Evaluating Data Fragmentation across Event-based Social Networks

MYEONG LEE, George Mason University, USA
JULIA H.P. HSU, George Mason University, USA

The rapid growth of online social network platforms enabled many studies to leverage social media data for community-based problem solving. Because many data-intensive studies tend to rely on one information source, using a big social media dataset for predicting and understanding community characteristics could lead to biased interpretations. In other words, overlooking the fragmented nature of data across multiple platforms might result in fairness and accountability issues in understanding the roles of algorithm-powered systems in local communities. To illuminate the importance of considering multiple data sources in community-based data analytics, we disambiguate local event data from three different event-based social network platforms (EBSNs) for three major U.S. cities over 20 months. Through machine learning-based disambiguation of local events, we provide baseline characteristics of local event data from EBSNs. Based on the descriptive analyses of EBSN data in different socio-temporal contexts, this paper discusses the data fragmentation issue across EBSN platforms as an important factor in the fairness and accountability of community-based data analytics.

CCS Concepts: • **Human-centered computing** → **Collaborative and social computing**; • **Computing methodologies** → **Cross-validation**; **Machine learning approaches**.

Additional Key Words and Phrases: Local Events, Data Disambiguation, Data Fragmentation, Machine Learning, Event-based Social Networks

ACM Reference Format:
Myeong Lee and Julia H.P. Hsu. 2018. Evaluating Data Fragmentation across Event-based Social Networks. In . ACM, New York, NY, USA, 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Data collected from online social network platforms provides new opportunities for researchers and practitioners to understand human and organizational behaviors with increase statistical power. Among them, data-driven approaches to solving local community issues often analyze a big dataset from a single online data source such as either Twitter, Facebook, or Meetup [3, 14]. While the volume of data from a single data source is large enough for providing meaningful analysis results, their purposes could be different from one another.

One the one hand, using data from a single source could be reasonable when the goal of the analysis is to develop novel methods and improve the performance of a certain computational task. For example, data-driven research on event recommender systems focuses on improving the performance of the recommendation algorithm within the target system rather than its community-level impact [21]. In this case, maintaining the data quality consistent by focusing on the target data source would be a reasonable option for assessing the performance of the suggested method. On the other hand, community-based applications also make use of geo-tagged data for predicting, understanding, and identifying community dynamics. For example, Foursquare data was used to identify the dynamically-changing neighborhood boundaries by modeling the social structure of places based on people's check-in patterns [5]. Geo-tagged Flickr data

Manuscript submitted to ACM

was collected in another study to model cultural capital and see whether it is associated with economic development in cities [11].

While single-source studies provide novel approaches to understanding or predicting community dynamics based on a large-scale social media data, they often do not faithfully represent a diverse community because of biases in implicit samples comprised of the platform user population [27]. In other words, using one or two social media platforms for data collection would result in using only part of the available data in the local community. Oftentimes, such data might be biased in their users and contents when used for understanding their communities. For example, any one of the local event information sources may contain up to 20% of the entire information available in the community [18].

This suggests that using data from more than one data source would be a reasonable practice for data-driven, community-based problem-solving for (1) reducing platform-specific sampling bias and (2) increasing the statistical power. While issues of data quality and comprehensiveness exist in any application of data science, in community-oriented application, they take on a greater importance because of the possibility for these applications and studies to unintentionally magnify inequality and existing biases [15]. Hence, community-based researchers have an obligation to be particular vigilant about the problem of data in community-oriented application and take special care to take steps to improve and ensure that the analyses are based on the highest quality data possible.

However, there are a few challenges in taking this practice into account. Data formats and available information could be inconsistent across different sources, which makes the data aggregation burdensome. Also, unique norms and affordances of different platforms have affected the data quality and inconsistency. While data disambiguation techniques in some application domains have been studied extensively, data disambiguation practices in community-based problem-solving are under-developed. For example, a recent work on event recommender systems disambiguated local event data by using naïve text matching for event titles, and assumed that all duplicate events were detected in the data cleaning pipeline [16]. While the performance of data disambiguation was not a main focus of this study, the unknown performance of the text matching for detecting duplicate data makes it hard to assess the final results.

As the first step to shed light on the importance of such data practices, this paper aims to explore the characteristics of community-driven data across different platforms. We disambiguate 20-months local event datasets, one of the widely-used community data, from three different event-based social networks (EBSNs) to showcase what the web-based community data looks like across different platforms. By training machine learning (ML) models in different spatio-temporal contexts, we first provide a benchmark of data disambiguation techniques. Based on the data disambiguation results, we show how much event data is fragmented across platforms in different cities and times. Finally, we discuss future directions on research practices that can help increase the accountability of community-based data analytic work. As the benchmarks and the data disambiguation results show varying performances in different contexts, community-based data science work can benefit from this study by considering data characteristics that pertain to disambiguation techniques, spatio-temporal contexts, and platforms.

2 DATA DISAMBIGUATION AND COMMUNITY-BASED RESEARCH

There is a large body of research on data disambiguation [6, 7, 13, 20, 28]. Named entity disambiguation aims to identify people or entities with same names that are found in various platforms such as Wikipedia, ResearchGate, Twitter, and closed domain knowledge bases (e.g., biomedicine, enterprise) [1, 2, 6, 7, 13, 17]. Techniques used to disambiguate the data include supervised learning by modeling the task as a 2-bin classification problem [2, 10], unsupervised learning to cluster all the data that refer to the same entity [12, 17, 22] and graph-based models to disambiguate the data using

the network features [29]. These studies make use of the content information and biographical features as the input to the disambiguation models.

However, only a few of them used spatio-temporal features in the data disambiguation processes [20], because the contexts of the disambiguation work have rarely focused on community-based data. Although [8] built a convolutional neural tensor network to extract latent features from raw location data (i.e., longitude and latitude) for entities in EBSNs, for example, the model were used only for classifying group and user entities. Rather, community-based research has largely focused on community dynamics themselves, assuming the data from a single data source reasonably represents the dynamics. It is partly because the emergence of location-based social media such as Foursquare invoked the increased amount of geo-tagged data on the internet, which has allowed researchers to study and predict community characteristics at scale [9, 23]. For example, natural language processing and machine learning methods were developed to predict economic development using geo-tagged texts from Wikipedia articles [24]. [5] introduced a clustering-based method to understand city's dynamic based on Foursquare data. Similarly, unsupervised machine learning techniques were developed to analyze city logistics using Twitter data [26]. [19] used geo-tagged Twitter data to estimate local commuting patterns. While these methods are novel, they used data from a single data source as with many other community-based research, suggesting the need for further exploring how those community-based data looks when it comes to multiple data sources.

Overall, the facts that (1) disambiguation techniques have been developed largely outside of the community-based data contexts and (2) community-based research have largely focused on the community dynamics and models rather than the sources of data show the gaps between the literature: the distribution of community-based data across different platforms and their disambiguation performances still need further exploration. Although the FAccT community already acknowledges the importance of this kind of issue and has developed the meaning of data representativeness extensively (e.g., [4]), the fragmentation of community-based data in problem-solving practices is still at its infant stage, making it difficult to further discussing nuanced FAccT issues. This motivate us to provide baseline benchmarks at the data level first, as a means to provide a basis for researchers who focus on the nuances and complications embedded in the community-based data.

To assess the effectiveness of existing disambiguation techniques on community data and to understand the characteristics of community-based data, we ask the following questions:

- RQ1: How does the baseline disambiguation performances look?
- RQ2: How does the disambiguation performance changes when training the models across different times, spaces, and data sources?
- RQ3: How much is the local event data fragmented across different sources and over time?

While RQ1 and RQ2 are for assessing the performance of ML-based disambiguation techniques, RQ3 is for providing the results of the data disambiguation work, which scales up the previous work on the event data fragmentation work [18]. Based on the answers to these questions, we discuss the implications for accountability issues in community-based research.

3 APPROACH

The purpose of event data disambiguation is to identify whether two or more local events identified from different data sources are physically the same event or not. After being able to detect physically same events from different data sources, it becomes possible to understand the distribution of community-based data.

Table 1. The volumes of events in target cities and platforms.

	Washington D.C.	New York	Austin
Meetup	102,567	166,536	51,432
Eventful	99,701	241,256	30,424
Yelp	2,452	7,301	1,733

3.1 Data Collection

Local event data was collected from January 2017 through August 2018 (20 months) for three U.S. cities: Washington D.C., New York City (NY), and Austin (TX). These cities are selected based on the geographical contexts that present variability in the locations, the enough volumes of available event data in multiple platforms, and the socio-cultural variability. The data sources that we targeted are Meetup.com, Eventful.com, and Yelp.com. Each website provides Application Program Interfaces (APIs) from which researchers and developers can collect various kinds of data available on the platform. Custom Python and PHP scripts were used to collect this data over time. The number of events per city is presented in Table 1.

3.2 Data Pre-processing

While each data source provides common attributes such as start time, event title, event description, venue address, and ZIP code, there are some inconsistencies in attributes as well. For example, unlike Eventful and Yelp that provide start time and end time, Meetup provides start time and duration to compute the end time of event. If duration is not specified in a Meetup record, the API documentation states that the default duration of an event is automatically set to three hours. Yelp and Eventful sometimes do not include end time as well. In this case, we set the end time to the end of the day. In addition to the start/end time generation, UTC offset is also taken into account to make the times precise. Because event data is from cities in different time zones, the offset information from the data was used to adjust all the times to local times.

Event titles and descriptions were also processed. Because event titles and descriptions from multiple sources are in different formats (e.g., the inclusion of HTML tags, special characters, and varying text lengths), it was necessary to process them so to make the data consistent to some degree. For example, Yelp's event description is automatically cut off after certain amount of text length, which creates inconsistency with other event datasets. To minimize the inconsistency, all the HTML tags and special characters were removed from event titles and descriptions. Stop words were also removed from event descriptions to reduce the noise of textual data. Also, all the words in these fields were changed to lower case and stemmed.

3.3 Feature Engineering

Because there are spatial, temporal, and textual information that could inform the similarity between two different events, we generated pair-wise features for all the pairs of events from different sources. Based on these features, we designed the data disambiguation problem as a 2-bin classification problem (i.e., match or non-match). If two events are physically the same event, it is "match" and, if not, "non-match." We extracted various features that consisted of semantic, temporal, and physical similarity indicators to enrich the limited information obtained from different sources. The final list of the features is listed in Table 2.

Table 2. Pair-wise features generated for machine learning models.

Feature	Description
Start time difference	The start time difference between two events in hours.
Time overlap hours	The overlapping hours between the two events' time periods.
Time overlap rate	The extent to which two events' time periods overlap given the total time period of the two events. $$overlap\_rate = \frac{t_{overlap}}{t_{max} - t_{min}}$$
Period with hours	The event duration in hours.
Name similarity	The Jaccard similarity between two events' names.
Description similarity	The Jaccard similarity between two events' description.
Physical distance	Physical distance between two events' locations (km).
Geocoding type	The lowest resolution of geocoding type between two events.

3.3.1 Temporal feature extraction. The start times in the events data are relatively accurate across different datasets, compared to the end times. Although the availability of events' end times are often inconsistent and unpredictable, start time is one of the powerful predictors in identifying whether two different events are physically the same event or not. Time overlap hours and overlap rates provide other useful information, especially for records that have end time information.

3.3.2 Semantic feature extraction. Jaccard similarity is used for measuring the similarity based on event titles and descriptions because it is known that Jaccard similarity outperforms other metrics in sparse document clustering tasks [25]. When event descriptions are written in non-English languages (e.g., French speaking meetup), which can be detected by using the *googletrans* library in Python, we translated them into English using the translation library.[1] When event descriptions are empty in one of the event pair due to the removal of stop words or URLs, we statistically imputed this feature using the average Jaccard similarity score of all the other pairs in that city.

3.3.3 Geospatial feature extraction. The physical distance between two event locations is also a useful indicator to predict match/non-match. However, a complication in the geo-coordinates of the Meetup datasets is that some Meetup records do not contain location information at all or partial addresses (e.g., no information about longitude/latitude but only the physical address of their event locations available). In the meantime, the data quality of location information in the Eventful and Yelp data is better than that of Meetup due to the nature of data curation. Through qualitative examinations, we found that Yelp data has the highest quality in location information for events. There is no inconsistency found between geo-coordinates and physical addresses in random samples of Yelp events. Although Eventful data had some issues in the location data, similar to Meetup's, Eventful data provides a useful attribute called *geocoding type*, which indicates whether the geo-coordinates of the location is precise or not.

In the Eventful dataset, each event record is tagged with one of the geocoding types: place-level, zipcode-level, and city-level. Place-level geocoding provides the precise geo-coordinates of an event; zipcode-level geocoding provides the accuracy at the zip code level; and city-level geocoding indicates the city-level accuracy (i.e., not accurate). As a result, it is possible to identify whether the event location information is precise or not by considering the geocoding classifications in Eventful dataset. Moreover, the geocoding type attribute also makes it possible to complement the

[1] https://pypi.org/project/googletrans/

errors in Meetup's geo-coordinates, because there is no accuracy indicator for the location data and the number of errors is not negligible. In this regard, Meetup's location data was examined and improved through a series of data processing. Particularly, each record of the Meetup location data was examined automatically using scripts and, if necessary, was tagged with one of the geocoding schemes following those of Eventful. The automated feature extraction process of the geo-coordinates and geocoding types is as follows.

Step 1: Making use of "venue repinned": We observed that if the *venue_repinned* attribute of a Meetup event is TRUE, meaning the event organizer manually picked a particular location on the map for the event, its geo-coordinates is accurate; thus, the records with TRUE for the *venue_repinned* attribute were tagged "place-level" for the *geocoding_type* attribute.

Step 2: Geocoding for unregistered venues: Then, the event venues were examined to see if there were any anomalies. When the venue ID of a Meetup event *venue_id* is not available, it means this event does not make use of the registered venue data from Meetup's venue database; rather, the venue information is manually entered by the organizer in the *how_to_ find_us* field or not available at all. If both *venue_id* and *how_to_ find_us* attributes were not available for an event, this event was tagged with "city-level" and the center of the city was used for the geo-coordinates, because venue information was not available at all.

If *venue_id* is not available but the *how_to_find_us* field exists, it is possible that this attribute contains one of the following: physical address, geo-coordinates, a URL of further information, the email address of the organizer, or the phone number of the organizer. Regular expressions were used to detect the form of the information in this attribute, and location information was extracted from the *how_to_ find_us* field when relevant. For example, if the pattern matched longitude/latitude, the value was extracted and copied to the geo-coordinates attributes; then, this event was tagged "place-level" for the *geocoding_type* attribute. If the pattern matched email address or URL, these records were ignored from the processing.

If any of the pre-defined patterns were not relevant to the *how_to_find_us* field, the text was assumed as a physical address and geocoded using the Google Maps Geocoding API. When the Geocoding API returned an error or null value, the event record was tagged with "city-level" for the *geocoding_type* attribute. Also, when the geo-coordinates returned from the API was more than 25 miles from the city center, the record was tagged with "city-level," because it was highly possible that this geocoding value was wrong (mostly because the text in the *how_to_find_us* field does not correctly present the physical address).[2]

Step 3: Geocoding for registered online venues and TBDs: It is possible that registered venues on Meetup are actually not physical locations but online addresses, especially for the events that happen through web interfaces. Also, some venue addresses show "TBD" and provide a zip code- or city-level location only. Regular expressions were used to detect online events. These events are tagged with "city-level" for *geocoding_type*. If place-level location data is not available or tagged "TBD," other location-related fields such as city and zipcode attributes were examined to check the granularity of the location information. If venue address did not exist, the finest resolution of the location information was geocoded using Google Geocoding API and tagged accordingly. For example, if a record has zipcode but does not have a physical address, it is geocoded using the center of the zip code region and tagged with "zipcode-level" for the *geocoding_type* attribute.

Step 4: Geocoding for registered venues with no geo-coordinates: There are registered venues that contain physical addresses but without geo-coordinates. In this case, the physical address was reverse geocoded using the

[2]When we collected Meetup data, all the data that is located within a 25-mile radius from the city center was the target. This justifies 25 miles as the threshold to detect geocoding errors.

Google Geocoding API. Sometimes, a physical address does not provide the street-level address; in such cases, similar to Step 3, it was geocoded based on the finest granularity of location available in the data. For a sanity check, the distance between the geo-coordinates from API and the center of the city was calculated for each record. If this distance was longer than 25 miles, the geocoding result was assumed to be wrong, maybe due to incomplete address information, and tagged "city-level." Depending on the precision of the geocoding, each record was tagged with one of the three geocoding classifications.

3.4 Ground-truth Generation

After generating and engineering features, pairs with no overlap in their time periods (i.e., time_overlap_hours = 0) were removed from the pair dataset because there was no possibility that they were the same event. Then, the stratified sampling method was used to sample 1,800 pairs of events from the dataset randomly while balancing the number of events based on distance, similarity, and topics. To generate the ground-truth data, we manually compared each pair of events in the sample dataset by visiting the corresponding event's web page to check whether each pair was physically the same event.

Initially, the match/non-match was coded in two ways: the conservative coding and flexible codings. The conservative coding assesses the match/non-match of two events based on their physical location and organizer; in the meantime, the flexible coding assesses only their relations to a physical event (e.g., Science March by two different groups are regarded as "match" in flexible coding, but "non-match" in conservative coding).

Through qualitative examinations and ML performance tests, the flexible codings were decided as the main object variable, because it makes more sense to focus on physical events rather than people who create online events and, on a more practical level, flexible codings yield a better consistency in the ML tests.

3.5 Machine Learning Benchmarks

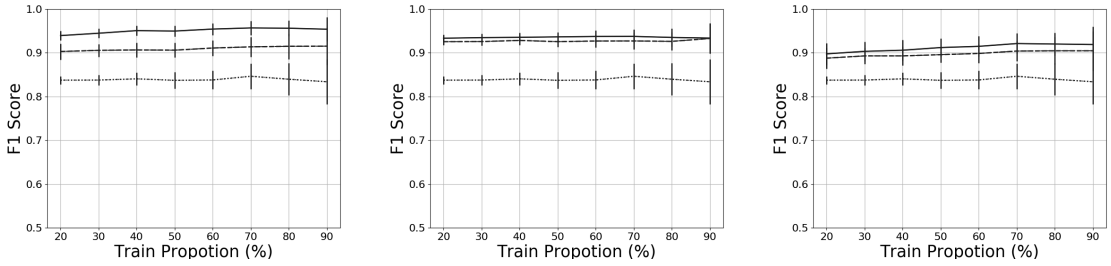
3.5.1 ML Models. With the manual coding results as the ground-truth data, we modeled the disambiguation of event data as the binary classification task. We used Decision Tree, Random Forests, and Support Vector Machine (SVM) to identify "match/non-match" of the event pairs. Features that consist of semantic, temporal, and physical similarity indicators were used to train the models. Grid search was used for tuning model parameters. Specifically, the minimum number of samples required to split an internal node for tree-based models was 4, and the Radial basis function was used for the kernel function of SVM. We compared the performance of our models with naive textual matching and other models that involved name similarity comparison. The performances were evaluated by precision, recall, and F1 scores. Moreover, the overall performance was evaluated by 100 independent validations, in which random seeds were generated in each iteration.

3.5.2 Baseline Performance Measures. We compared the ML-based methods to the following baseline methods: (1) random guess: generate predictions uniformly at random. (2) text matching of event name: generate predictions by pure text matching of event name (used in [16]. Specifically, for each pair of event, if one of the event name is a subset of the other, the prediction is match. (3) event name Jaccard similarity: use Jaccard similarity of event name as feature to train ML models and generate predictions. To ensure the robustness of each model, the performance was measured based on the average F1 score of 100 independent predictions. In addition, different combinations of training data were generated based on the time, locations, and datasets to validate that our ML models provide stable performances across time, locations, and datasets.

4 RESULTS

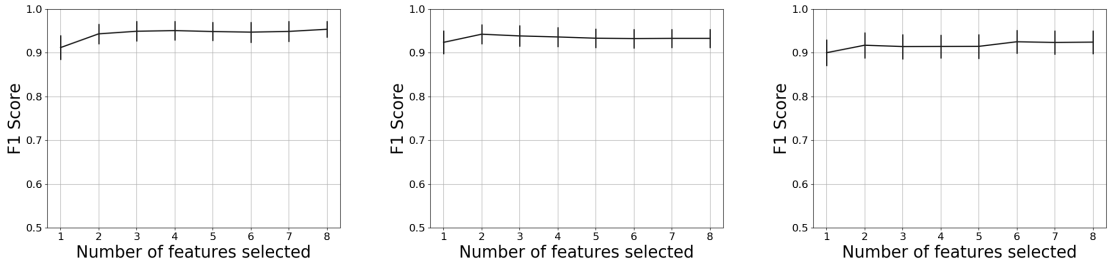
4.1 RQ1: General ML Performance

The performances of ML models are measured using F1 scores based on the average of 100 test results. For each test of the models, the proportions of the training set range from 20% to 90% to show the robustness of the model. Figure 1 shows the average F1 scores of ML models within the 1800 pairs of events. Table 3 shows the average precision, recall and F1 score of each method. Our ML models that use temporal, spacial and textual features to disambiguate event data have better performance than other baseline methods do. Specifically, Random Forests presents the best performance in its average F1 score, 0.956, when 80% of the data are used for training. SVM and Decision Tree show good performances as well, but F1 scores are 0.937 and 0.917, respectively. Figure 2 shows the average F1 scores versus number of features used over 100 independent predictions. Random Forests has the highest average F1 score when all of the features are included. In addition to F1 score, recall is also important when the number of matches is very small. As shown in Table 3, Random Forests that is trained with various features has the highest average recall value of 0.978, while pure textual matching of event title only has the average recall value of 0.753. This suggests that including temporal and spacial features can help find true matches and increase the recall in the given dataset.



(a) Random Forests. (b) SVM. (c) Decision Tree.

Fig. 1. F1 scores of different machine learning models. Each score is the average of 100 independent predictions. Feature-based ML, name similarity-based ML, and name textual matching are represented by solid line, dashed line, and dotted line, respectively



(a) Random Forests (b) SVM (c) Decision Tree

Fig. 2. F1 scores versus number of features. Each point is the average F1 score of 100 independent predictions. The order of features is name similarity, description similarity,time overlap rate, time overlap hours, period hours,start time difference, geocoding type, and physical distance.

Table 3. Average precisions, recalls and F1 scores of 100 independent predictions using 80% of data for training

	Precision	Recall	F1 score
Random guess	0.133	0.495	0.209
Textual matching of event name	0.940	0.753	0.835
RF (name similarity)	0.902	0.929	0.914
SVM (name similarity)	0.900	0.956	0.927
DT (name similarity)	0.901	0.902	0.900
RF	**0.935**	**0.978**	**0.956**
SVM	0.907	0.970	0.937
DT	0.930	0.906	0.917

4.2 RQ2: Model Robustness across Times, Cities, and Data Sources

4.2.1 Cross-time Validation. To show the stability of models across time, we divide the data into training and testing sets by months. We use January 2017's and July 2017's data to train the ML models, respectively, and evaluate the performances of the models with the remaining data. Figure 3 shows that the ML-based models still have F1 scores higher than 0.90 when the training set only includes data from a specific month. The average precision, recall and F1 score of each model are shown in Table 4.

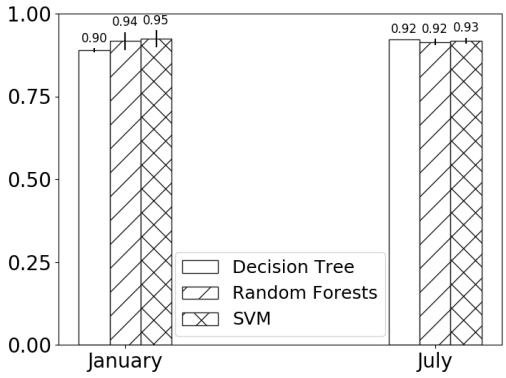


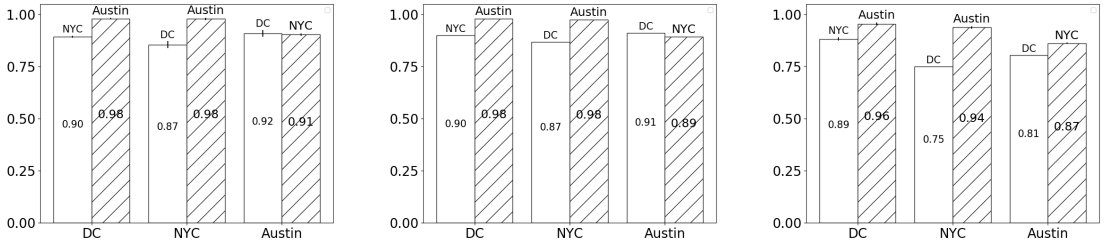
Fig. 3. F1 scores of cross-time validation

4.2.2 Cross-city Validation. Similar to the cross-time validation, the data is divided into training and testing sets based on the locations for cross-city validation. We use data from Washington D.C., New York and Austin to train the ML models respectively, and evaluate the performance of models with the remaining data. The performance of each ML model that is trained with different cities' data is shown in Figure 4. Table 5 shows that including various features could help improve the performance of Random Forests and SVM.

4.2.3 Cross-dataset Validation. Finally, event-pair data from Eventful & Meetup, Eventful & Yelp and Meetup & Yelp are used to train ML models, respectively, for cross-dataset validation. The performances of ML models that use different datasets for training are shown in Figure 5. Unlike other results of validations, Random Forests and Decision Tree do

Table 4. Average precisions, recalls and F1 scores of cross-time validation

	Precision	Recall	F1 score
Random guess	0.144	0.507	0.224
Textual matching of event name	0.534	0.434	0.479
RF (name similarity)	0.891	0.955	0.922
SVM (name similarity)	0.897	0.953	0.924
DT (name similarity)	0.897	0.862	0.878
RF	0.901	0.951	0.929
SVM	**0.899**	**0.969**	**0.932**
DT	0.891	0.930	0.910



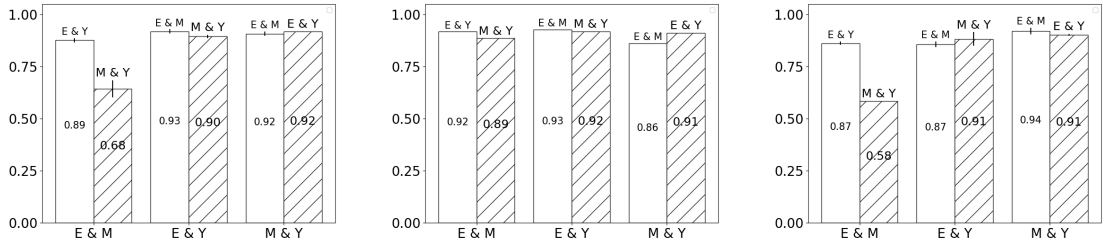
(a) Random Forests (b) SVM (c) Decision Tree

Fig. 4. F1 scores of cross-city validation

Table 5. Average precisions, recalls and F1 scores of cross-city validation

	Precision	Recall	F1 score
Random guess	0.132	0.521	0.210
Textual matching of event name	0.941	0.756	0.838
RF (name similarity)	0.887	0.915	0.899
SVM (name similarity)	0.900	0.957	0.927
DT (name similarity)	0.888	0.914	0.899
RF	**0.912**	**0.948**	**0.928**
SVM	**0.909**	**0.951**	**0.928**
DT	0.908	0.868	0.885

not perform well when training set only includes Eventful & Meetup event-pair data and excludes Yelp data. Excluding Yelp data in the training set might lead to lower performance, because there is a limitation of Yelp's API, which do not provide complete access to event description of each event. Table 6 shows the average precision, recall, and F1 score of cross-dataset validation, where models that are trained with name similarity feature have higher average values of F1 score.



(a) Random Forests (b) SVM (c) Decision Tree

Fig. 5. F1 scores of cross-dataset validation

Table 6. Average precisions, recalls and F1 scores of cross-dataset validation

	Precision	Recall	F1 score
Random guess	0.132	0.499	0.208
Textual matching of event name	0.944	0.750	0.834
RF (name similarity)	0.892	0.890	0.887
SVM (name similarity)	**0.906**	**0.946**	**0.924**
DT (name similarity)	0.895	0.886	0.886
RF	0.919	0.828	0.864
SVM	0.912	0.903	0.907
DT	0.907	0.784	0.835

4.3 RQ3: Data fragmentation rates across different sources and time

We use the following formula to show the duplication rate in a city. Based on the number of organized events in each information source (i.e., $N_{meetup}, N_{yelp}, N_{eventful}$), their pair-wise duplicates (i.e., $N_{m\cap y}, N_{m\cap e} and N_{e\cap y}$) and the number of overlapping events across all the three sources (i.e., $N_{m\cap y\cap e}$), the duplication rate can be calculated as follows:

$$duplication\ rate = \frac{D}{N}, \tag{1}$$

where

$$D = N_{m\cap y} + N_{m\cap e} + N_{e\cap y} - 2 \times N_{m\cap y\cap e} \tag{2}$$

and

$$N = N_{meetup} + N_{yelp} + N_{eventful} - N_{m\cap y} - N_{m\cap e} - N_{e\cap y} + N_{m\cap y\cap e} \tag{3}$$

Figure 6 shows that the duplication rates of local event data keep increasing over time based on one data source's increasing data curation practices (the volume of Eventful data has been increasing continuously during the data collection period). Even though, the duplication rates are very low at around 2% if we assume these three sources cover a enough breadth of the available local events. Given the fact that the three data sources that we use reflect only part of the entire local event data available in the cities [18], there needs to be further studies that examine more data sources. The results suggest that local event data is highly fragmented across different data sources, and future community-based

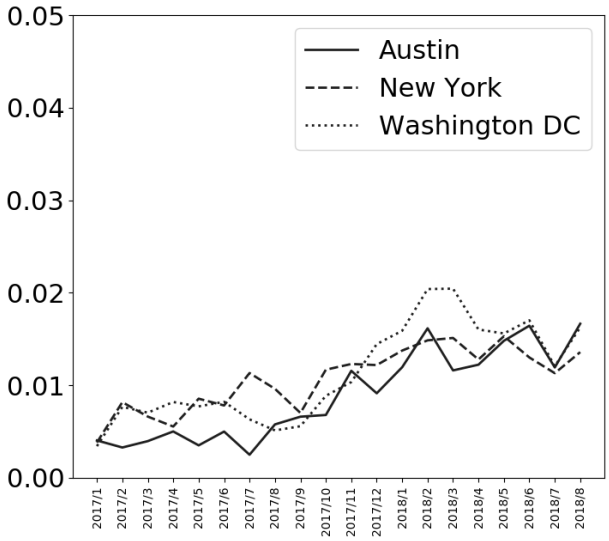


Fig. 6. Duplication rate

applications will need to take this issue seriously to ensure the data accountability in community-based data science works.

5 DISCUSSION AND CONCLUSION

In recent data science applications that focus on community dynamics and data disambiguation, it was unclear (1) whether the data represents community dynamics and (2) what the data characteristics are when it comes to multiple data sources. As the results suggest, data disambiguation performances vary depending on how to divide train and test sets across times, cities, and platforms. Particularly, data pre-processing with an insufficient understanding of the data fragmentation issues in community-based data science work could lead to a serious accountability issue from a scientific rigor and ethics perspective. These inferences and results provide useful implications for community-based data.

First, it is obvious that focusing on one data source could target only a small portion of available data in the community, as shown in the results of RQ3. Community-based data science work needs to take the phenomenon of data fragmentation into account seriously as each data source often has a high level of platform-specific biases. Although it would be difficult to eliminate sampling bias entirely from data collected from online platforms, using multiple data sources will help alleviate these potential biases.

Second, the stability of data disambiguation techniques differs by training sets for ML models. Depending on the data quality and inconsistency, the stability of performance varies. When using a trainset from two data sources to disambiguate data from another two platforms, the prediction power was unstable. Particularly, Eventful data as part of the training set leads to decreased performance and increased instability. One possible reason is that the training set does not include data from Yelp, where event descriptions are not complete due to the limitation of Yelp API. Because

of this, we checked the feature importance and visualized the decision tree using Scikit-Learn library [3], and found that the tree split the data based on the description similarity at the first level while using Eventful and Meetup pairs for training. When training ML models for event data disambiguation, it is recommended to combine both high and low quality datasets, if unavoidable. Also, data scientists can conduct sensitivity tests across times, spaces, and platforms for train and test sets to ensure the robustness of the data disambiguation performance. Finally, generating pair-wise features that consisted of semantic, temporal, and physical similarity indicators is necessary to disambiguate event data from different EBSN platforms due to the inconsistencies of data format and available information across different sources.

Third, the performances across geospatial and temporal ranges was relatively stable in the target EBSNs.

Surprisingly, Table 4 shows that in the cross-city validation, textual matching of event name has lower recall value compare with random guess. Naïve textual matching of event name does not provide good results probably because event names are usually short, which might increase the number of false positive predictions and thus a lower recall value. Especially for those events that are related to festivals or holiday seasons tend to have similar names and time periods. For instance, there were many New Year's Eve parties in January, while most of them were actually different events but with similar event names. As a result, data scientists need to be cautious when they aim to disambiguate events using data within a limited time. We have shown that our models which include various kinds of features are robust across times and are able to disambiguate events using data within a limited time for training.

Also, it was possible to disambiguate events in other cities without including large amount of data from different cities for training. Although Table 5 shows that SVM (training with event name similarity) provides a good performance as well, using event name as the only feature might increase the number of false positive predictions as discussed above as well as the risk of under-fitting. Therefore, including various features could help improve the robustness of the model.

Beyond data scientists, this is also a call for research on data fragmentation issues for FAccT scholars. Although FAccT research has uncovered important dimensions in the FAccT issues, the data fragmentation issues in community-based data have been relatively under-studied. The contexts of community-based data are unique from a FAccT perspective, because people's affordances of platforms, cultural practices, and temporal patterns are different even on the same platform. This presents a high level of nuances and complexities embedded in the data as well as in the problem itself. We believe this study is the initial effort to open discussions about this problem space. We call for researchers' attention to the data selection and disambiguation processes, as well as a broader spectrum of data fragmentation issues, when designing strategies for community-based problem solving.

REFERENCES

[1] Mehmet Ali Abdulhayoglu and Bart Thijs. 2017. Use of ResearchGate and Google CSE for author name disambiguation. *Scientometrics* 111, 3 (2017), 1965–1985.

[2] Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. (2006).

[3] Jonathan Chang and Eric Sun. 2011. Location 3: How users share and respond to location-based data on social networking sites. In *Proceedings of the fifth international AAAI conference on weblogs and social media*. AAAI Press, 74–80.

[4] Kyla Chasalow and Karen Levy. 2021. Representativeness in Statistics, Politics, and Machine Learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 77–89.

[5] Justin Cranshaw, Raz Schwartz, Jason I Hong, and Norman Sadeh. 2012. The livehoods project: Utilizing social media to understand the dynamics of a city. *International AAAI Conference on Web and Social Media* (2012).

[3] https://scikit-learn.org

[6] Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*. 708–716.

[7] Alexandre Davis, Adriano Veloso, Altigran Soares, Alberto Laender, and Wagner Meira Jr. 2012. Named entity disambiguation in streaming data. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 815–824.

[8] Daizong Ding, Mi Zhang, Xudong Pan, Duocai Wu, and Pearl Pu. 2018. Geographical feature extraction for entities in location-based social networks. In *Proceedings of the 2018 World Wide Web Conference*. 833–842.

[9] Simona Giglio, Francesca Bertacchini, Eleonora Bilotta, and Pietro Pantano. 2019. Using social media to identify tourism attractiveness in six Italian cities. *Tourism management* 72 (2019), 306–312.

[10] Hui Han, Lee Giles, Hongyuan Zha, Cheng Li, and Kostas Tsioutsiouliklis. 2004. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004.* IEEE, 296–305.

[11] Desislava Hristova, Luca M Aiello, and Daniele Quercia. 2018. The new urban success: How culture pays. *Frontiers in Physics* 6 (2018), 27.

[12] Jian Huang, Seyda Ertekin, and C Lee Giles. 2006. Efficient name disambiguation for large-scale databases. In *European conference on principles of data mining and knowledge discovery*. Springer, 536–544.

[13] Yiting Ju, Benjamin Adams, Krzysztof Janowicz, Yingjie Hu, Bo Yan, and Grant McKenzie. 2016. Things and strings: improving place name disambiguation from short texts by combining entity co-occurrence with topic modeling. In *European Knowledge Acquisition Workshop*. Springer, 353–367.

[14] Juntao Lai, Guy Lansley, James Haworth, and Tao Cheng. 2020. A name-led approach to profile urban places based on geotagged Twitter data. *Transactions in GIS* (2020).

[15] Karen Layne and Jungwoo Lee. 2001. Developing fully functional E-government: A four stage model. *Government information quarterly* 18, 2 (2001), 122–136.

[16] Cheng Li, Michael Bendersky, Vijay Garg, and Sujith Ravi. 2017. Related event discovery. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 355–364.

[17] Yang Li, Shulong Tan, Huan Sun, Jiawei Han, Dan Roth, and Xifeng Yan. 2016. Entity disambiguation with linkless knowledge bases. In *Proceedings of the 25th international conference on world wide web*. 1261–1270.

[18] Claudia López, Brian Butler, and Peter Brusilovsky. 2014. Does anything ever happen around here? Assessing the online information landscape for local events. *Journal of Urban Technology* 21, 4 (2014), 95–123.

[19] Graham McNeill, Jonathan Bright, and Scott A Hale. 2017. Estimating local commuting patterns from geolocated Twitter data. *EPJ Data Science* 6, 1 (2017), 24.

[20] Greg Morrison, Massimo Riccaboni, and Fabio Pammolli. 2017. Disambiguation of patent inventors and assignees using high-resolution geolocation data. *Scientific data* 4 (2017), 170064.

[21] Zhi Qiao, Peng Zhang, Yanan Cao, Chuan Zhou, Li Guo, and Binxing Fang. 2014. Combining heterogenous social and geographical information for event recommendation. In *Twenty-Eighth AAAI conference on artificial intelligence*.

[22] Luís Sarmento, Alexander Kehlenbeck, Eugénio Oliveira, and Lyle Ungar. 2009. An approach to web-scale named-entity disambiguation. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 689–703.

[23] Lisa Schweitzer. 2014. Planning and social media: a case study of public transit and stigma on Twitter. *Journal of the American Planning Association* 80, 3 (2014), 218–238.

[24] Evan Sheehan, Chenlin Meng, Matthew Tan, Burak Uzkent, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. 2019. Predicting economic development using geolocated wikipedia articles. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2698–2706.

[25] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. 2000. Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, Vol. 58. 64.

[26] Simon Tamayo, François Combes, and Arthur Gaudron. 2020. Unsupervised machine learning to analyze City Logistics through Twitter. *Transportation Research Procedia* 46 (2020), 220–228.

[27] Zeynep Tufekci. 2014. Big questions for social media big data: Representativeness, validity and other methodological pitfalls. *International AAAI Conference on Web and Social Media* (2014).

[28] Haiwen Wang, Ruijie Wan, Chuan Wen, Shuhao Li, Yuting Jia, Weinan Zhang, and Xinbing Wang. 2020. Author Name Disambiguation on Heterogeneous Information Network with Adversarial Representation Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 238–245.

[29] Baichuan Zhang, Tanay Kumar Saha, and Mohammad Al Hasan. 2014. Name disambiguation from link data in a collaboration graph. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*. IEEE, 81–84.