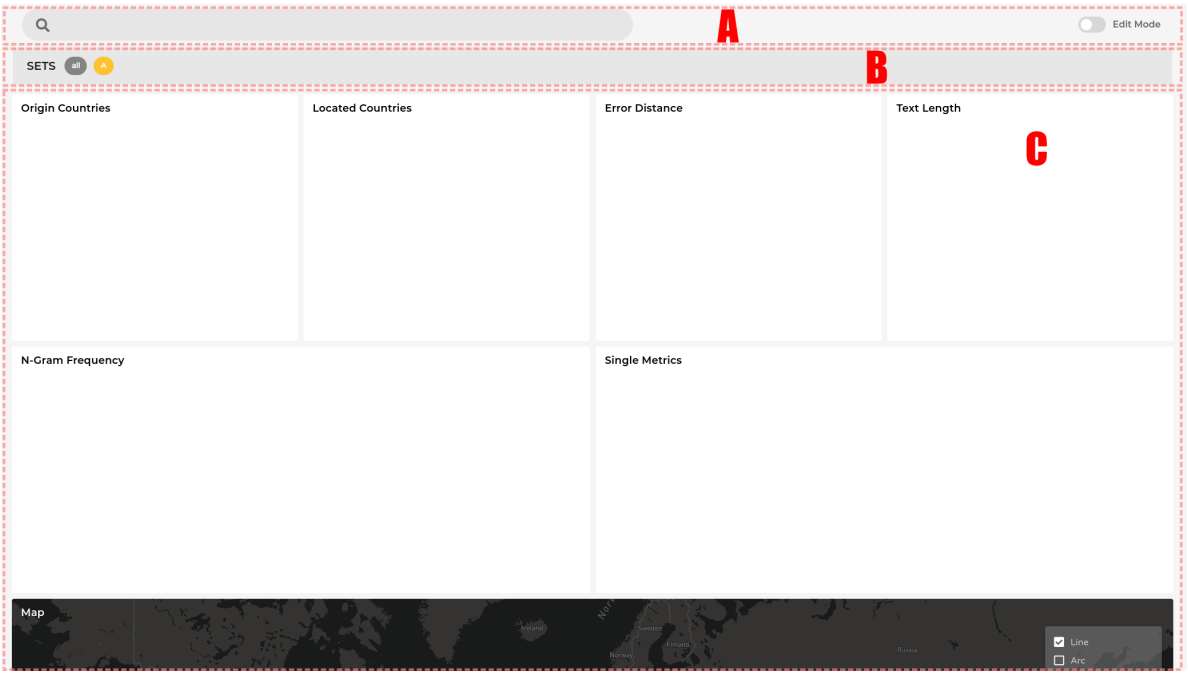
Country connections denote all unique connections between ground truth and inferred countries in a set. Each connection is described by the country of its ground truth location, the country of its inferred location, the tweet inferences of such connections in a set, and the number of inferences. This list of connections is later used in widgets *Origin Countries*, *Located Countries* and *Map*.

For *N-Gram Frequency* widget, the following **n-gram-based metrics** are computed for each n-gram (i.e. unigram, bigram, trigram) list: term frequencies, normalized term frequencies, total counts of term instances, quantiles and quantile distributions. Term frequencies are a list of terms paired with their number of occurrences in a given set. For each term, a normalized term frequency is obtained by dividing each frequency by the total number of terms in a set. Quantiles are values that cut the list of normalized term frequencies in a given proportion. Then a Quantile Distribution is simply the binned result of applying each quantile. The proportions range from 0.01 to 0.99, yielding the list of 99 quantiles and the distribution of 100 bins.

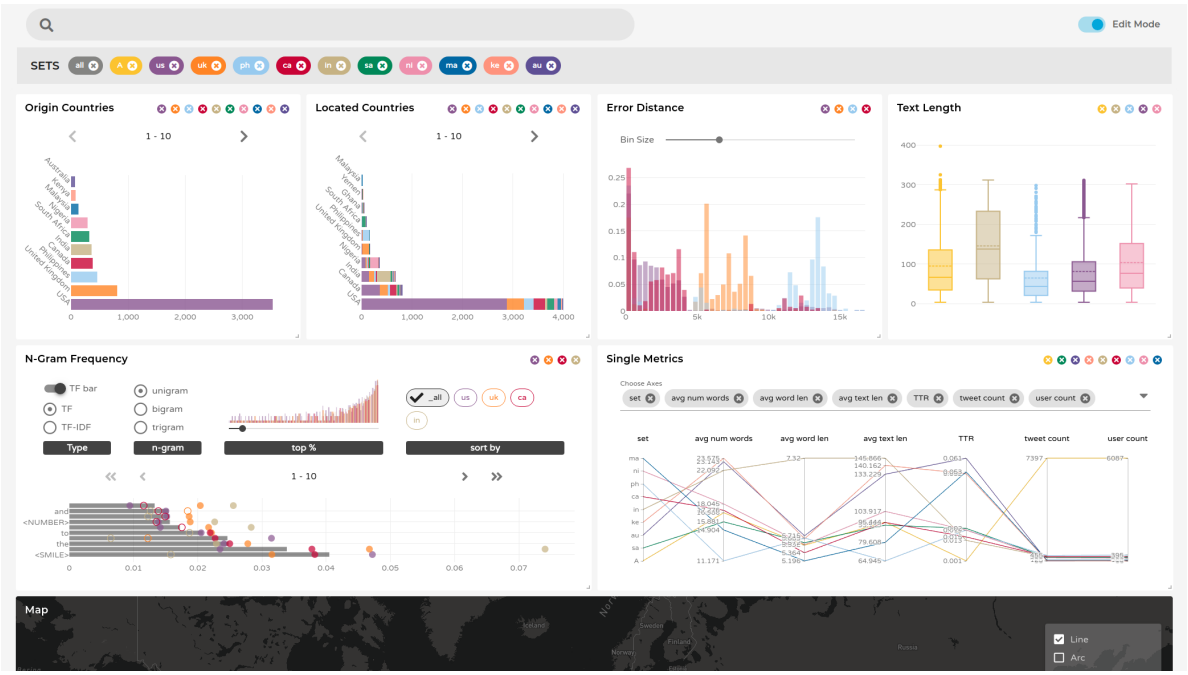
Finally, there are **summary statistics** that describe the given set with a single metric. The metrics included in the summary statistics are of 3 types: text statistics (average number of words, average length of words, average length of tweet texts, type-to-token ratio), evaluation metrics (mean & median error distances, accuracy, accuracy@161), and twitter statistics (the number of tweets, the number of users). In *Single Metrics* widget, the values for the above metrics are plotted using a parallel coordinates plot.

3.3 UI components

The interface of the implemented tool comprises mainly three components that are placed horizontally: the control bar, the set bar, and the grid layout.



(a) Default View



(b) Edit Mode

Figure 3.5: Various application views. In edit mode, set deletion and layout arrangement are enabled (Figure b).

Figure 3.6: Rearranged View.

The control bar consists of a search box and a toggle switch (Figure 3.5a A). The search box is meant for highlighting tweet inferences in each widget. The idea was to allow users to search with a keyword that is either a term or a location. Terms are to be searched in the tweet text field and locations are to be searched among the place fields. The search function, however, has yet to be implemented. A toggle switch is for triggering the edit mode. When the switch is on, the user can delete sets in the set bar and widgets, or change the size and the position of the widgets.

The set bar contains sets. A set is represented as a chip, a compact element with rounded corners (Figure 3.5a B). Each chip is labeled with its set name, and painted with its designated color. Sets can be removed with the close buttons that appear when the edit mode is on. All set chips can be dragged and dropped for set actions. Since sets are represented as simple circle icons without set names in each widget to reduce the visual clutter, the set bar also serves as a legend that represents the encoding of set names to their colors.

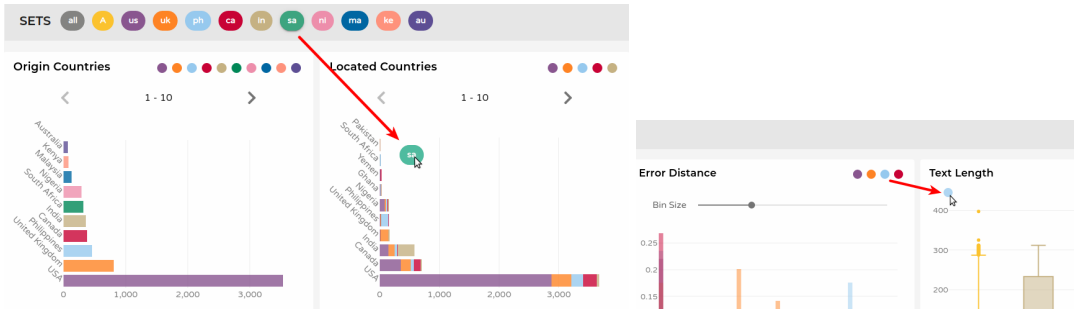
A grid layout is where widgets reside (Figure 3.5a C). Users can rearrange the layout by dragging around and resizing widgets. The layout is also responsive—its widgets change positions according— to the window width.

A widget accommodates a visualization. The title of a widget is placed on the top left. The set members added to the widget appear as circle icons in each set color on the top right corner. In edit mode, users can delete sets from widgets by clicking on the set icon with the close symbol. Furthermore, the widget size can be changed with the resize handle at the bottom right of the widget. When the widget is resized, the visualization of the widget automatically gets resized. For instance, from the view in Figure 3.5b to Figure 3.6, the *N-Gram Frequency* widget has been enlarged to better fit the visualization, whilst affecting the size of the neighboring widget *Single Metrics*. Such responsiveness of widgets and the layout gives users flexibility to adjust the interface to their needs, e.g. expanding the widget which has more comparing sets than the other widgets. Detailed information on each widget is given in the next chapter.

3.4 Set Actions

Set is an important concept in our interface based on which users interact with the data. Initially, at least one set of tweet inferences should be provided to the system as an **input**. Multiple sets can be loaded by using files with different names. From the initial input, users perform the following actions: add to a widget for visualization; delete from a widget; create a new set through an inter-set operation in a Venn diagram; and create a new set through subset selection in a widget.

Set Addtion Sets can be added to visualization widgets simply with drag-and-drop actions. Users can either drag a set chip **from the set bar** (Figure 3.7a), or a set icon **from another widget** (Figure 3.7b) onto the widget where the set should be added. The latter option allows users to add sets more easily when the target widget is distant from the set bar in the browser view, without having to scroll or drag for a long distance.

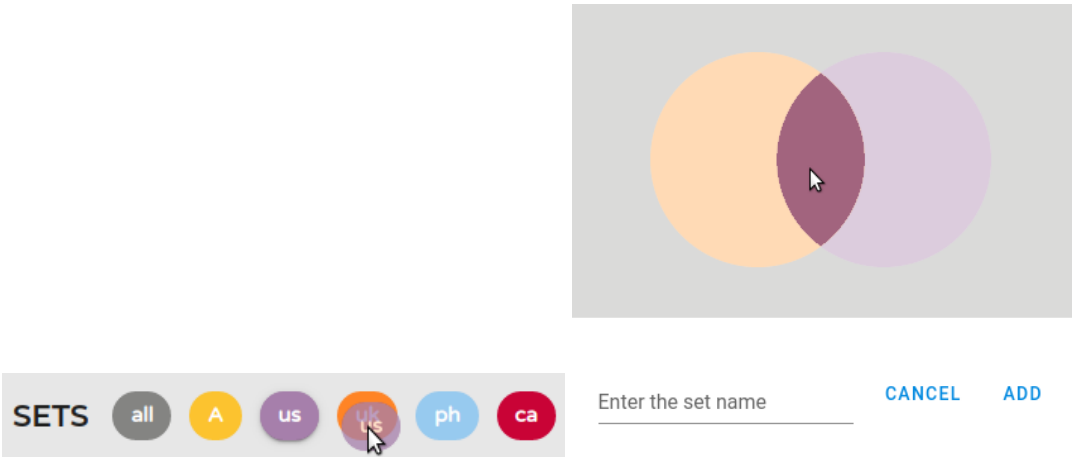


(a) From the Set Bar **(b)** From Another Widget

Figure 3.7: Two ways of adding a set to a widget: dragging from the set bar (left) and from another widget (right).

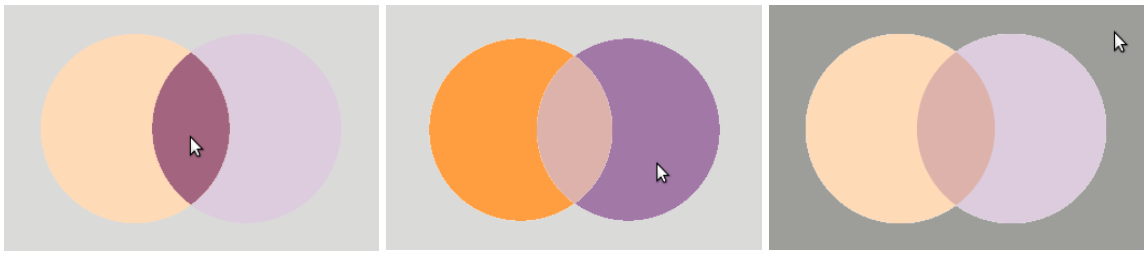
Set Deletion Set deletion in a widget can be done by clicking the close button on set chips in the set bar or set icons in a widget. The close buttons appear when the edit mode is toggled on (Figure 3.5b). When a set is deleted **from a widget**, the visualization widget gets updated correspondingly. Deleting a set **from the set bar**, on the other hand, removes the set entirely from the system. Therefore, all widgets that had the set as their members are affected, possibly resulting in multiple widget updates.

Inter-set Operation New sets can be created by **inter-set operations**. With two selected sets, users can perform set operations such as union, intersection, subtraction, or exclusive-or. For this, a set chip has to be dragged onto the other set chip in the set bar (Figure 3.8a). Then users are prompted with a dialog with a Venn diagram view (Figure 3.8b). Set operations can be performed by choosing one or more of the four parts of the diagram: (1) dragged set-only; (2) targeted set-only; (3) intersection; (4) neither of the sets (the background). As shown in Figure 3.9, the selected part of the diagram becomes darker.



(a) Triggering the Dialog **(b)** The Dialog

Figure 3.8: Set creation process via inter-set operation.



(a) Intersection **(b)** Exclusive-OR **(c)** Complement of the Union

Figure 3.9: Example set operations in a Venn diagram. The selected section is indicated by the darker color.

Widget Subset Selection As a primary way of creating a new set, subset selection is supported in two widgets, based on the sets previously added to the widget. To enable the selection mode in a widget, the SHIFT key should be pressed. The selection is done while the key is pressed. After selection, releasing the key triggers the set creation dialog. Each widget offers a unique way of choosing subsets. In *Countries* widget, where the frequencies of countries are portrayed in a bar chart, a country can be selected by clicking an a corresponding bar.

The *Map* widget, which is the main visualization for showing geospatial summary on the error patterns, provides more varieties of selection schemes. In the widget, tweets are aggregated into rose-pie charts which combines two types of charts: a pie chart representing tweets with locally inferred locations within the cluster; a rose chart representing tweets

with locations inferred in a different cluster than the the ground truth one, which is again divided into direction-indicative petals. While hovering on each component of the chart (a pie chart or a petal in a rose chart) shows the distribution of the counterpart tweet locations, further clicking on them triggers subset selection. Here the selection mode is not needed. Alternatively, the selection mode can also be used for more fine-grained level of selection. When the SHIFT key is pressed, individual tweet dots appear on their origin locations which can then be selected by brushing. More detailed descriptions on how subset selection works in the *Map* widget are provided in the next chapter.

4 Visualization Widget

The chapter consists of detailed explanations on each visualization widgets. The first two widgets (*Map* and *N-gram Frequency*) serve as main visualizations with more complex design. The rest of the smaller-scale widgets were developed to complement other widgets.

4.1 Map

The aim of the *Map* widget is to provide a geospatial overview of how errors occur. Especially, the widget is designed to help users select inferences as a subset based on their geographic features in order to further explore the existing datasets. To show the pattern of the errors with least clutter as possible, a rose-pie chart is applied to describe each tweet cluster as the main layer. Tooltips, heat maps, the minimap and additional map layers complement the main layer by providing more specifics on demand. Details on each aspect of the visualization will be given throughout the section.

4.1.1 Rose-Pie Chart

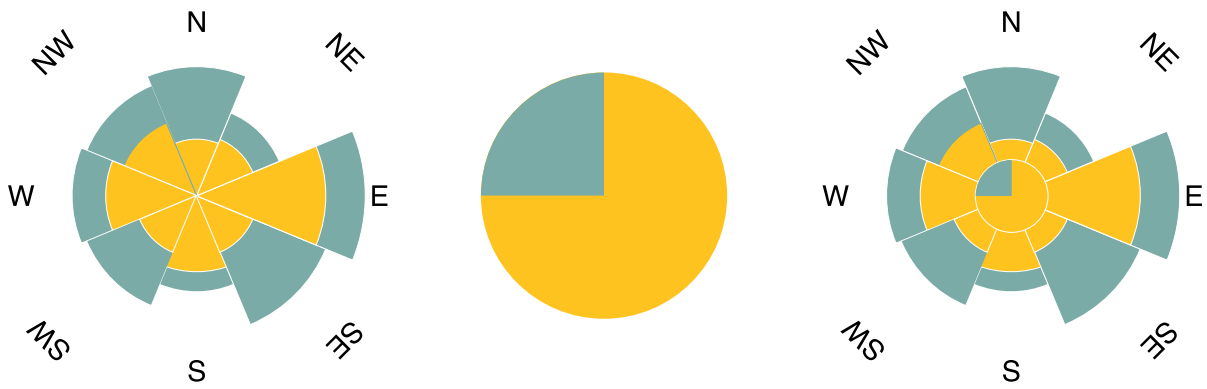
Plotting errors of tweet inferences is very similar to OD visualization. In OD flow maps, the connections between origin and destination locations are represented as lines connecting them. The trivial difference between OD data and tweet inference data is that no particular direction is implied between the two locations in a tweet inference. The problem of using lines in a map, however, is the visual clutter created by overlapping lines, as can be seen in the figure 4.8. To improve readability and aesthetics, several design principles [25, 48, 30] and alternative visualizations have been proposed [39, 49, 35]. Some of the suggested

solutions include using curved flow lines (Figure 4.8b and 4.8d) and bundled lines (Figure 4.8c and 4.8d). As shown in the figures, these solutions still have the potential to produce clutter.

Instead, an alternative visualization technique, a rose-pie chart, is proposed in this thesis. As its name implies, **a rose-pie chart** is a combination of two charts; a rose chart surrounding a pie chart (Figure 4.1). The core design decision was not to use lines at all. It was based on the fact that no physical route exists between a ground truth and an inferred location in a tweet inference unlike some traditional OD data. Similar to the approach described by van den Elzen and van Wijk [42] and Vrotsou et al. [34], rose-pie charts reduce visual clutter through aggregation and interactive filtering. In rose-pie charts, tweet inferences are aggregated into clusters to show the overview of tweet locations of the selected type (ground truth or inferred). Especially, rose-pie charts are designed to let users interactively filter the data by direction, similar to the work done by Vrotsou et al. [34]. Upon hovering or clicking on a sub-component of a chart, only the filtered results are shown or selected in an aggregated way.

There are two types of locations based on which the charts can be drawn; an inferred or a ground truth location of tweets. When the chart is drawn based on ground truth locations, the user can get information about how tweets from one region are located. The chart based on inferred locations, on the other hand, shows how tweets which had been predicted to be from one region are actually coming from.

Based on the chosen type of locations, tweet inferences are aggregated into clusters with a pre-defined pixel radius. The tweet inference whose location could not have been aggregated is drawn as a circle. For differentiation, ground truth locations are represented as filled circles (●), while inferred locations are represented as empty stroked circles (○). Aggregated tweet inferences are then described by rose-pie charts.



(a) Rose Chart **(b)** Pie Chart **(c)** Rose-Pie Chart

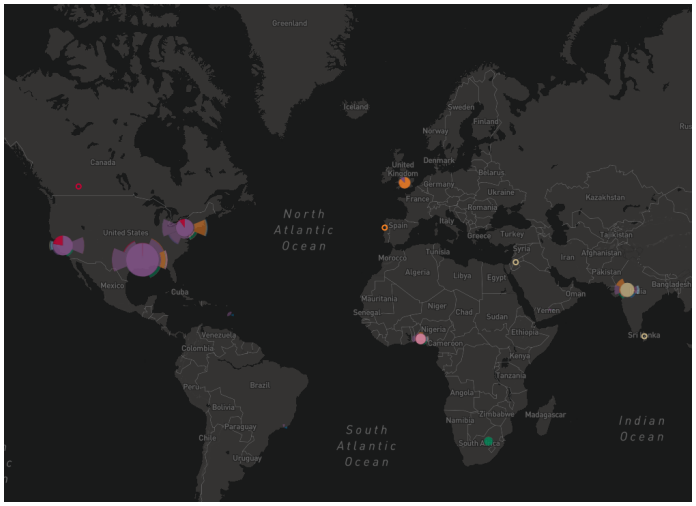
Figure 4.1: A Rose-Pie chart as a combination of a pie and a rose chart. A pie chart represents local connections while a rose chart represents non-local connections in a cluster.

A pie chart is a circle divided into pieces each of which is sized proportionally to its data size (Figure 4.1b). In this case, each pie, colored with its corresponding set color, represents local connections of the set. An inference has a local connection when both of its ground truth and inferred location are located within the same cluster on the map. While the fact that two locations fall within the same cluster does not necessarily mean that the inference was precise, it can indicate *reasonably* accurate inference when the cluster radius is small enough.

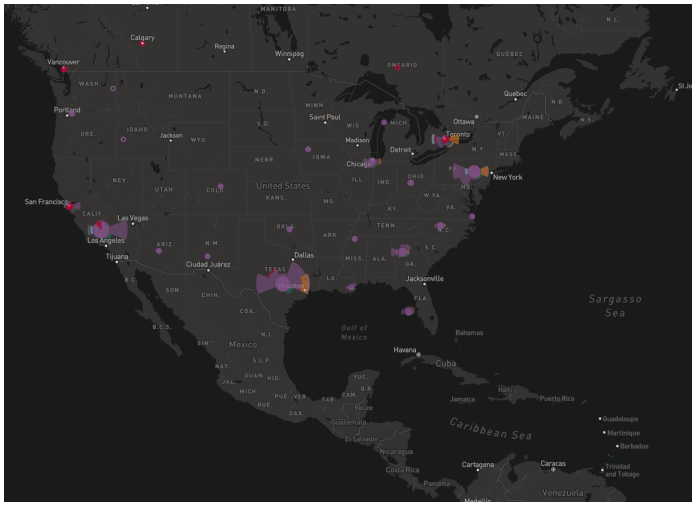
When the connection of an inference is non-local, its ground truth and inferred locations are a member of different clusters. The non-local connections of inferences are represented by rose charts. **A rose chart**, also known as a polar area chart or a coxcomb chart, is a circular chart in which each segment of equal angles represents its quantitative value by its magnitude along the polar axis (Figure 4.1a). The radial layout along with equal-sized angles in a rose chart is suitable to depict directional information. A direction is described by cardinal (north, east, south and west) and intercardinal directions (northeast, southeast, southwest, and northwest). To obtain the direction value, the bearings between the ground truth and inferred locations are calculated, which are then classified into one of the eight directions. Correspondingly, a rose chart also has 8 segments with an angle of 45 degrees to indicate 8 directions. Exhibiting multiple set values is done by horizontally stacking the segments of different sets in the same direction. It can also be interpreted as dividing the radius of a segment proportionally to the volume of a given set.

To combine the two charts, a rose chart is offsetted by the radius of the inner pie chart. The total radius of a combined chart is decided globally. In other words, each rose-pie chart is sized according to the size of its tweet cluster; the bigger the chart is, the more tweet instances it represents. Within the chart, a pie and rose charts are drawn in a way that the radius of each also reflects the amount of tweets it has. Such way of sizing charts allows users not only to get a grasp on the ratio between local and non-local inferences, but also to compare among different regions on their local and non-local inferences respectively.

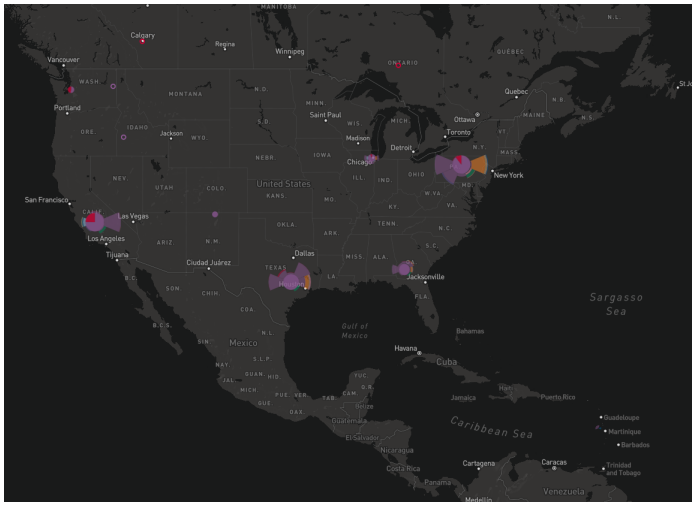
Since the radius is based on pixels, clusters are updated whenever the zoom level changes. However, this can interrupt an intuitive work flow; a user might expect the charts to remain the same while zooming in for more details on the map. For this, a toggle switch is provided to lock the current clusters. When the lock is on, charts stay the same upon the zoom level change. When the lock is turned back off, clusters are updated according to the current zoom level.



(a) before zooming in



(b) zoomed in



(c) zoomed in with locked clusters

Figure 4.2: Cluster Locking in Map. Zooming in with the cluster lock on does not change the aggregation and thus the charts.

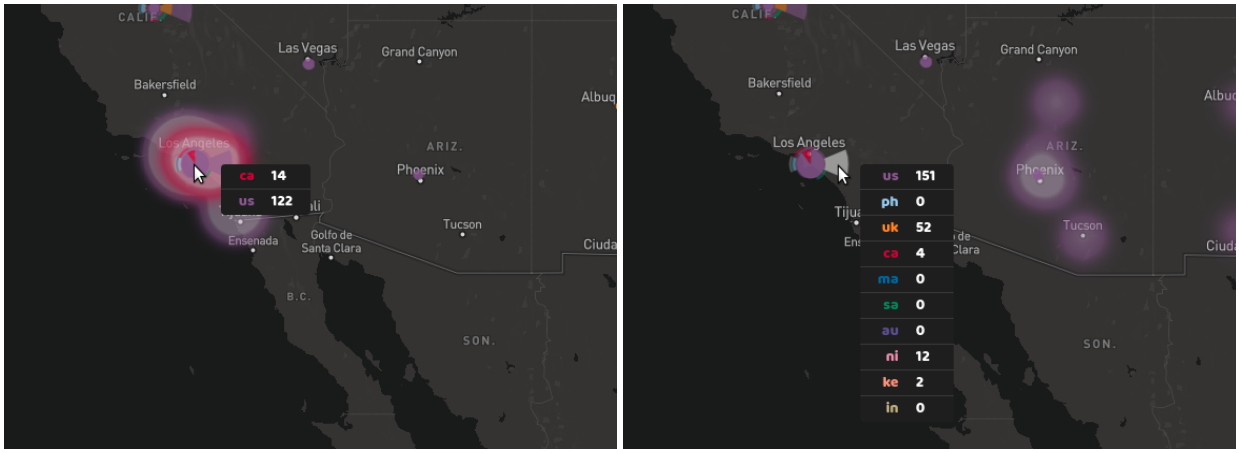
4.1.2 Details on Demand

Overall, a pie and a rose chart integrate well with each other as both have a radial layout while effectively revealing the semantic difference, i.e. inward and outward connections.

However, a rose-pie chart has inherent perceptual problems stemming from its components. In pie charts, the tweet counts are encoded in two ways: angles and areas of the slices. Perceiving and comparing sizes by these two attributes tend to be inaccurate, especially compared to length which is used in bar charts [50, 51, 52]. The problem gets even worse in rose charts. While a petal's radius is split proportionally to the set size, petals end up having segments of different shapes and areas which do not convey accurate proportions. More specifically, the segments on the outer side of the petal get larger than the inner ones and thus more emphasized. It can be problematic as how sets are ordered from its center to the edge is merely decided by the the order of their additions to the widget.

Another aspect of the charts that makes perception less accurate is their small sizes. Since the charts are overlaid on top of the map, they are bounded by its corresponding geospatial region. Additionally, the total sizes of the charts are encoded with their tweet counts. As a result, charts are drawn in very small sizes in general and the values are harder to read. Even with the charts that are relatively big, too many sets in one plot or a very small value compared to the others yields a segment which can get too small to be recognized.

A simple solution to the problem was to provide details on demand through **tooltips**. When hovering over a pie chart or a petal in a rose chart, a tooltip appears to show the list of included set names and their values (Figure 4.3).



(a) Hover on a Pie **(b)** Hover on a Petal

Figure 4.3: Two types of connections represented by a rose-pie chart and heatmaps: local connections by a pie chart (left) and non-local connections by rose petals (right).

Another detail being shown is the **heatmap** of the counterpart locations, i.e. inferred locations to their ground truth locations, and vice versa. A heatmap represents "density of points over an area" [53]. While heatmaps are not recommended for detecting accurate values, they provide brief overview of spatial data, allowing identification of minimum and maximum hotspots [54]. Not to be confused with choropleth maps where points are aggregated within fixed boundaries, heatmaps show density with continuous gradient among points. Heatmaps are chosen here since clustering occurs arbitrarily based on the zoom level and the view.

In heatmaps, the color for each pixel is decided based on its density value. The density value at each pixel is decided by summation of tweets, each of which is detected with a fixed radius around it. For this, a sequence of colors has to be defined to be mapped to each density value. The most pervasively used color mapping is the *rainbow* color scheme. However, it is generally known that the rainbow color scheme is a poor choice for a color map due to its unnatural perceived ordering and non-uniform pattern of changes [55, 56, 57]. Instead, Moreland et al. [57] suggests to use a flat scalar field that goes from less to more brightness [1]. This way of mapping colors is also applied in the widget. The range of colors are simply extracted as an array consisting of equi-distant colors along the gradient between the white color and the set color.

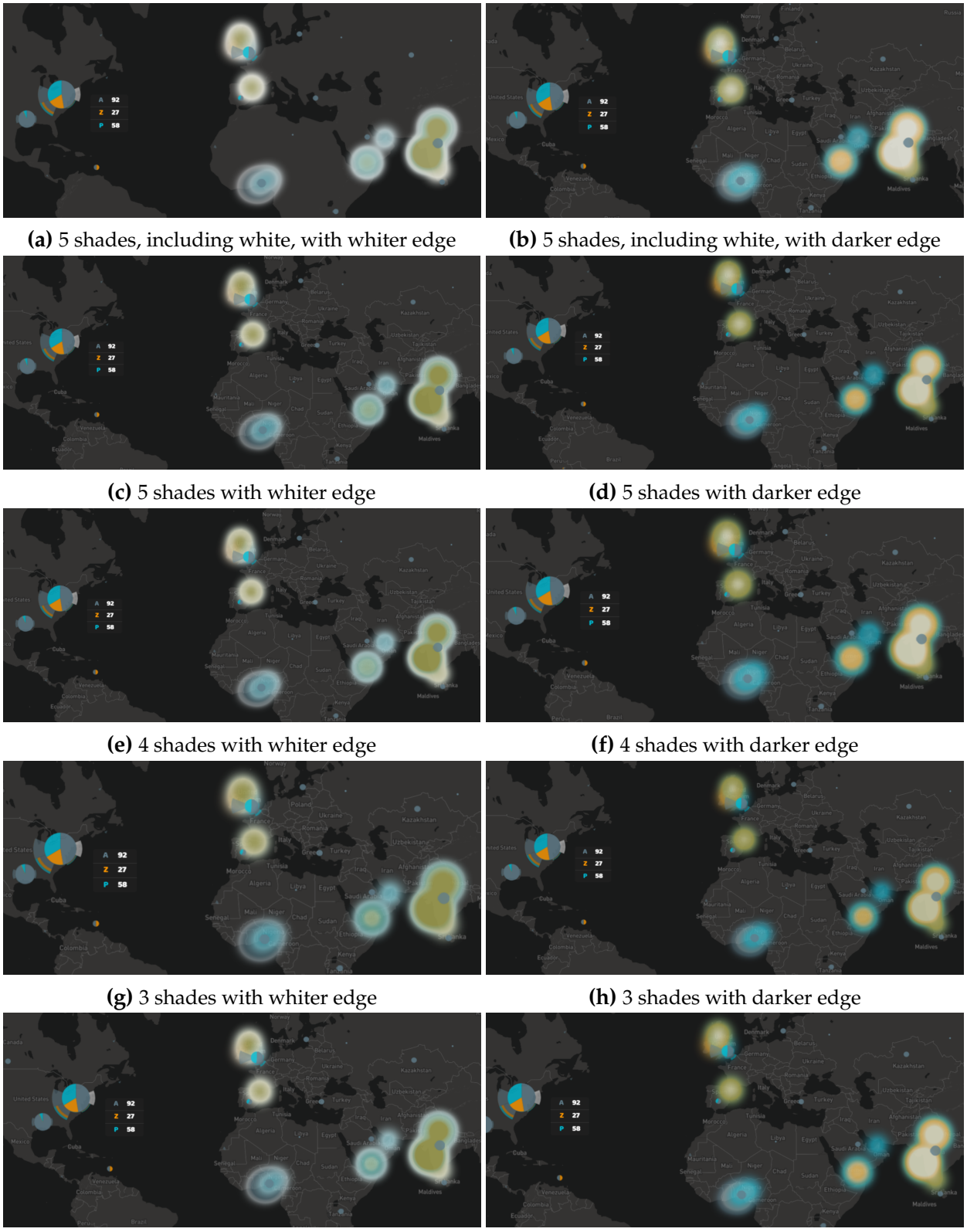
1 The minimum brightness is equal to black and the maximum equal to white

With the predefined range of colors for a set, how to relate each color to a density value was also an important design decision. The challenging aspect of the task was to clearly project areas of different sets with minimal occlusion. Figure 4.4 shows heatmaps with different density mappings. Three factors were adjusted: the usage of white, the order of shades, and the spacing of colors.

The use of white means whether to include white color—the color of maximum brightness— in the palette. While doing so fits the previous suggestion from Moreland et al. [57], it emphasizes the problem of occlusion when heatmaps are partly overlapped. In figures 4.4a and 4.4i where white was used, the two different clusters formed in the region of the UK are not clearly separated. This is because two clusters have strong halos of the same white color.

The order of shades refers to the direction in which the shades are used: the sparser the brighter; or the denser the brighter. The former yields heatmaps with brighter edges (Figure 4.4a, 4.4c, 4.4e, 4.4g, 4.4i) while the latter with darker edges (Figure 4.4b, 4.4d, 4.4f, 4.4h, 4.4j). The observation was that brighter edges naturally create stronger halos which makes it hard to distinguish partly overlapped heatmaps. Darker edges alleviate this problem by having weaker halos. For instance, two overlapping clusters of the UK are more distinguishable in the right column with darker edges than the left one with lighter edges.

The spacing of colors was experimented with different number of shades mapped. More specifically, colors were mapped differently between the mapping of the minimum (0) and maximum (1) density to create different spacings of colors. In Figure 4.4, 5 shades of color were mapped in the top four, while 4 shades were mapped in the middle two, and finally 3 shades in the last four. While the results do not indicate any significant insight that can be generalized, they served more as options for design decision. Based on discussions above, the heatmap with 3-shade mapping with darker edge (4.4h) was chosen.



(a) 5 shades, including white, with whiter edge

(b) 5 shades, including white, with darker edge

(c) 5 shades with whiter edge

(d) 5 shades with darker edge

(e) 4 shades with whiter edge

(f) 4 shades with darker edge

(g) 3 shades with whiter edge

(h) 3 shades with darker edge

(i) 3 shades, including white, with whiter edge

(j) 3 shades, including white, with darker edge

Figure 4.4: Heatmaps with varying density mappings. **(a)**, **(b)**, **(c)** and **(d)** have 5-shade mapping of color, **(e)** and **(f)** with 4 shades, and **(i)**, **(j)**, **(g)** and **(h)** with 3 shades. **(a)**, **(b)**, **(i)**, and **(j)** have white in their shades while others don't. Two columns use the inverted order of colors from each other.

One problem with the heatmap visualization is that some heatmaps located outside the zoomed view are not shown when the zoom level is too high. The user might have to pan the map to check other heatmaps, or zoom back with locked clustering to see the overview. This results in an inefficient work flow. To solve this problem, a **minimap** has been integrated into the map. The minimap is a reduced-sized overview of the base map with synchronized heatmaps. In Figure 4.5, the heatmaps for the hovered petal are only partially shown in the US in the main map view, but fully shown in the minimap.

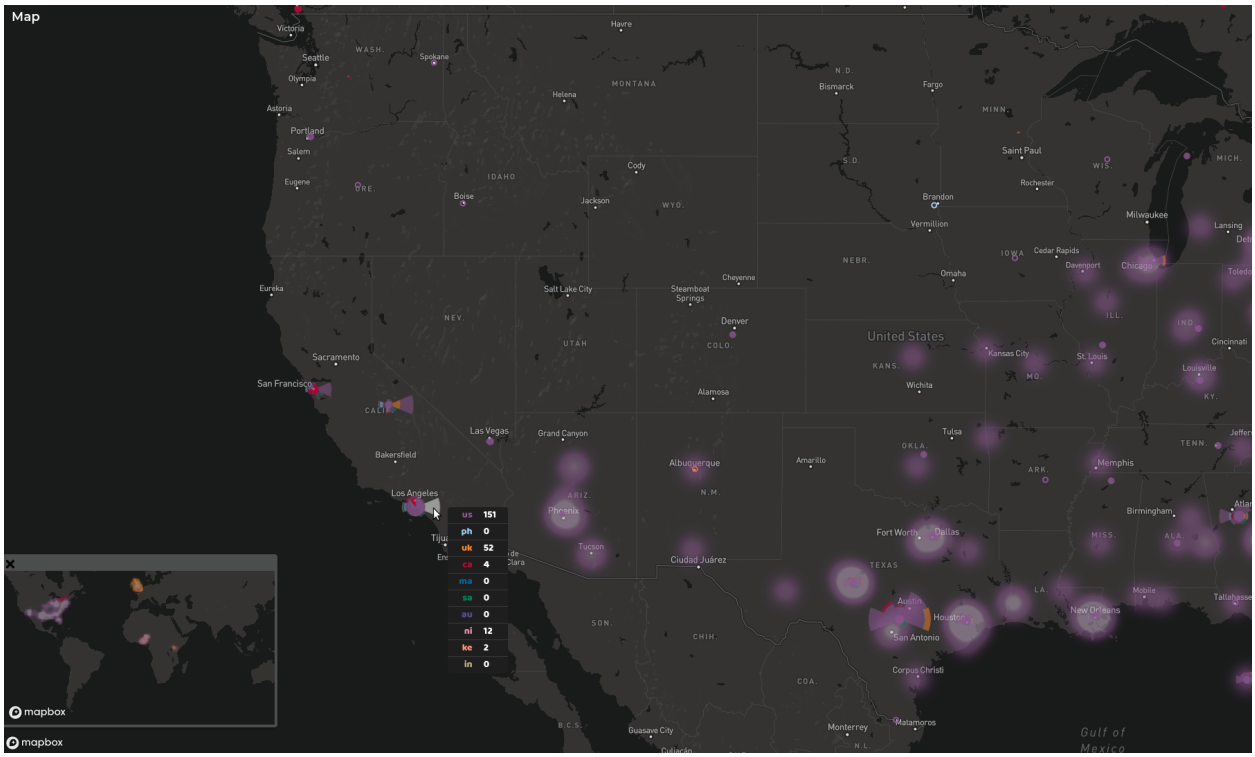
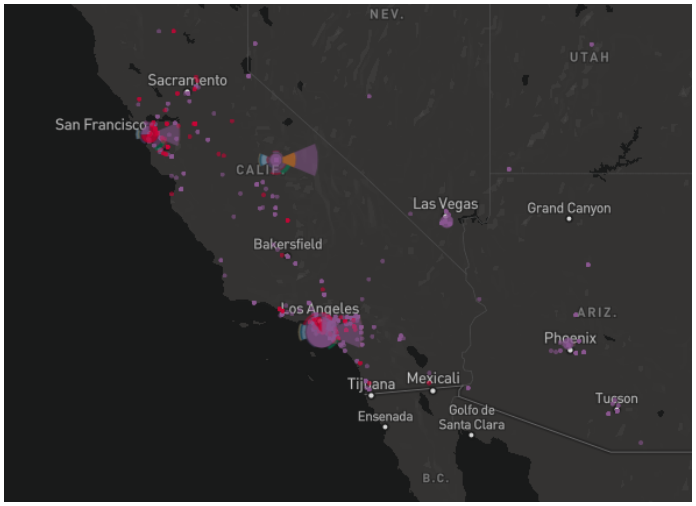


Figure 4.5: Minimap showing the overview.

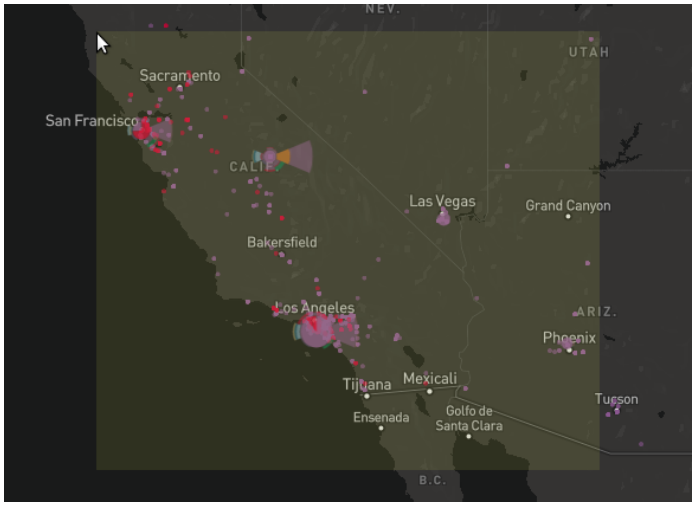
The minimap can be toggled open by clicking on a globe icon placed at the bottom left corner of the map. Initially, the map has the lowest zoom level to show a reasonably good overview of the world in a small window. The minimap is updated on each hover event to have synchronized heatmap visualization as the main widget map. While direct manipulation of data is not possible, basic interactions such as panning or zooming is enabled in the minimap.

Subset selection are supported in two ways. One way is to click a segment in a chart,

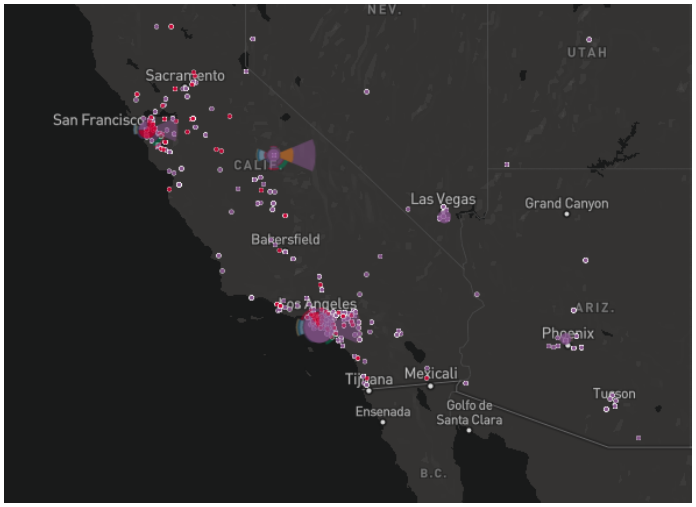
either a petal or a pie chart. Another way to select a subset is by brushing in the selection mode. The selection mode is triggered when the SHIFT key is pressed, showing the dots representing the ground truth locations of tweets (Figure 4.6a). A region is selected by dragging while the key is still pressed (Figure 4.6b). The tweet dots within the selected area are then highlighted (Figure 4.6c). When the SHIFT key is released, the set creation dialog appears. This way of subset selection allows users to freely choose the area of interest, which is limited in the main visualization as clusters for charts are automatically created.



(a) Selection Mode Triggered



(b) Brushing



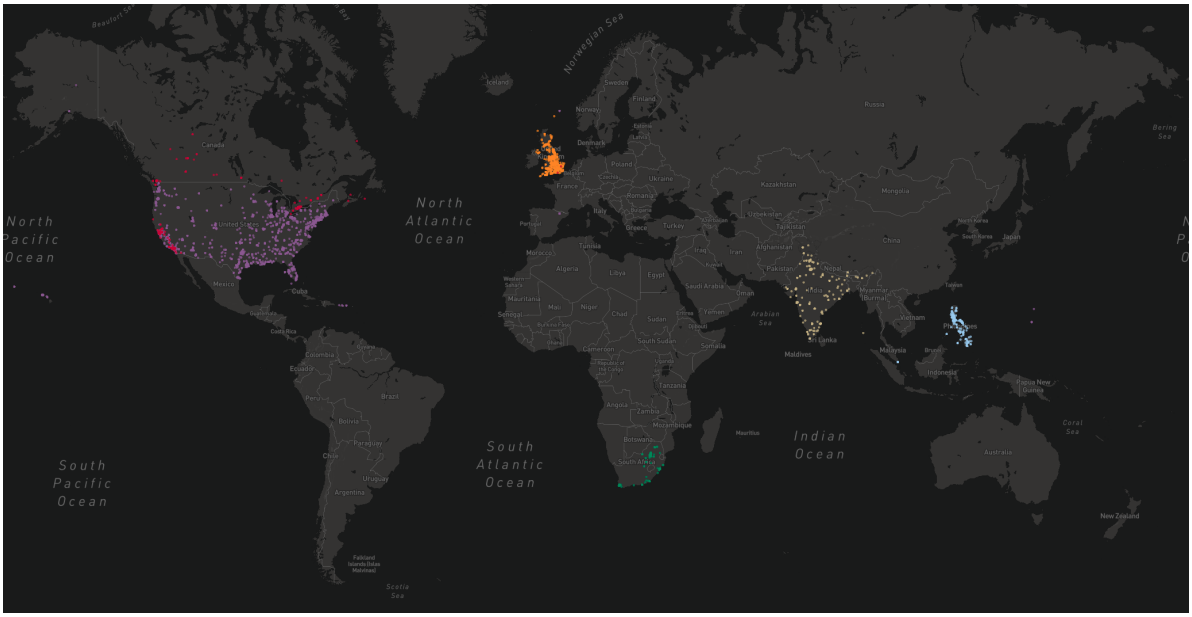
(c) Selected

Figure 4.6: Set selection process by brushing in the Map widget.

4.1.3 Additional Layers

While rose-pie charts serve as the main visualization in the Map widget, there are other types of visualization that can be layered on top of each other. Two types of layers are provided: a dot layer and a line layer. These layers have initially been made as the preliminary process of designing the final visualization, Rose-pie charts. Nonetheless, each layer still remains as an option to be drawn as it can be useful when details are needed.

dot In a dot layer, each tweet is represented as a dot, or a small circle. Figure 4.7 shows dot layers with two different types of locations from tweet inferences: ground truth and inferred locations. When hovered, a tooltip appears showing essential attributes of a tweet inference: a user's screen name, a tweet text, place names of ground truth and inferred locations, and an error distance. As shown on the figures, dot maps suffer from occlusion problems, especially on the low zoom level. Nonetheless, they complement the main visualization by showing exact distribution of tweets, the detail missing in both rose-pie charts and heatmaps.



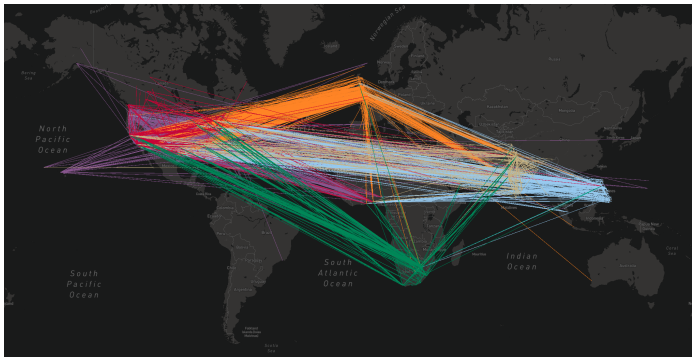
(a) Dots for Ground Truth Locations of Individual Tweets

(b) Dots for Inferred Locations of Individual Tweets

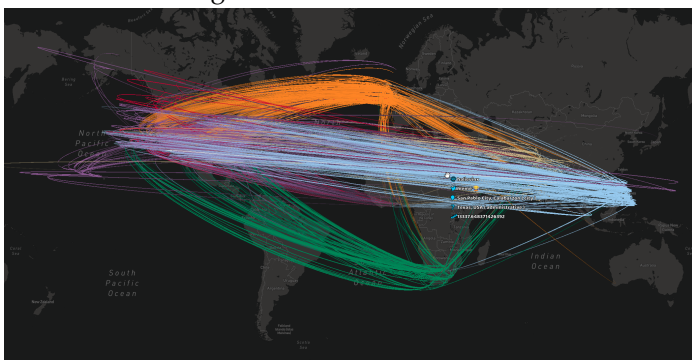
Figure 4.7: Dot Layers in Map. Dots represent ground truth locations (left) or inferred locations (right) of each tweet.

line In a line layer, two locations are simply connected by a line, as shown in Figure 4.8. As pointed out earlier, line layers can easily get cluttered and therefore hard to read. Using the curved lines (Figure 4.8b) improves readability compared to the straight lines

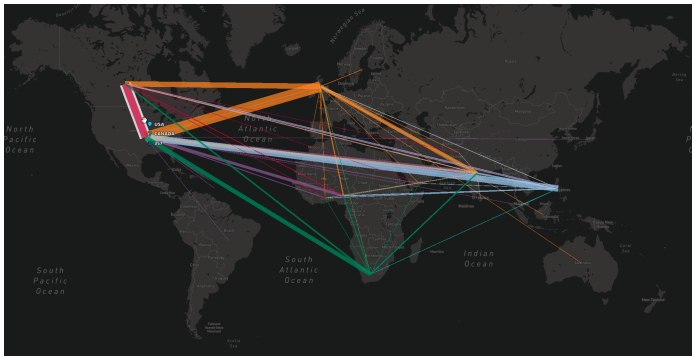
(Figure 4.8a), as suggested in Jenny et al [30]. Lines can also be aggregated to reduce clutter, showing country-to-country connections (Figure 4.8c and 4.8d). The same tooltip from the dot layer appears on a line connecting individual tweets. An aggregated line has a different information on its tooltip: a country of its ground truth tweets, a country of its inferred tweets, and a number of tweets (Figure 4.8d).



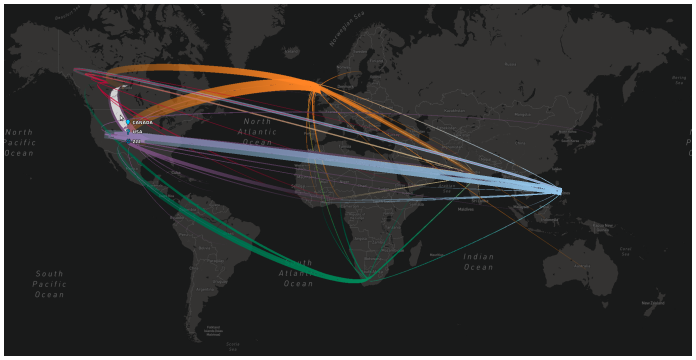
(a) Straight Lines for Individual Tweets



(b) Arcs for Individual Tweets



(c) Straight Lines for Connections between Countries



(d) Arcs for Connections between Countries

Figure 4.8: Line Layers in Map. **(a)** and **(b)** show connections between ground truth and inferred location in individual tweets. In **(c)** and **(d)**, connections are aggregated on country level.

4.2 N-Gram Frequency

The widget *N-Gram Frequency* displays n-grams based on their frequencies. The visualization combines a scatter plot and a bar chart with the shared axes. The y-axis represents n-grams and the x-axis represents the frequency values. While the scatter plot is used to indicate frequencies of individual sets, the bar chart is used for the combined value of entire sets. As the list of terms can get extensive, only 10 terms are shown in one page. The rest of the section explains the two data types (n-grams and frequencies) and how the look of the markers change according to the user-defined values in filtering and sorting.

4.2.1 N-Grams

An n-gram is a sequence of n words extracted from a text. In the widget, three sizes are available for an n-gram: 1-gram (unigram), 2-gram (bigram), and 3-gram (trigram). Before the n-grams are generated, each tweet text goes through basic preprocessing. The process involves converting all letters from upper case to lower case and replacing specific elements (e.g. URLs, hashtags, mentions, emojis and numbers) with the standardized words that define their types (e.g. <URL>, <HASHTAG> and <USER>). This way, the resulting n-grams are ensured to have some level of consistency despite their input text being noisy and informal.

One thing to note here is that, n-grams are to maintain the noisiness of tweets as much as possible. This is because locality can be indicated by small and subtle differences in expressions, beyond the different use of vocabularies. Therefore, some common approaches taken in other Natural Language Processing (NLP) applications, e.g. stemming (reducing words to their stems), lemmatization (reducing words to their dictionary forms), or stopword removal (removing commonly used words like 'the' or 'is'), are not employed here.

4.2.2 Frequency Measures

There are two measures of frequency by which the data values are determined; term frequency (TF) and term frequency-inverse document frequency (TFIDF).

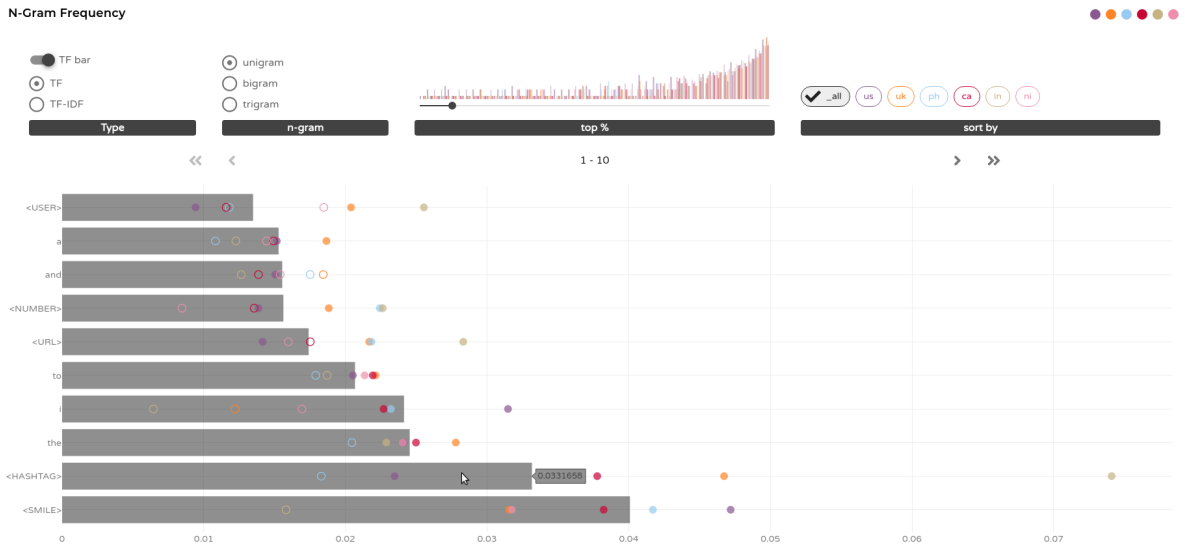
TF indicates how frequently the given n-gram appears in a document (Figure 4.10 and 4.9). Since most tweets are short and noisy, tweets in a set are treated as a single document, rather than individual documents. These documents are likely to be of different lengths, which can result in certain n-grams appearing more in larger documents than smaller ones. Therefore, TF is adjusted by dividing the raw frequency count of an n-gram by its document length, namely the number of all n-grams in a set.

Some n-grams frequently appear across all tweet corpus, e.g. unigrams like *'i'* and *'and'* or bigrams like *'it is'* and *'in the'* are commonly used in english tweets. These terms likely have high TFs, but are not necessarily meaningful. What would be important is the term occurring uniquely in one set. For instance, users might be interested in words that are used in India but not in the US, and vice versa. To better convey the significance of terms in each set, **TFIDF** can be used to define frequency, which takes the presence of a term in a document into account (Figure 4.12). Calculating TFIDF requires inverse document frequency(IDF). IDF measures how unique the term is across all documents, which can be calculated as follows:

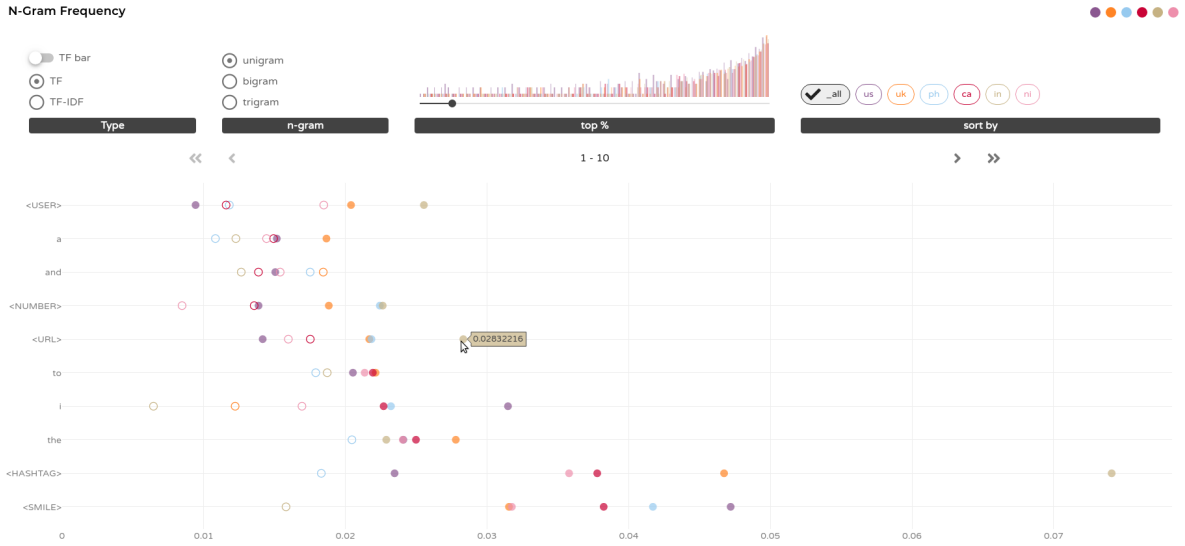
$$IDF(\text{n-gram}) = \ln \frac{(\text{total number of sets})}{(\text{number of sets where the n-gram occurs more than once})}$$

Then the TFIDF is the product of TF and IDF.

One of the differences between a TF and a TFIDF is that TFIDF calculation is dependent on the set membership in the widget while a TF of a term in a set can be calculated independently of other sets. For this reason, TF values are precomputed when the set is created while TFIDF values are computed only after the set is added to the widget. Consequently, TFIDF values of already-existing sets are updated as well each time the set membership changes in the widget, i.e. set addition or deletion.



(a) TF Bar Mode On



(b) TF Bar Mode Off

Figure 4.9: N-gram frequency visualization with (left) or without (right) the TF bar.

An additional value is the TF value for the entire tweet corpus in the widget, named *_all* [2]. In other words, the ***_all*** set is the set combining all sets added to the widget. The TF values of *_all* set are provided for comparison between the term significance within a set and within the whole corpus. Similar to TFIDF, the *_all* TF values also get updated when a set is added or deleted in a widget. On the other hand, TFIDF values for *_all* set are not

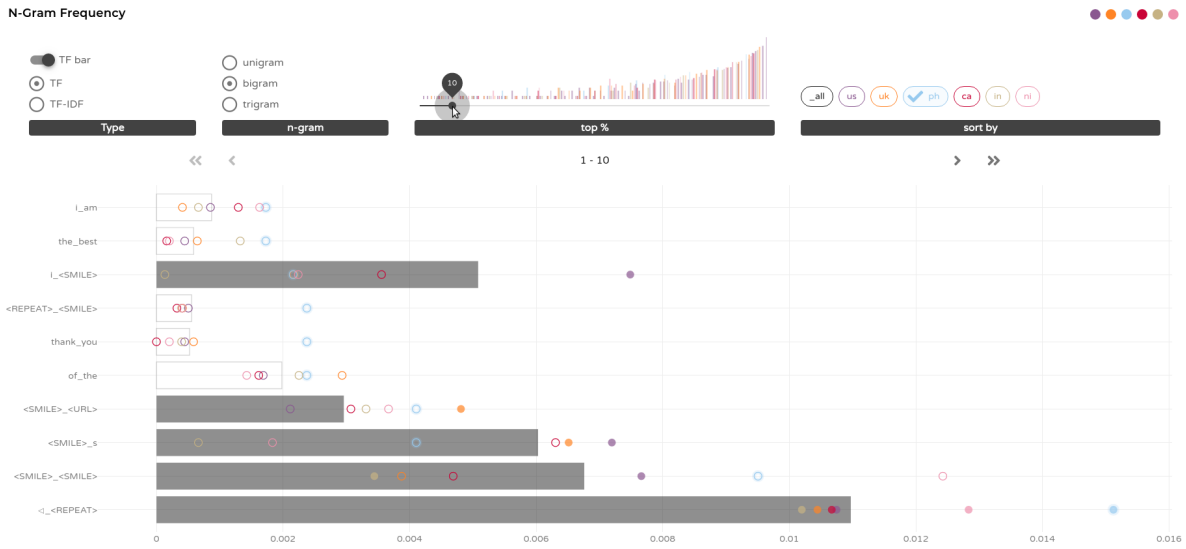
2 It is a locally combined set, which should be differentiated from the *all* set in the set bar, which is the globally combined set.

calculated. This is because TFIDF measures the relative importance of terms, and TFIDF calculated on the universal set, i.e. the set of all possible values, does not come across as meaningful.

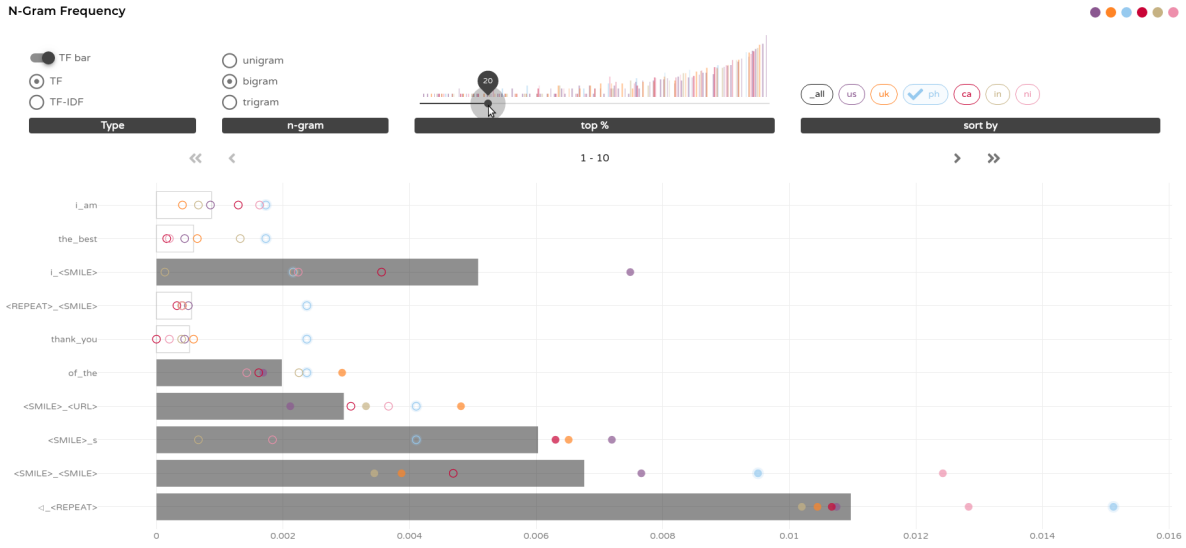
To be differentiated from values of individual sets, the values of *_all* set is represented as a bar, rather than a circular marker in the scatter plot. The bar chart shares both x- and y-axis with the scatter plot. The visibility of the bar can be toggled with the switch at the top left corner of the widget (Figure 4.9). Since TFIDF value does not exist for *_all* set, the TF bar always shows the TF value even in TFIDF mode. In this case, the scatter plot showing TFIDF values have a different scale from the bar chart, thus the bar's length cannot be directly compared to the positions of scatter plot markers and should be perceived as an independent reference.

4.2.3 Percentile Filtering

The markers in the scatter plot and the bar in the bar chart can either be filled or empty. The filling of the markers is parameterized by the percentile value. Given the x th percentile, a filled marker indicates that the corresponding n-gram's frequency value falls within the top x % of all values in the set (Figure 4.11b and 4.11d). When the value falls above the percentile, the marker remains empty (Figure 4.11a and 4.11c). The aim of percentile filtering is to help users compare the significance of a certain term among different sets. For instance, a term might appear in two sets A and B with similar frequencies, but can be placed differently in each percentile rank and thus considered more significant in one set than the other.



(a) 10th Percentile



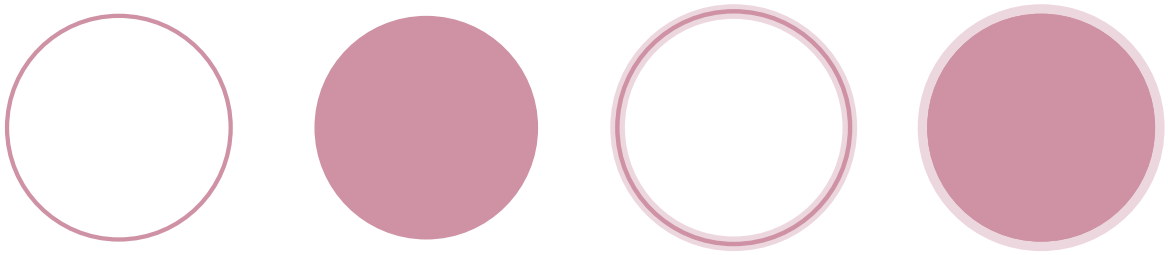
(b) 20th Percentile

Figure 4.10: N-gram frequencies with two different percentile values: 10th (left) and 20th (right).

While the default percentile value is set to 10th, it can be adjusted by users through the slider above the plot. Right above the slider, a histogram is placed to help users to choose the percentile based on the distribution of the frequency values in each set. The x-axis of the histogram represents the percentile, ranging from 1 to 99, and the y-axis is the frequency count of all n-grams whose frequency values fall right on the given percentile. Bars are overlaid with semi-transparency. Tooltips show each set values on hover, which

can be useful in a small plot.

Figure 4.10 shows the same chart with different percentile values. More markers are filled with higher percentile value of 20th (Figure 4.10b), compared to the 10th percentile (Figure 4.10a). For example, the n-gram *'of the'* is found in the top 20% of frequent terms in the sets *_all* and *'uk'*, but not in their top 10%.

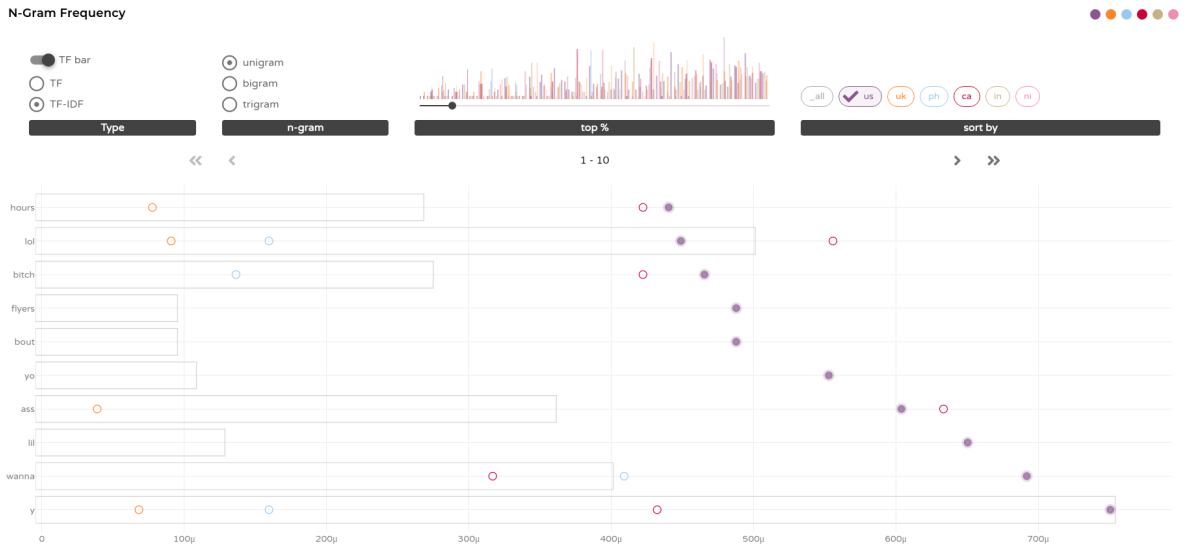


(a) Filtered Out **(b)** Filtered In **(c)** Filtered Out & Sorted **(d)** Filtered In & Sorted

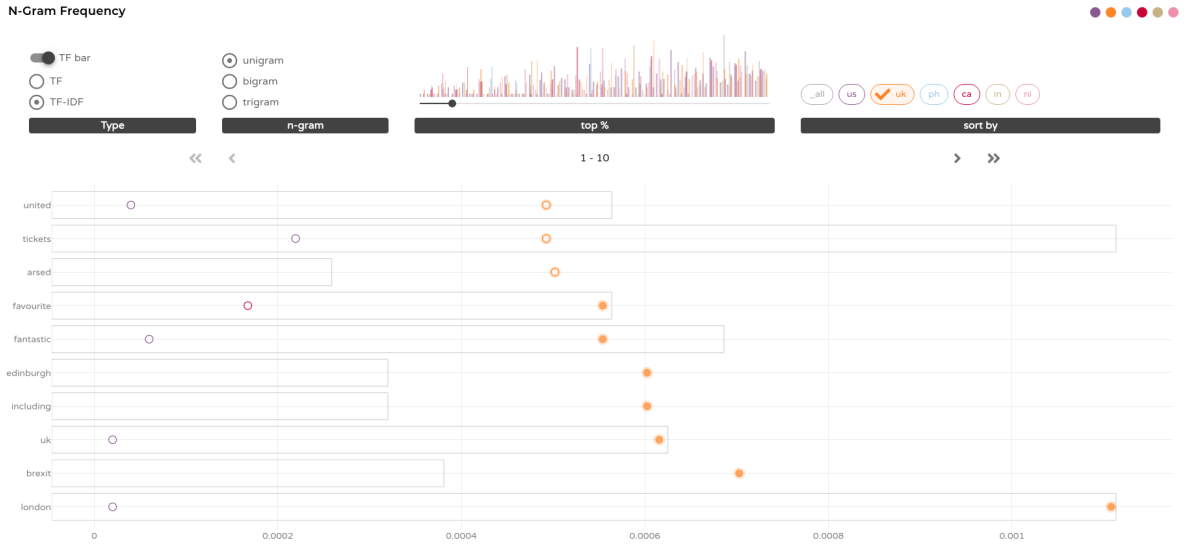
Figure 4.11: 4 types of dot markers used in the scatter plot.

4.2.4 Sorting

By default, the n-grams are sorted by their TF values in the *_all* set. Alternatively, the sort order can be adjusted to the values of a certain set. This allows users to see how terms that are frequent in one set are in other sets. The sort order can be changed by selecting the corresponding chip in the *sort by* option, placed at the top right corner of the widget. Each chip indicates the set by its color and label, which altogether can serve as a chart legend in case the widget is placed distantly from the set bar. Since the TFIDF values for the *_all* set are not available, as explained earlier, its chip is disabled when TFIDF mode is on. When a set is selected as the sort option, the circular markers in the plot representing the set gets highlighted with halos around them (Figure 4.11c and 4.11d). In Figure 4.12, the highlighted markers represent the chosen set; purple markers of *'us'* set in (a) and orange markers of *'uk'* set in (b).



(a) Sorted by 'us'



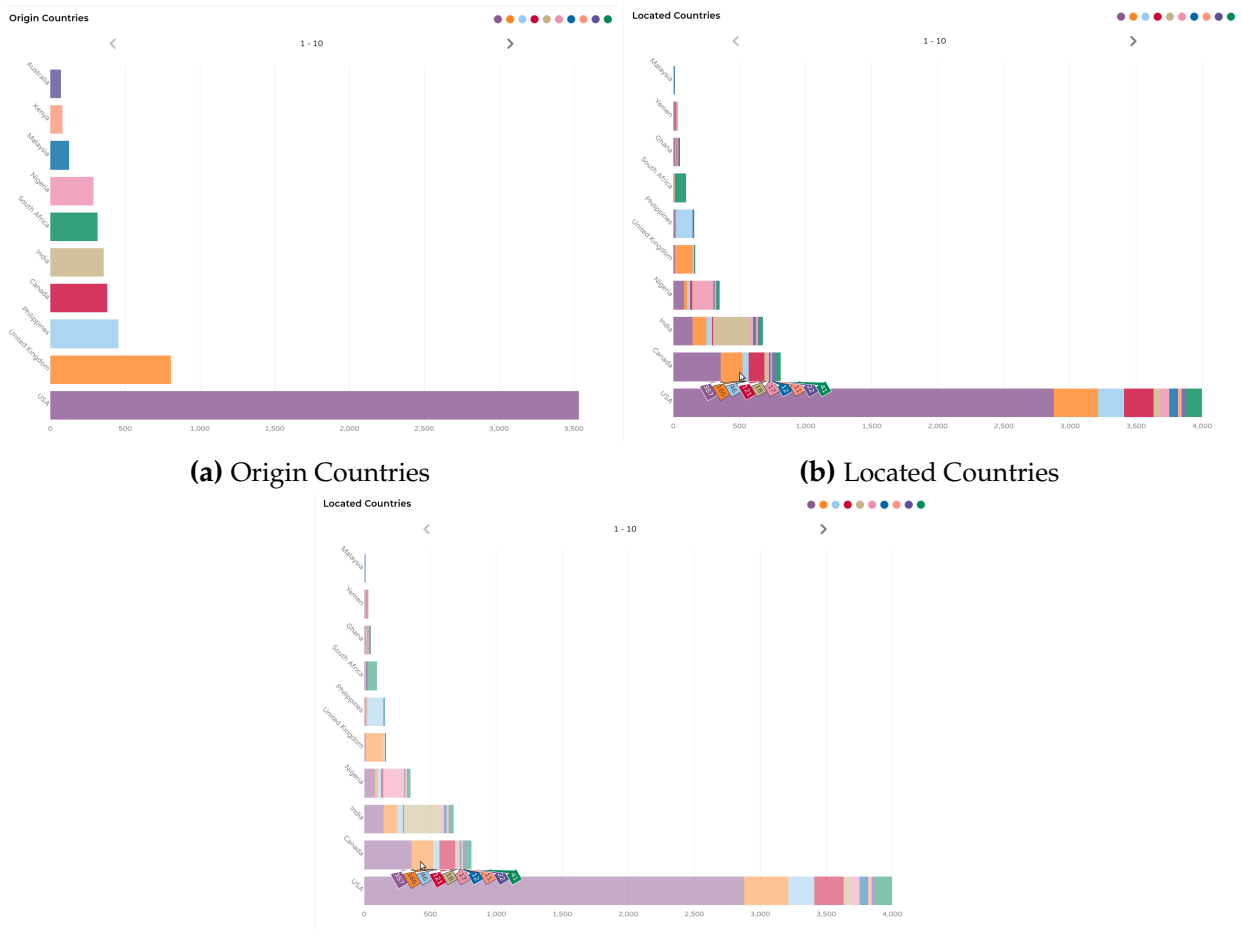
(b) Sorted by 'uk'

Figure 4.12: N-grams sorted by different sets. Here each set name refers to the origin country of its members.

4.3 Countries

There are two widgets featuring country statistics; *Origin Countries* and *Located Countries*. The main objective of the widgets is to show the most frequent countries out of all sets and comparison of frequencies among sets. Countries from ground truth locations are

depicted in *Origin Countries* widget while countries from inferred locations are in *Located Countries*.



(a) Origin Countries

(b) Located Countries

(c) Selection Mode

Figure 4.13: Two types of widgets showing country statistics in a bar chart: (a) Countries of ground truth locations, (b) Countries of inferred locations. The last one shows semi-transparent bars in the selection mode.

For easy comparison of numerical values among countries as a categorical attribute, the frequency of each country is compared with each other in a bar chart. In the charts, the y-axis and the x-axis represent countries and the number of occurrences respectively. Accordingly, each bar length is proportional to the frequency of the country. The bars are horizontally drawn and the tick labels of country names are rotated 45 degrees counterclockwise to provide enough space for long country names (e.g. 'United Arab Emirates').

To accommodate multiple sets, stacked bar charts were used. While having countries as the primary category, the bars from different sets of the same country get stacked end-to-end. The total length of the bar then shows the overall frequency count across all sets. However, it is often difficult to compare values of different sets within one bar when the stacked bar is too small compared to others. As a compromise, tooltips show each stacked values in a bar on hover (Figure 4.13b).

The bars are ordered by the count value, starting from the bottom. That is to say, the bottommost bar always has the highest count as well as the biggest length. For simplicity and less clutter, only ten data items are shown in one plot. Users can navigate to other pages to see different ranges of frequency counts.

In the bar charts, a subset can be selected by clicking on a bar in the selection mode. The selection mode is being activated while the SHIFT key is pressed. In the selection mode, the bars are semi-transparent except for the selected one (Figure 4.13c). When the user releases the SHIFT key after selection, a dialog is displayed to prompt the user to input the set name.

4.4 Text Length

The *Text Length* widget shows the distribution of text lengths in each set. To efficiently compare distributions of multiple sets at a glance in a limited space, box plots are used. While the text length is already included as an attribute in the *Single Metrics* widget, a box plot shows more information by depicting how the numeric values are spread out and how skewed they are by grouping data points according to the quartiles. It is especially useful when two datasets have the same measures of central tendency (i.e. mean, median, and mode) but have different dispersion of the data. Each box plot is drawn vertically and constructed of three elements; a box, a set of whiskers and outlier dots.

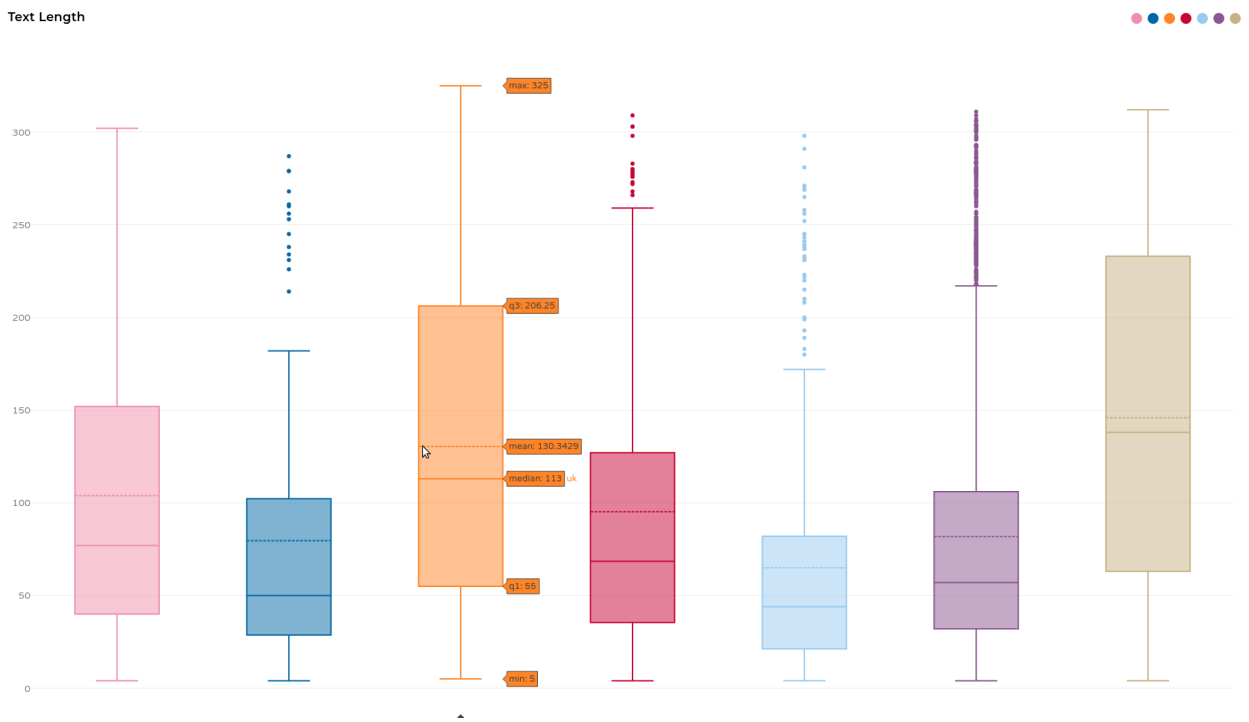


Figure 4.14: A Text Length Widget with 7 Sets.

A box is drawn with four numbers; the first quartile, the median, the mean, and the third quartile. The first quartile (q1) forms the lower end of the box, while the third quartile (q3) the upper end of the box. Inside the box, a line is drawn where the median (q2) is. Additionally, the mean is also represented as a dotted line.

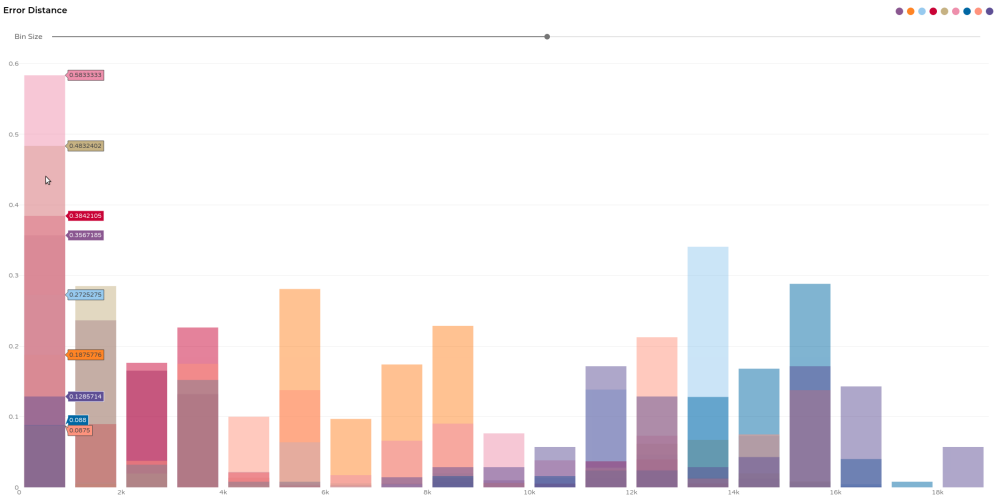
To indicate variability outside these quartiles, a set of whiskers are drawn as lines extending from each side of the box. For this, the interquartile range (IQR) is calculated as the distance between the first and the third quartile. Then the whiskers range from each end of the box (i.e. the first and the third quartiles) to each data point that fall within the distance of 1.5 * IQR.

Finally, any data point outside the whiskers becomes an outlier. Outliers are represented as filled dots. When there are no such outliers, the lower whisker simply extends all the way down to the minimum text length and the upper whisker up to the maximum.

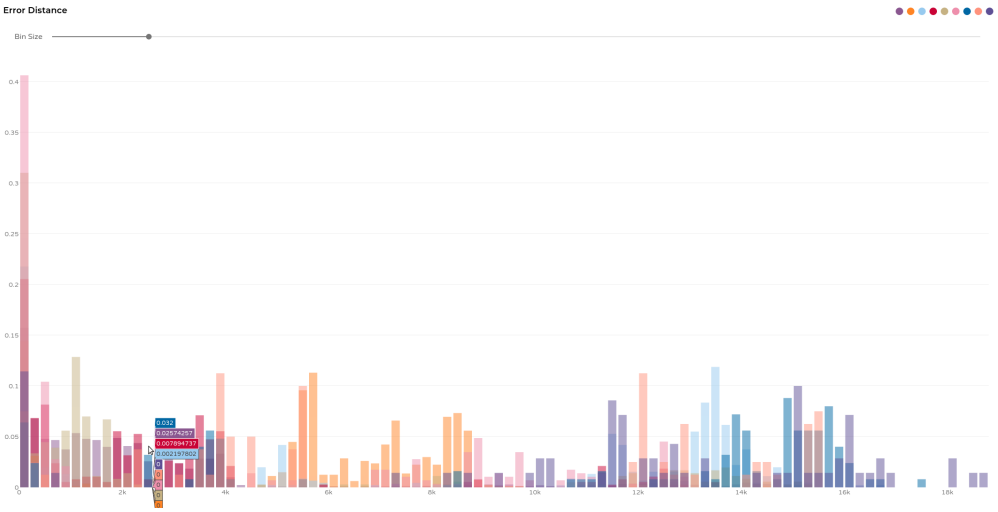
When hovering over a box, all numbers constructing the box appear as tooltips; maximum, q3, median, mean, q1, and minimum. In addition, the ends of the whiskers are specified as upper/lower fence if there are outliers.

4.5 Error Distance

The *Error Distance* widget illustrates the distribution of error distances using histograms. An error distance here refers to the difference between the ground truth and inferred location in a tweet measured in kilometers. For this, the whole range of error distances are divided into a sequence of non-overlapping intervals, called bins, by the user-defined bin size. Then a bar is placed for each bin whose height denote the frequency of the error distance range in a given set.



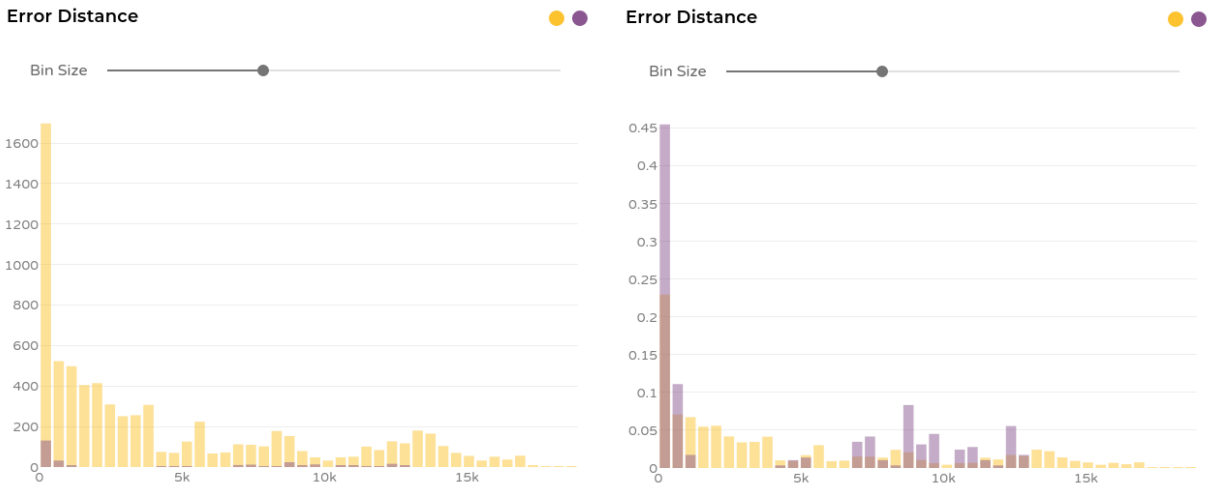
(a) Bin Size = 1000 km



(b) Bin Size = 200 km

Figure 4.15: Histograms in the Error Distance Widget with Varying Bin Sizes: (a) a histogram with the default bin size of 1000; (b) a smoother histogram with the smaller bin size of 200.

Users can make the histograms flatter or smoother by changing the bin size in a slider. The slider changes the bin size by 5 km each step, with its default size being 1000 km. Figure 4.15 shows histograms with two bin sizes. While the minimum bin size in a slider is set to 5, the maximum value is recalculated each time a new set is added, in a way that the histograms have at least 10 bins. In addition, tooltips appear on hover at each bar to show the distance range for the bin (x-axis) and the probability (y-axis). This is useful when exact numbers are required or the values are hard to be compared due to overplotting or being unrecognizably small compared to others (Figure 4.15b).



(a) Frequency as an Error Distance

(b) Probability as an Error Distance

Figure 4.16: Comparison between Frequency and Probability as an Error Distance.

Instead of merely counting the number of data values to show frequency, the bar heights are normalized to be relative frequencies, or probability. A probability can simply be computed by dividing the number of inferences in a bin by the total number of inferences in an entire set. As a result, the sum of all bar heights become 1, and each height is now interpreted as the probability of a tweet inference having the error distance within the range. The normalization is needed for accurate comparison of error distance distributions among sets of different sizes. Figure 4.16 shows that distributions of error distances are easier to be compared when probability is used to define an error distance (Figure 4.16b) than when frequency is used (Figure 4.16a).

4.6 Single Metrics

The *Single Metrics* widget concurrently exhibits several metrics each of which summarizes the given set with a single value. As a way of visualizing multivariate data of multiple sets, a parallel coordinates plot is used.

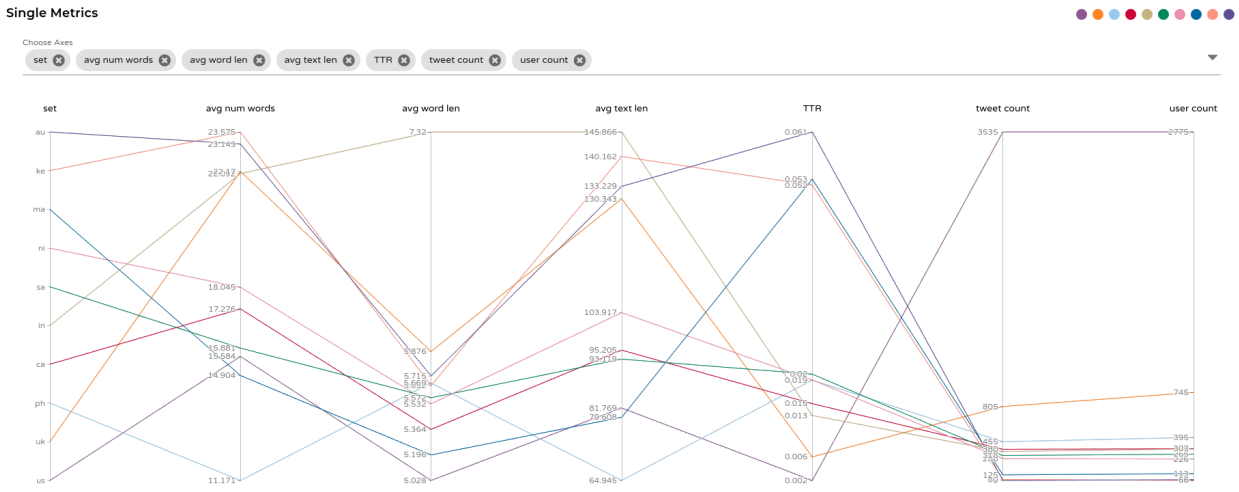


Figure 4.17: Single Metrics Widget.

In the plot, an axis is drawn vertically for each metric. Then the values of each set are plotted across each axis and connected as a line. As a result, multiple lines of sets colorfully intersect with each other to reveal the relationship among sets with different attributes. Each axis has different units and ranges. For example, type-to-token ratio is represented as a percentage (decimal) ranging from 0 to 1, while the tweet count is always a positive integer. In any case, the range of the axis is defined by its minimum and maximum data value.

There are three types of metrics that define the attributes; text statistics, evaluation metrics, and twitter statistics. The text statistics include average number of words, average length of words, average length of tweet texts, and type-to-token ratio. The evaluation metrics are mean and median error distances, accuracy, and accuracy@161. As suggested by Mouroud et al. [19], two types of granularity are provided for accuracy: country level and fine-grained level. On the fine-grained level, the names of locations at their finest level are compared (e.g. *Phoenix, AZ, US* versus *Austin, TX, US*), whereas only the countries of

locations are compared on the country level (e.g. *US vs. Canada*). Finally, the number of tweets and users are provided as twitter statistics.

Plotting all 12 attributes can clutter the widget view with the limited space. To reduce the clutter, users can filter out unnecessary axes through the select input field at the top of the widget. Along with the axis filtering, interactions techniques like brushing and reordering of axes are also supported for clutter reduction. Brushing technique highlights the lines that belong to the sets of interest that are within the dragged ranges in any axes, while fading out the other lines. This is useful when there are too many plotted lines (set items) in a widget or in a specific range. To better reveal the patterns of attribute values, reordering of axes can be done by dragging an axis across the widget.

5 Evaluation

To evaluate the effectiveness, ease of use, and overall usability of the implemented visual analytics tool, an expert review was given. As an expert in the field of social media analysis and natural language processing, the researcher at the Webis Group tested the implemented web application from a target user's perspective. Since he participated in the development process of the application as a consultant, it was assumed that he already had enough understanding of how the application functions.

The expert carried out the test by freely exploring the dataset in his personal working environment without any tasks or time limit. The dataset used here was the location inference results from his research team. Then the review was given in an interview guided by open-ended questions on each component, e.g. "How useful did you find the widget?" The rest of the chapter presents the feedback content organized by the related feature.

Edit Mode The edit mode was found interruptive to workflow. The user toggles on the edit mode mostly to delete sets, but then forgets to toggle it off. This often led to an unintentional behavior of widgets. For instance, he tried to pan the map by dragging in the *Map* widget or change the bin size on a slider in the *Error Distance* widget, but instead the whole widget was dragged due to the edit mode that he had forgot to turn off. Such inconsistency of widget behavior between the on and off states of the edit mode, as well as the fact that a user needs to toggle the switch back to off, was a major factor for inefficient workflow. The suggestion was to get rid of the edit mode and make editing (i.e. set deletion and widget rearrangement) possible in the default mode. It is also worth noting that the user didn't use the edit mode to rearrange the layout. While he sometimes resized widgets only to fit them to his screen size, he did not feel the need to personalize the view by changing how widgets are laid out.

Set Actions The drag-and-drop interactions by which sets are communicated felt natural to the user. However, he considered widget-to-widget set addition is redundant with set addition from the set bar. When it comes to set creation, he found the automatic color designation and its color scheme to be convenient and aesthetically pleasing respectively. At the same time, he expressed that automatic set naming would have been convenient as well, as it is annoying to specify the set name each time a new set is created.

Subset Selection The different ways of selecting subsets were easy to apply. The most used techniques when selecting subsets for the user were bar selection in the *Countries* widget and dot selection in the *Map* widget. What he found missing in subset selection was the multi-selection feature. In the *Countries* widgets, for instance, he wanted to select the two bars of the UK and the US at the same time but was not possible. While the similar task can be done in the Venn diagram dialog by selecting their union [1], multi-selecting two sets would be more efficient by reducing 3 steps (creating the US set — creating the UK set — and selecting their union) to only 1 step (creating the UK & the US set). The selection schemes applied in two widgets felt complementary to each other in that selection in a bar chart offers more accurate selection while selection in a map offers less accurate but more intuitive selection.

Map The user discovered the visualization in the *Map* widget to be satisfactory in terms of both usefulness and aesthetics. He especially liked the functionality and the overall look of rose-pie charts. The separate representation of the local and non-local connections in rose-pie charts were easy to understand. The minimap also came in handy when the view with the high zoom level does not show heatmaps on the regions hidden from the view. While the interactive filtering and aggregation of data via hovering and heatmaps gave the user a good impression on the error patterns, he did not necessarily *"care to use"* tooltips much. However, the inverted color scheme in heatmaps does not come as intuitive and the heatmaps were still hard to understand when overlapped. In addition, it was against his expectation that the charts did not scale up on zoom in when the clusters are locked. Additional layers were almost never used. When individual tweet dots were needed to be displayed, the user just pressed the SHIFT key to quickly turn on the selection mode to see

1 The union of two sets and the their combined set are not exactly the same. In the combined set, the intersection between the two sets are redundantly included unlike in the union.

the origin dots. Some features like arc lines *"looked cool"* but did not convey any useful message.

N-Gram Frequency The user had a positive experience using the *N-Gram Frequency* widget. Especially, each of frequency measure, TF and TFIDF, had their own merit. To be specific, he found that TF values are more intuitive and suitable for getting the first impression of the data, while TFIDF values offer more interesting insights on unique term usages in each set. On top of that, sorting function was also significantly convenient to quickly understand the differences among sets. The chart immediately adapted to show different information with the adjusted sorting, from which he barely felt the need to navigate to the second or the third page. It was especially a pleasant surprise for the user to recognize some linguistic differences among different regions using these features in the widget. For example, he could find out that some specific slangs like 'bro' or 'damn' are dominantly used in the US, while words like 'lovely' or 'liverpool' appears more frequently in the UK. On the other hand, the user found percentile filtering too complicated to understand. He could not exactly grasp what information he could gain from the feature, and mostly used the feature to remove the clutter in the chart.

Single Metrics While the *Text Length* widget was found to be least useful, the user suggested to integrate it with the *Single Metrics* widget; by allowing a metric to be dragged from the *Single Metrics* widget to the box plot which shows more details on that metric. Interaction techniques such as brushing and axis reordering in the *Single Metrics* widget *"did not really shine"* as the parallel coordinates plot does not really have to accommodate high dimensional data due to the 20 number limit of creatable sets.

Apart from criticisms on specific aspects of each feature and component, the user also pointed out the inconsistency of interactivity across widgets. While some widgets felt highly interactive (its visual items being mostly draggable or clickabale), other widgets did not. For example, he expected a chip describing a metric in the *Single Metrics* widget to be reactive when clicked, but *"nothing happpned."*

6 Conclusion and Future Work

In this thesis I developed a visual analytics tool that allows interactive exploration of the results from location inference methods for social media data. The primary aim of the tool is to provide deeper insights into how well the algorithms infer locations, beyond comparisons using conventional evaluation metrics such as error distances or accuracy. Its design invites users to follow an iterative process of visualizing, comparing and creating new *sets* with its interaction schemes. A dashboard layout is used to accommodate various types of visualizations as widgets including the map and the n-gram chart.

To achieve interactivity, the concept of a *set*—an interactional dataset— was introduced to represent a group of location-inferred tweets. Sets can be dynamically added to or deleted from widgets to be compared with other sets. Furthermore, new sets can be created by performing set operations in a Venn diagram (intersection, subtraction, exclusive or, and union) or by selecting a subset in widgets such as *Map* or *Countries*. Through the process of selectively visualizing and defining sets, users can discover similarities and dissimilarities among sets of different qualities, e.g. what the linguistic differences are between two sets of tweets from the same region with different inferred countries.

As the main visualization, the *Map* widget presents the geospatial aspects of tweet inferences using aggregation and interactive filtering. To reveal the patterns of errors with minimal visual clutter, rose-pie charts were proposed. A rose-pie chart represents tweets aggregated based on their geographic proximity in a combined form of a rose and a pie chart: a pie chart portrays the locally inferred tweets within the cluster and thus indicates the tolerably correct inferences, while a rose chart represents wrongly inferred tweets, where the directions of errors are manifested in each petal. Each component of a chart shows its geographic error distribution as heatmaps on hover, which can also be selected

as a subset. Features such as cluster locking, tooltips, the minimap, and additional layers are included as well to complement the main layer.

The *N-Gram Frequency* widget serves as another main visualization for text analysis, as this thesis targets analysis of text-based inference methods. In the widget, scatter plots are used to show the comparison of the frequency values of n-grams among each set, along with the bar chart indicating the value of the combined sets. To allow users to flexibly investigate the term usage, various options are provided to rearrange the chart: the type of frequency value, the size of n-gram, the percentile value for filtering, and the way terms are sorted.

Aside from the two widgets, 4 more types of complementary widgets are provided. The *Origin Countries* and *Located Countries* widgets use bar charts to represent the frequencies for countries of ground truth and inferred locations respectively, where each bar can be selected to create a subset. In the *Text Length* widget, the distribution of text lengths in each set are described in a box plot. The *Error Distance* widget shows the distribution of error distances by overlaying histograms of each set with an adjustable bin size. Finally, in the *Single Metrics* widget various metrics, each of which describes a set with a single value like an average number of words or a median error distance, are displayed for each set in the axis-filterable parallel coordinates plot.

In the review given by an expert, most visualizations were easy to understand and interacted with as expected. The two main widgets especially were appraised to be effective in detecting patterns across sets and aesthetically appealing to work with. In the *Map* widget, rose-pie charts incorporated with the minimap and heatmaps were considered to be a better alternative in displaying pairs of locations, compared to other types of visualizations such as a flow map. The *N-Gram Frequency* widget was deemed useful for discovering local usage of words; the user was able to find the differences of the frequently used terms in tweets between the US and the UK using both TF and TFIDF frequency values, as well as the sorting option.

Selecting subsets appeared to be an important part of the work flow, as was intended in the design. Two types of subset selection were found to complement each other: bar selection, which is suitable for more accurate selection; and map brushing, which is more intuitive. The most often used method of choosing a subset was by turning on the selection mode with the SHIFT key, compared to the selection by clicking on chart components.

This indicates that interaction techniques for subset selection across widgets should be consistent.

One criticism was that the constant need to toggle on and off the edit mode to delete sets or rearrange the layout leads to unnecessary disturbance to the work flow. For better user experience, the features in the edit mode can be merged into the default mode in the next version of the application. In addition, percentile filtering in the *N-Gram Frequency* widget was hard to understand with intuition and was not found useful. The support for multi-selection in subset creation was also found missing as an important feature.

While the review from the expert offered a glimpse of how users would interact with the application, the assessment of the tool should be conducted more thoroughly with a broader range of users, to see how the requirements differ according to their inference methods. Furthermore, datasets of different sizes and a variety of inference algorithms should be used to test the application's general effectiveness.

Some features which were parts of the design but could not have been implemented yet should be included in the further development. As already mentioned in Chapter 3, the search function which enables users to search and highlight tweets of certain terms or locations is missing. The main challenge here will be to find ways to highlight the searched result in different visualizations with aggregated tweets. For a more complete and consistent interaction experience, subset selection should be supported in the remaining widgets apart from the *Map* and *Countries* widgets. Similar to set colors, the set names can also be automatically designated for more efficient workflow.

The application can be expanded by adding more types of visualization widgets. For example, a widget simply showing the list of tweets in each set would be useful when users want to view the individual tweet contents within sets. Other widget ideas are to apply the types of OD visualizations other than the map. As discussed in Chapter 2, 3D flow maps or non-geospatial plots such as OD matrices, Sankey flow diagrams, and Chord diagrams can be adopted as independent widgets or integrated with other visualizations. As some inference methods utilize different contextual attributes of tweets such as posting time or user profile information, new widgets can be developed to adapt such aspects of data.

While the visual analytics tool in this thesis only focused on location inference methods with tweet-level text-based approaches, it can be made more generally applicable by considering methods with different approaches, e.g. user-level inference methods with network-based or hybrid approaches. Moreover, the application can be improved by considering different granularities of locations, as most inference methods are based on a specific level of location granularity. For instance, frequency bar charts can be made not only for countries but also for states, cities or even POIs. Another idea is to allow users to dynamically update the map visualization by adjusting the granularity level based on which tweets are aggregated in the map.

Bibliography

[1] Hootsuite & We Are Social. Digital 2020 global digital overview. `http://datareportal.com/reports/digital-2020-global-digital-overview`, 2020. [Accessed: 20-05-2020].

[2] Jie Bao, Defu Lian, Fuzheng Zhang, and Nicholas Jing Yuan. Geo-social media data analytic for user modeling and location-based services. *SIGSPATIAL Special*, 7(3):11–18, jan 2016.

[3] Narayan CHATURVEDI, Durga TOSHNIWAL, and Manoranjan PARIDA. Twitter to Transport: Geo-Spatial Sentiment Analysis of Traffic Tweets to Discover People's Feelings for Urban Transportation Issues. *Journal of the Eastern Asia Society for Transportation Studies*, 13:210–220, dec 2019.

[4] Philip N. Howard, Bence Kollanyi, Samantha Bradshaw, and Lisa-Maria Neudert. Social Media, News and Political Information during the US Election: Was Polarizing Content Concentrated in Swing States? feb 2018.

[5] A. J. Morales, J. Borondo, J. C. Losada, and R. M. Benito. Measuring political polarization: Twitter shows the two sides of Venezuela. *Chaos*, 25(3):033114, mar 2015.

[6] Xiangyang Guan and Cynthia Chen. Using social media data to understand and assess disasters. *Natural Hazards*, 74(2):837–850, oct 2014.

[7] Chao Fan, Fangsheng Wu, and Ali Mostafavi. A Hybrid Machine Learning Pipeline for Automated Mapping of Events and Locations from Social Media in Disasters. *IEEE Access*, 8:10478–10490, 2020.

[8] Twitter. Tweet geospatial metadata. `https://developer.twitter.com/en/docs/tutorials/tweet-geo-metadata`. [Accessed: 20-06-2020].

[9] Xin Zheng, Jialong Han, and Aixin Sun. A Survey of Location Prediction on Twitter. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1652–1671, 2018.

[10] Bo Han, Afshin Rahimi, Leon Derczynski, and Timothy Baldwin. Twitter geolocation prediction shared task of the 2016 workshop on noisy user-generated text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 213–217, 2016.

[11] Reid Priedhorsky, Aron Culotta, and Sara Y Del Valle. Inferring the origin locations of tweets with quantitative confidence. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 1523–1536, 2014.

[12] David Flatow, Mor Naaman, Ke Eddie Xie, Yana Volkovich, and Yaron Kanza. On the accuracy of hyper-local geotagging of social media content. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 127–136, 2015.

[13] Mark Dredze, Miles Osborne, and Prabhanjan Kambadur. Geolocation for twitter: Timing matters. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1064–1069, 2016.

[14] Clodoveu A Davis Jr, Gisele L Pappa, Diogo Rennó Rocha de Oliveira, and Filipe de L. Arcanjo. Inferring the location of twitter messages based on user relationships. *Transactions in GIS*, 15(6):735–751, 2011.

[15] David Jurgens. That's what friends are for: Inferring location in online social media platforms based on social relationships. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[16] Mohammad Ebrahimi, Elaheh ShafieiBavani, Raymond Wong, and Fang Chen. Twitter user geolocation by filtering of highly mentioned users. *Journal of the Association for Information Science and Technology*, 69(7):879–889, 2018.

[17] Bo Han, Paul Cook, and Timothy Baldwin. Geolocation prediction in social media data by finding location indicative words. In *Proceedings of COLING 2012*, pages 1045–1062, 2012.

[18] Chieh-Yang Huang, Hanghang Tong, Jingrui He, and Ross Maciejewski. Location prediction for tweets. *Frontiers in Big Data*, 2:5, 2019.

[19] Ahmed Mourad, Falk Scholer, Walid Magdy, and Mark Sanderson. A practical guide for the effective evaluation of twitter user geolocation. *ACM Transactions on Social Computing*, 2(3):1–23, 2019.

[20] Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287, 2010.

[21] Bo Han, Paul Cook, and Timothy Baldwin. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, 49:451–500, 2014.

[22] Fernando Melo and Bruno Martins. Automated geocoding of textual documents: A survey of current approaches. *Transactions in GIS*, 21(1):3–38, 2017.

[23] Milan Gritta, Mohammad Taher Pilehvar, and Nigel Collier. A pragmatic guide to geoparsing evaluation. *Language Resources and Evaluation*, pages 1–30, 2019.

[24] Ahmed Mourad, Falk Scholer, and Mark Sanderson. Language influences on tweeter geolocation. In *European Conference on Information Retrieval*, pages 331–342. Springer, 2017.

[25] Waldo R Tobler. Experiments in migration mapping by computer. *The American Cartographer*, 14(2):155–163, 1987.

[26] Satoshi Kawamura, Yoshimitsu Tomita, Masahiko Itoh, Daisaku Yokoyama, Masashi Toyoda, and Masaru Kitsuregawa. An effective use of tokyo metro passengers flow by visualization of smart card ticket 'pasmo'origin-destination data for public transport network to be sustainable. *Proc. WECC*, 2015.

[27] Arthur H Robinson. The 1837 maps of henry drury harness. *The Geographical Journal*, 121(4):440–450, 1955.

[28] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on visualization and computer graphics*, 12(5):741–748, 2006.

[29] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE transactions on visualization and computer graphics*, 14(6):1277–1284, 2008.

[30] Bernhard Jenny, Daniel M Stephen, Ian Muehlenhaus, Brooke E Marston, Ritesh Sharma, Eugene Zhang, and Helen Jenny. Design principles for origin-destination flow maps. *Cartography and Geographic Information Science*, 45(1):62–75, 2018.

[31] Diansheng Guo and Xi Zhu. Origin-destination flow data smoothing and mapping. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2043–2052, 2014.

[32] Yalong Yang. Visualising geographically-embedded origin-destination flows: in 2d and immersive environments. *arXiv preprint arXiv:1908.00662*, 2019.

[33] ST Eick. Aspects of network visualization. *IEEE computer graphics and applications*, 16(2):69–72, 1996.

[34] Katerina Vrotsou, Georg Fuchs, Natalia Andrienko, and Gennady Andrienko. An interactive approach for exploration of flows through direction-based filtering. *Journal of Geovisualization and Spatial Analysis*, 1(1-2):1, 2017.

[35] Yalong Yang, Tim Dwyer, Bernhard Jenny, Kim Marriott, Maxime Cordeil, and Haohui Chen. Origin-destination flow maps in immersive environments. *IEEE transactions on visualization and computer graphics*, 25(1):693–703, 2018.

[36] Ennio Cascetta. Estimation of trip matrices from traffic counts and survey data: a generalized least squares estimator. *Transportation Research Part B: Methodological*, 18(4-5):289–299, 1984.

[37] Wei Zeng, C-W Fu, Stefan Müller Arisona, Alexander Erath, and Huamin Qu. Visualizing waypoints-constrained origin-destination patterns for massive transportation data. In *Computer Graphics Forum*, volume 35, pages 95–107. Wiley Online Library, 2016.

[38] Nikola Sander, Guy J Abel, Ramon Bauer, and Johannes Schmidt. Visualising migration flow data with circular plots. Technical report, Vienna Institute of Demography Working Papers, 2014.

[39] Jo Wood, Jason Dykes, and Aidan Slingsby. Visualisation of origins, destinations and flows with od maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[40] Aidan Slingsby, Jason Dykes, and Jo Wood. Using treemaps for variable selection in spatio-temporal visualisation. *Information Visualization*, 7(3-4):210–224, 2008.

[41] Daniel M Stephen and Bernhard Jenny. Automated layout of origin–destination flow maps: Us county-to-county migration 2009–2013. *Journal of Maps*, 13(1):46–55, 2017.

[42] Stef Van den Elzen and Jarke J Van Wijk. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2310–2319, 2014.

[43] Erick Cuenca, Fr éd éric Docquier, Siegfried Nijssen, and Pierre Schaus. Evoflows: an interactive approach for visualizing spatial and temporal trends in origin-destination data. 2019.

[44] Ieva Dobraja and Menno-Jan Kraak. Principles of dashboard adaptability to get insights into origin-destination data. *Journal of Location Based Services*, pages 1–21, 2020.

[45] Twitter. Introduction to tweet json. `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json`. [Accessed: 20-06-2020].

[46] P Green-Armytage. A Colour Alphabet and the limits of colour coding. *JAIC-Journal of the International Colour . . .* , (5):1–23, 2010.

[47] Kenneth L Kelly. Twenty-two colors of maximum contrast. *Color Engineering*, 3(26):26–27, 1965.

[48] Caglar Koylu and Diansheng Guo. Design and evaluation of line symbolizations for origin–destination flow maps. *Information Visualization*, 16(4):309–331, 2017.

[49] Ilya Boyandin, Enrico Bertini, Peter Bak, and Denis Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. In *Computer Graphics Forum*, volume 30, pages 971–980. Wiley Online Library, 2011.

[50] Graphical Perception. Theory, experimentation, and application to the development of graphical methods william s. cleveland; robert mcgill. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[51] Stephen M Kosslyn and Stephen Michael Kosslyn. *Graph design for the eye and mind*. OUP USA, 2006.

[52] Stephen Few and Perceptual Edge. Save the pies for dessert. *Visual Business Intelligence Newsletter*, pages 1–14, 2007.

[53] Mapbox. Make a heatmap with mapbox gl js. `https://docs.mapbox.com/help/tutorials/make-a-heatmap-with-mapbox-gl-js/`. [Accessed: 20-05-2020].

[54] Rostislav Netek, Tomas Pour, and Renata Slezakova. Implementation of Heat Maps in Geographical Information System - Exploratory Study on Traffic Accident Data. *Open Geosciences*, 10(1):367–384, 2018.

[55] Adam Light and Patrick J Bartlein. The end of the rainbow? color schemes for improved data graphics. *Eos, Transactions American Geophysical Union*, 85(40):385–391, 2004.

[56] David Borland and Russell M Taylor Ii. Rainbow color map (still) considered harmful. *IEEE computer graphics and applications*, 27(2):14–17, 2007.

[57] Kenneth Moreland. Why we use bad color maps and what you can do about it. *Human Vision and Electronic Imaging 2016, HVEI 2016*, pages 262–267, 2016.