2015

Extracting City Traffic Events from Social Streams

Pramod Anantharam
Wright State University - Main Campus, anantharam.2@wright.edu

Payam Barnaghi

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

Repository Citation

Anantharam, P., Barnaghi, P., Thirunarayan, K., & Sheth, A. P. (2015). Extracting City Traffic Events from Social Streams. *ACM Transactions on Intelligent Systems and Technology, 6* (4).
https://corescholar.libraries.wright.edu/knoesis/1039

Extracting City Traffic Events from Social Streams

Pramod Anantharam, Wright State University
Payam Barnaghi, University of Surrey
Krishnaprasad Thirunarayan, Wright State University
Amit Sheth, Wright State University

Cities are composed of complex systems with physical, cyber, and social components. Current works on extracting and understanding city events mainly rely on technology enabled infrastructure to observe and record events. In this work, we propose an approach to leverage citizen observations of various city systems and services such as traffic, public transport, water supply, weather, sewage, and public safety as a source of city events. We investigate the feasibility of using such textual streams for extracting city events from annotated text. We formalize the problem of annotating social streams such as microblogs as a sequence labeling problem. We present a novel training data creation process for training sequence labeling models. Our automatic training data creation process utilizes instance level domain knowledge (e.g., locations in a city, possible event terms). We compare this automated annotation process to a state-of-the-art tool that needs manually created training data and show that it has comparable performance in annotation tasks. An aggregation algorithm is then presented for event extraction from annotated text. We carry out a comprehensive evaluation of the event annotation and event extraction on a real-world dataset consisting of event reports and tweets collected over four months from San Francisco Bay Area. The evaluation results are promising and provide insights into the utility of social stream for extracting city events.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms: Design, Algorithms, Experimentation

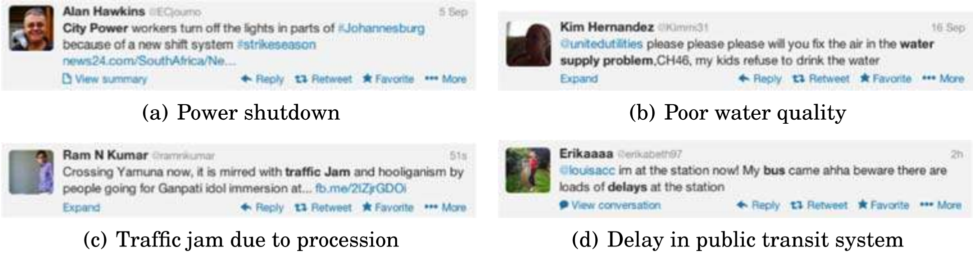
Additional Key Words and Phrases: smart cities, citizen sensing, city events, tweets, event extraction, physical-cyber-social systems

ACM Reference Format:
Pramod Anantharam, Payam Barnaghi, Krishnaprasad Thirunarayan, Amit Sheth, (pre-print copy). *ACM Trans. on Intellig. Syst. & Tech.* V, N, Article A (January YYYY (pre-print copy)), 26 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Cities have been a thriving place for citizens over centuries due to a range of rich socio-economic opportunities offered by them. By 2001 over 285 million people lived in cities of India, which was more than the population of the entire United States then [Pucher et al. 2004]. This trend of citizens moving to cities is creating tremendous pressure on the city infrastructure. Understanding the status and interactions between city systems is crucial to enable smooth functioning of a city. City authorities face numerous challenges in deploying, maintaining, and optimizing operations and interactions between various city departments and services (collectively called infrastructure). They are also pressed for ways to minimize wastage of resources, improve efficiency, and be economically self-sustainable. Understanding city events is of great contemporary interest [Naphade et al. 2011; Lindsay 2010; Kehoe et al. 2011] emphasizing the crucial need for extracting and analyzing city events. Citizen sensing [Sheth 2009; Burke et al. 2006] component that may provide complementary or corroborative information is often ignored in the current state-of-the-art analytics for Smart Cities [Filipponi et al. 2010; Lu et al. 2010]. We show that social data streams harnessed in the context of Smart Cities provide a comprehensive view of events in a city (complementing other modalities such as observations from city authorities).

Twitter (a microblogging platform) has developed into a near real-time source of information spanning heterogeneous topics of varying importance. With over 500 million



Alan Hawkins @ECjourno 5 Sep
City Power workers turn off the lights in parts of #Johannesburg because of a new shift system #strikeseason news24.com/SouthAfrica/Ne...

(a) Power shutdown

Kim Hernandez @Kimm31 16 Sep
@unitedutilities please please please will you fix the air in the **water supply problem**,CH46, my kids refuse to drink the water

(b) Poor water quality

Ram N Kumar @ramnkumar 51s
Crossing Yamuna now, it is mirred with **traffic Jam** and hooliganism by people going for Ganpati idol immersion at... fb.me/2lZjrGDOi

(c) Traffic jam due to procession

Erikaaaa @erikabeth97 2h
@louisacc im at the station now! My **bus** came ahha beware there are loads of **delays** at the station

(d) Delay in public transit system

Fig. 1. Tweets reporting various concerns about a city spanning power supply, water quality, traffic jams, and public transport delays

users world-wide, twitter generates 500 million tweets a day[1]. Increasingly, tweets do provide interesting and vital information such as status of public transport, traffic and environmental conditions, public safety, and general events in a city.

We address the following research questions: How do we extract city infrastructure related events from twitter? How can we exploit event and location knowledge-bases for event extraction? How well can we extract city traffic events? In addressing these questions, we outline the challenges in extracting events related to city infrastructure from informal text, and demonstrate the effectiveness of our approach through a comprehensive evaluation in the context of traffic related events. Specifically, we compare with ground truth provided by 511.org traffic incident reports showing the promise of our approach.

The following are the contributions of our work: (1) We hypothesize and validate the role of citizen sensing in extracting city traffic events. (2) We develop an automatic training data creation process to train the annotation model by utilizing existing event and location knowledge-bases. (3) We design and implement a city event extraction algorithm from annotated textual data. (4) We evaluate our approach concretely by comparing events extracted from social streams and events reported on 511.org using three orthogonal metrics to emphasize their complementary, corroborative, and timely nature.

2. MOTIVATION AND BACKGROUND

We illustrate the role of citizen sensing [Sheth 2009] for understanding city infrastructure related events and present related work on event extraction.

2.1. Motivation

Typically, a city has many departments such as public safety, urban planning, energy and water, environmental, transportation, social programs, and education [Bélissent 2010; 2013]. Some of the services offered by these departments are dynamic, e.g., transportation services and their behavior may vary in response to sporting and music events, accidents, and weather conditions. Timely understanding of the situation is important for city authorities to manage city resources. Figure 1 depicts real-world city events reported directly by citizens in near real-time on twitter. They relate to power outages, poor water quality, a procession, and a delay experienced on public transport system. This information complements sensor data or textual data from conventional sources such as city departments. For example, sensors deployed on a road may report reduced speed of vehicles which can be explained by the procession obstructing traffic.

[1] http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html

In many cities around the world, there is an immense pressure on city infrastructure. Fine-grained sensor data may not be available from such cities due to the lack of extensive instrumentation. Citizen sensing can play a crucial role in filling the void in such environments [Anantharam and Srivastava 2013].

2.2. Related Work

We outline the challenges in extracting city related events from informal text and motivate the need for specialized approaches beyond the current state-of-the-art. Works such as [Mladenić and Moraru 2012] assume the presence of event data sources such as sensor data (e.g., loop detectors) and formal report of events (e.g., eventful[2]) in a city. Such a source of events may not be available in all the cities warranting the need for city event extraction from textual data. Formal events when present can serve as ground truth to validate the event extraction approaches. We organize the related work on event extraction from formal and informal text into two categories: open domain (unknown event types) and closed domain (known event types).

2.2.1. Formal Text. Event extraction can be done on grammatical text such as news documents. Parts of speech information and sentence parse can be exploited in processing the content.
Open Domain: Event identification using a combination of text classification and use of named entities from news articles has been carried out by [Kumaran and Allan 2004]. Similarly, to alleviate information overload in daily news, key entity and significant event extraction is done on news documents in [Liu et al. 2008]. A bipartite graph is induced based on the entities and their associations to documents using mutual reinforcement principle capturing salient entities and the documents with salient entities to rank the news events. Extraction of local events from blog entries has been carried out by [Okamoto and Kikuchi 2009].
Closed Domain: Use of lightweight patterns to extract global crisis events from news text is presented in [Tanev et al. 2008]. A combination of patterns specified manually and learned from data are utilized to determine event specific semantic roles (e.g., date and location, actors, event type). Patterns for event specific roles are then used by event aggregation algorithms. An evaluation of accuracy of event extraction is carried out on a news corpus. Twenty seven out of twenty nine violent events were detected using the approach in [Tanev et al. 2008]. Event extraction in the context of detecting infectious disease outbreak was done by [Grishman et al. 2002]. The event schema consists of date range, geo-location, disease name, organism type and number affected by the disease, and the organism survival information. The event extraction is done by finite-state pattern matching on the tokenized input text. The extracted events are compared against ground truth from ProMed[3] and WHO Infectious Disease Reports[4]. Creation of succinct summaries of events from news sources was carried out by [Naughton et al. 2006]. A hierarchical clustering algorithm to cluster sentences referring to the same event has been presented as a baseline. Sentences in a news article that do not really describe the event are filtered out before clustering as an improvisation over baseline to show improved clustering accuracy in [Naughton et al. 2006].

2.2.2. Informal Text. Event extraction is done on user-generated content with no overt structure and that contains lot of slangs and non-standard abbreviations. Text fragments may not follow any rules of grammar making it hard to process using traditional techniques.

[2] http://eventful.com/
[3] http://www.promedmail.org/
[4] http://www.who.int/topics/infectious_diseases/en/

Open Domain: Event extraction from informal text such as tweets has received increased research attention. Synthesizing subgraphs in a graph of keywords (nodes representing keywords and edges representing co-occurrence statistics) using community detection techniques is studied by [Sayyadi et al. 2009]. Each subgraph formed by a community of keywords can represent an event. Clustering based approach to detect events and adapting it to streaming data is presented in [Aggarwal and Subbian 2012]. This clustering based approach caters to open domain event extraction where there is no prior knowledge on the number of event types. In [Dou et al. 2012], event extraction techniques are organized based on four tasks: new event detection, event tracking, event summarization, and event association. New event detection techniques are used to identify first story of an event. Event tracking captures the evolution of an event. Event summarization involves creating summaries from bursts of messages. Event association uncovers relationships between events leading to domain insights. Open domain extraction of events from informal text is addressed in [Ritter et al. 2012]. This work demonstrates that building a calendar of significant events is feasible using twitter stream using an unsupervised approach to process tweets and extract event types such as sports, concert, protests, politics, TV, and religion. The approach models each entity in terms of a mixture of event types and each event type in terms of a mixture of entities. It requires minimal supervision for labeling the event descriptors but provides a fairly convincing approach to handle noisy, redundant, and informal nature of tweets. The evaluation compares it with a supervised baseline with improvement in F1 score. Using tweets for predicting hit and run crimes has been proposed by [Wang et al. 2012]. A latent topic based model is constructed over semantic role labels [Màrquez et al. 2008] of events from tweets. A generalized linear regression model learns the association between topics and crimes from a training dataset. The Receiver Operating Characteristics (ROC) curve based evaluation compares this approach with a baseline that associates uniform priors to crimes on all days.

Closed Domain: Using twitter streams to estimate the occurrence of events and its intensity using a supervised learning approach has been proposed by [Lampos and Cristianini 2012]. It uses an optimized feature selection approach coupled with regression to estimate the intensity of events based on event markers. An evaluation based on ground truth from rain gauges is used for validation. They also extend the evaluation to identify Influenza Like Illness and compare it with the data from Health Protection Agency[5]. The study concludes the feasibility of using tweets for estimating events and its intensity. A clustering based approach is used by [Becker et al. 2011] for distinguishing tweets related to real-world events from non-events. Temporal (volume changes), social (replies, broadcast), topical (coherence of clusters), and twitter-centric (multi-word hashtags) features are used to train a classifier that performs better than the baseline. It utilizes Term Frequency - Inverse Document Frequency (*tf-idf*) vector of textual content as features of a tweet.

Although there has been a lot of work on event extraction from social media streams, there is very little work on identifying various events which impacts traffic flow in a city. Most of the event extraction techniques presented as related work do not emphasize location of the real-world events except for [Lampos and Cristianini 2012]. We believe this is crucial for extracting city traffic events. Knowing the location of city traffic event, such as traffic jam, is important to take further action. Impact assessment of events provides insights into the magnitude of events and allows city authorities to prioritize resources. Efforts reported so far lack integrated use of event localization and impact assessment. The approach to extract city traffic related events presented

[5] http://www.hpa.org.uk/

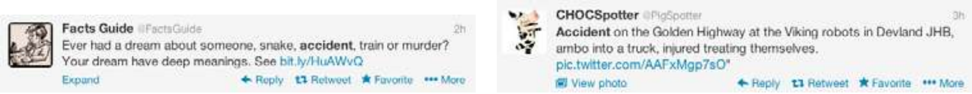
in this paper addresses these shortcomings of the state-of-the-art approaches by providing location and impact assessment over real-world city traffic events.

2.2.3. Challenges and Opportunities. Extracting city related events from informal textual streams is similar in spirit to open domain event extraction from informal text. However, there are additional challenges to be addressed. Events in a city unfold in real-time and social media is shown to be a good source reporting real-world events such as earthquake [Sakaki et al. 2010]. In contrast, relying on conventional news streams to report observations may lead to some delays. City events are usually reported by a few people (limited number of sources) leading to the challenges of biased reporting. Since twitter is also used by citizens to exchange mundane daily activities and events, detecting and filtering the signals related to city infrastructure is a challenging task. Location and time play a crucial role in event extraction by aggregating information that has spatio-temporal distribution and the need for timely, actionable decisions for smooth functioning of city infrastructure. These issues and challenges are the focus of our comprehensive approach to improving the state-of-the-art. We show how events related to city traffic infrastructure can be extracted from twitter stream and processed to derive actionable intelligence. We address the challenge of noisy twitter data by using location and content based filters that utilize knowledge bases such as OSM for location names in a city and 511.org hierarchy of traffic events. We propose an event aggregation algorithm that is capable of deriving location, duration, and impact of an event using techniques to compartmentalize the city into smaller units called grids. While the approach we propose is generic enough to deal with any city related event, we will constrain our evaluation to the domain of traffic as we have ground truth from city traffic authorities to validate our results.

3. APPROACH

Current event extraction techniques use event specific patterns based on event types [Tanev et al. 2008; Grishman et al. 2002]. The text is expected to have some structure (e.g., news documents). Such a technique does not scale for city events from twitter text due to the informal nature of tweets. Further, the aggregation is done at a cluster level which is too coarse grained for city related events. E.g., important city events may be reported by few citizens given the wide scope of topics on twitter [Kwak et al. 2010]. In such situations, clustering based techniques [Naughton et al. 2006] may fail to separate minority tweets related to the city infrastructure.

Twitter messages may be noisy and convoluted with little context in which the message was generated. Such a characteristic of tweets challenges a dictionary based approach for spotting location and event terms. For example, Figure 2 shows one of many instances where a single event term ‘accident’ is used in different contexts. The tweet in Figure 2(a) refers to the dream a person had while the tweet in Figure 2(b) actually refers to an accident. The tweet in Figure 2(a) does not contain any location information while the tweet in Figure 2(b) has “Golden Highway” as the location. It is clear that relying solely on a dictionary based context free spotting of event terms cannot capture these nuances due to context-sensitive dependencies between words. It is possible to use a purely dictionary based approach for spotting event terms but would require human inspection for accurate tagging. Such manual inspection of the results of event spotting is infeasible because of the volume and velocity of the tweets, and the need for quick action. In order to automate this process, we formalized this problem of spotting event terms and location names as a sequence labeling problem. We then evaluated the performance of dictionary based spotting of event and location terms for a relative comparison with sequence labeling models. We provide some insights on the benefits of using dictionary based approaches for creating training data



Facts Guide @FactsGuide 2h
Ever had a dream about someone, snake, **accident**, train or murder? Your dream have deep meanings. See bit.ly/HuAWvQ

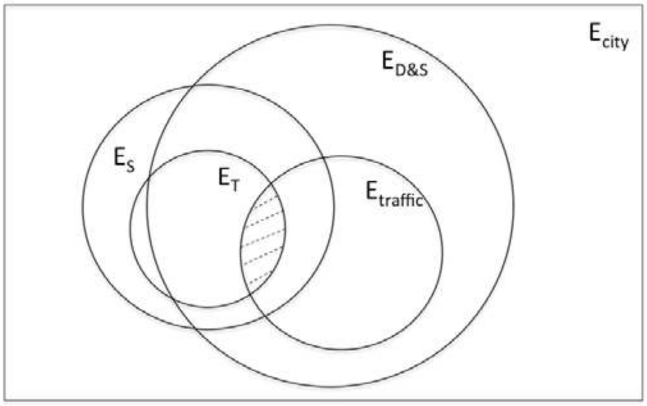
(a) Accident in the context of a dream

CHOCSpotter @PigSpotter 3h
Accident on the Golden Highway at the Viking robots in Devland JHB, ambo into a truck, injured treating themselves. pic.twitter.com/AAFxMgp7sO"

(b) Accident in the context of a road accident

Fig. 2. Addressing ambiguity: challenge for event extraction - tweets reporting very different events using the same event term 'accident'

Fig. 3. Depiction of city events (E_{city}, $\mathrm{E}_{D\&S}$, and $\mathrm{E}_{traffic}$) and its relationship to city events from social streams (E_S) and twitter (E_T)



instead of directly using the dictionary for entity spotting. In practice, the training data may require some cleaning depending on the required accuracy of spotting and availability of resources. Our city event extraction framework provides control over the manual effort required to clean the training data. We create a training set for building a Conditional Random Field (CRF) [Lafferty et al. 2001] model automatically, by using dictionary-based spotting, to reduce manual tagging effort. We organize the details of our approach into event annotation and extraction.

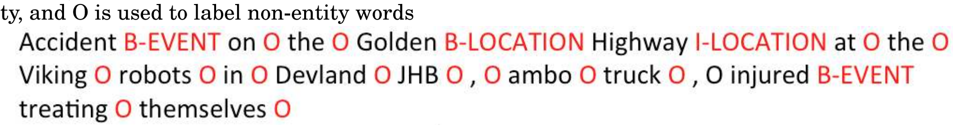
Our approach is motivated by the open domain event extraction from twitter [Ritter et al. 2012]. We introduce basic notations used in rest of the paper and elaborate on the solution components.

3.1. Preliminaries

Figure 3 summarizes the relationship between events in a city and citizen observations using Venn diagram. Let E_{city} be the set of city events. Let $\mathrm{E}_{D\&S}$ be the subset of events related to the city infrastructure (departments and services offered in a city) $\mathrm{E}_{D\&S} \subset \mathrm{E}_{city}$. $\mathrm{E}_{D\&S}$ is not directly observed but we have access to the social streams represented by S containing events E_S. E_S may contain a subset of events related to city department and services represented as $\mathrm{E}_S \cap \mathrm{E}_{D\&S}$. City events flow through two major information channels: formal reporting and informal reporting. In formal reporting, dedicated resources such as machine sensors or city department officials observe and report various city events. Citizens may report their observations of a city through location based services (e.g., foursquare[6]), event based services (e.g., eventful), and user generated content (e.g., blogs, posts, and tweets). We focus our attention on events from twitter stream, which have been widely accepted as a near real-time

[6] https://foursquare.com/

Fig. 4. A sample tag assignment to tokens (words) in a tweet where B-EVENT indicates beginning of an event entity, B-Location and I-Location indicates beginning and intermediate words or last word of a location entity, and O is used to label non-entity words



Accident B-EVENT on O the O Golden B-LOCATION Highway I-LOCATION at O the O Viking O robots O in O Devland O JHB O , O ambo O truck O , O injured B-EVENT treating O themselves O

source of citizen chatter [Nagarajan et al. 2009] about traffic events. The shaded region in Figure 3 represents traffic events from twitter. We represent the traffic events extracted from twitter by the set E_T. Specifically, we use E_T to present and evaluate our algorithms and techniques. We use $E_{traffic} \subset E_{D\&S}$ obtained from 511.org as ground truth due to its open availability.

3.2. Basic Notations

We define the event schema as a 5-tuple $\langle \hat{e}_{type}, \hat{e}_{loc}, \hat{e}_{st}, \hat{e}_{et}, \hat{e}_{impact} \rangle$ where $\hat{e}_{type}$ ranges over all the events in E_T extracted from twitter T. $\hat{e}_{type}$ refers to the event type such as accident, breakdown, and music event, $\hat{e}_{loc}$ refers to the location of the event (lat-long), $\hat{e}_{st}$ and $\hat{e}_{et}$ refers to the start time, and the end time of the event, and $\hat{e}_{impact}$ refers to a number quantifying the severity of the event. We use hat to emphasize that we are estimating these values as actual values are unavailable. When available, the ground truth of events from city authorities may be used as actual values.

3.3. Problem Formulation

We formulate the problem of detecting events from informal text as a sequence labeling and aggregation problem.

3.3.1. Annotation. A tweet is composed of a sequence of tokens, *tokens*($tweet_n$) where *tokens* is a function that emits tokens given input tweet, $tweet_n$. A tag is a label given to each token in the token sequence *tokens*($tweet_n$). To annotate multi-phrase entities, we use a variant of the widely accepted BIO notation [Ramshaw and Marcus 1999] in computational linguistics[7]. For a single word entity, we use the B- suffix/label. For a multi-word phrase, we use the B- suffix for the first word and I- suffix for all the subsequent words including the last word. If a word does not refer to an entity it is suffixed with O. Using BIO notation to annotate a location entity phrase *Golden Highway*, we get *Golden B-Location Highway I-Location*. Entities that are not related to events and locations are tagged as Other (O). In general, the tag set contains location tags (B-Location, I-Location) and event tags (B-Event, I-Event), Tag_{set} = { B-Location, I-Location, B-Event, I-Event, O }. We want to assign the most relevant tag to each token in *tokens*($tweet_n$) taking into account dependencies between tokens, e.g., phrase named entities and long distance dependencies. For example, occurrence of accident term along with a location name vs. dream in Figure 2. Figure 4 shows a sample tag assignment for tokens in a tweet.

3.3.2. Extraction. Once we have the most likely tag assigned to each token in a tweet, we proceed to perform city infrastructure related event extraction. Highly informal, redundant, and noisy nature of tweets requires us to rank and aggregate events based on location, time, and theme dimensions as detailed in the next section. Aggregation algorithm summarizes redundant report of events and creates a unique representation $\langle \hat{e}_{type}, \hat{e}_{loc}, \hat{e}_{st}, \hat{e}_{et}, \hat{e}_{impact} \rangle$ for each event by grouping quintuples $\langle n, e, t, d, l \rangle$ based on location, time, and theme dimension. As a pre-processing step, before emitting the

[7] http://en.wikipedia.org/wiki/Inside_Outside_Beginning

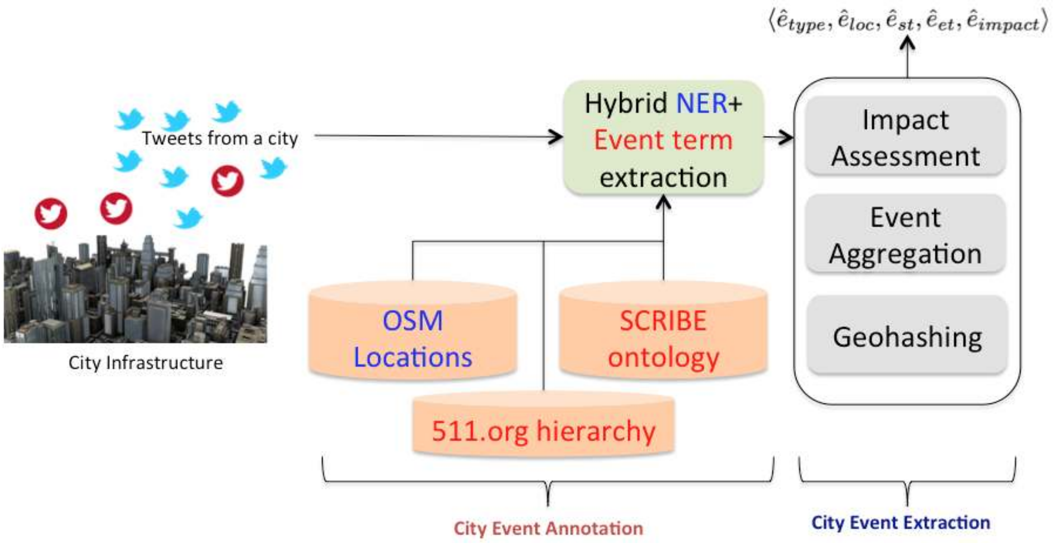


Fig. 5. Architecture for extracting city infrastructure related events from social stream such as tweets

event tuple $\langle \hat{e}_{type}, \hat{e}_{loc}, \hat{e}_{st}, \hat{e}_{et}, \hat{e}_{impact} \rangle$, we emit a quintuple $\langle n, e, t, d, l \rangle$ for each tweet collected from a city where *n* represents location terms, *e* consists of the event terms, *t* represents the time of day, d represents the day of week, and *l* represents the geohash location. We run aggregation algorithms on this representation to uncover events in an unsupervised manner. We cluster tuples $\langle n, e, t, d, l \rangle$ for deriving the event tuple, e$_i \in E_T$, based on event terms and then filter based on geohash location. Event terms that we spot in tweets directly map to event types (syntactically) in a comprehensive hierarchy of events provided by 511.org[8]. This mapping is due to the use of training data (to train the CRF model) containing event types from the same hierarchy. This hierarchy has *active-events* and *scheduled-events* as two major categories. Unlike conventional event extraction from text, city events require aggregation algorithms to be strongly tied with location (space) and time.

3.4. Solution Components

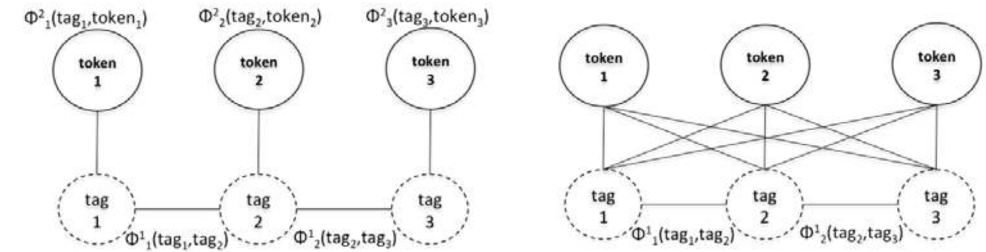
The tweet processing pipeline is shown in Figure 5 and the details of each solution component is presented here. Location and time are crucial components for city related event extraction. We are exploiting spatio-temporal context/coherence for city event extraction from informal text.

3.4.1. City Event Annotation. A CRF model is an undirected graphical model [Koller and Friedman 2009; Elkan 2008; Chen 2012] containing nodes that correspond to the set: ***tokens***$(tweets_n)$ $\cup$ Tag$_{set}$. The model defines factors to capture dependencies between (a) neighboring tags (tag_i, tag_{i+1}) and (b) tags and tokens sequence $(tag_1, token_1)$,...,$(tag_i, token_i)$,...,$(tag_m, token_m)$ where $tag_i \in Tag_{set}$ and $token_i \in$ ***tokens***$(tweets_n)$. A factor is a function that maps all possible values of input variable combinations to real numbers, formally represented as $V \mapsto \mathbb{R}$ where $V \subset$ ***tokens***$(tweets_n)$ $\cup$ Tag$_{set}$. This number is also called the potential for the input variable combinations, e.g., $\phi(tag_i, tag_{i+1})$ captures the number of times tag_i appears before tag_{i+1} in a corpus. Concretely, if tag_i is B-Location and tag_{i+1} is I-Location, ϕ(B-Location, I-Location)

[8] Metropolitan Transportation Commission, http://511.org

maps to the number of times this sequence appears in the corpus. This may not be a normalized value. If ***tokens***$(tweets_n)$ = $\{token_1, token_2, ..., token_m\}$, we define factors $\phi(tag_i, token_i)$ for each token where the $token_i$ is always observed. This factor captures the number of times the token $token_i$ was labeled with the tag tag_i. Concretely, if $token_i$ is *Golden* and tag_i is B-Location, then ϕ(B-Location, Golden) captures the number of times the token *Golden* was labeled with the tag B-Location in the corpus. A simplified example of the model is shown in Table I. If there are *m* tokens in a sequence, we need (*m* - 1) factors to define potentials between neighboring tags and *m* factors to define potentials between tags and tokens. Finding the most likely tag assignment can be formalized as maximizing the probability $P(tags|tokens(tweet_n))$ shown in Table I(a). $\widetilde{P}(tags, tokens)$ is the unnormalized score for a configuration of tokens and its tag assignment represented by *tags*. The term *argmax* selects the tag assignment for all the tokens based on the highest probability score. Even though the model captures potentials between adjacent tags, tag assignment is done based on the global maximum, i.e., tags that result in highest overall score are assigned to all the tokens. Such a global assignment of tags naturally captures long distance dependencies in text. The location and event spotting model use the linear chain CRF model presented in Table I(b) and implemented by LingPipe[Alias-i 2008]. The LingPipe implementation of CRF uses a slightly different model compared to the simplified one in Table I(a). Both the CRF models in Table I(a) and (b) are linear chain CRFs. Linear chain CRF restricts the factors to be defined between adjacent tags. Arbitrary tag dependencies are not allowed in the CRF model. The CRF model also disallows joint distribution among the tokens. But the tags may depend on any arbitrary feature extracted from the sequence of *tokens*. Each tag type and its positions in a corpus are extracted using a feature extractor function *f* ∈ F which takes current token position and tag assigned to the

Table I. Formalization of sequence labeling task using a Conditional Random Field (CRF) on the left and LingPipe CRF implementation on the right



(a)

$P(tags|tokens) = \frac{1}{Z(tokens)}\widetilde{P}(tags, tokens)$

$\widetilde{P}(tags, tokens)$
$= \prod_{i=1}^{m-1} \phi(tag_i, tag_{i+1}) \prod_{i=1}^{m} \phi(tag_i, token_i)$

$Z(tokens) = \sum_{tags} \widetilde{P}(tags, tokens)$

$argmax_{tags \in Tag_{set}} P(tags|tokens(tweet_n))$

(b)

$tokens = x[1], ..., x[N]$
$tags = 1, ..., m$

$F = f(pos = i, prevTag = j)\forall i = 1, j = null\, and\, i = 2, ..., N; j = 1, ..., m$
$\beta[m] < -LogisticRegression(F)$

$\widetilde{P}(tags, tokens) = \phi(tag_1, ..., tag_N|x[1], ..., x[N])$

$p(tag_1, ..., tag_N|x[1], ..., x[N]) =$
$\frac{\phi(tag_1,...,tag_N|x[1],...,x[N])}{Z(x[1],...,x[N])}$

$\phi(tag_1, ..., tag_N|x[1], ..., x[N]) = f(1, null) \cdot \beta[tag_1] +$
$f(2, tag_1) \cdot \beta[tag_2] + ... + f(N, tag_{N-1}) \cdot \beta[tag_N]$

$Z(x[1], ..., x[N]) =$
$\sum_{tag_1=1:m} \cdots \sum_{tag_N=1:m} \phi(tag_1, ..., tag_N|x[1], ..., x[N])$

previous token as input. The first token in the sequence will have *null* as the previous tag. For rest of the tokens in the input sequence, the feature function is invoked with all possible tags (1,..,m). $\beta[m]$ are the coefficient vectors learned for each output tag in the tag set T_{tags} where *m* is the number of tags from the corpus in the training phase. $\beta[m]$ is learned using Logistic Regression in the LingPipe implementation. The corresponding unnormalized score for tag assignment given tokens represented as $\phi(tag_1, ..., tag_N|x[1], ..., x[N])$ is computed using the dot product of extracted features and the coefficient vectors. To get the probability of tag assignment given a token sequence, this term needs normalization by summation over all possible tags represented by the term $Z(x[1], ..., x[N])$ as shown in Table I(b). Though the features are extracted locally using the function *f*, the global normalization captures long distance relationships in the token sequences.

Training the CRF Model: Our objective is to spot event and location terms in tweets. Identifying locations in a tweet is challenging as location references are hard to recognize especially in the presence of non-standard abbreviations, spellings, and capitalization convention. A sample tweet with location and event annotations is shown in Figure 4. To address these challenges, we train the sequence model with the knowledge of locations from Open Street Maps (OSM) [Haklay and Weber 2008] for a specific city. OSM data is available for most of the cities around the world. Identifying event terms in tweets is challenging, especially given the open domain nature of city related events. Background knowledge consisting of domain vocabulary is obtained from 511.org, which provides a hierarchical classification of traffic related events. E.g., music event, sporting event, and road work that are categorized as *scheduled events* and accident, break down, and protests are categorized as *active events*. We generate training data automatically using the knowledge of locations and event terms using a dictionary based spotting. The training data may be cleaned before using it for training the CRF model. Cleaning refers to removing annotated tweets that have ambiguous references (Figure 2). Depending on the availability of resources, our city event annotation framework offers flexible manual control. Desired accuracy of spotting location and event terms would determine the extent to which the training data should be cleaned. Given the open domain nature of city events and robustness of our event extraction algorithms, it was not necessary to clean the training data. We compare our CRF model trained on this automatically created training data (without cleaning) with the baseline CRF model reference which is trained on manually created training data. Our approach shows promising results as evident by Figure 9 and Figure 10 based on the precision, recall, and F-measure metrics.

3.4.2. City Event Extraction. Using the named entities and event phrases extracted from tweets, we derive unique events in the city. There may be multiple references to the same event. Further, an event phrase may co-occur with multiple event types. For a reliable event extraction, we follow a systematic approach as outlined below (each component in the city event extraction box of Figure 5 is detailed here).

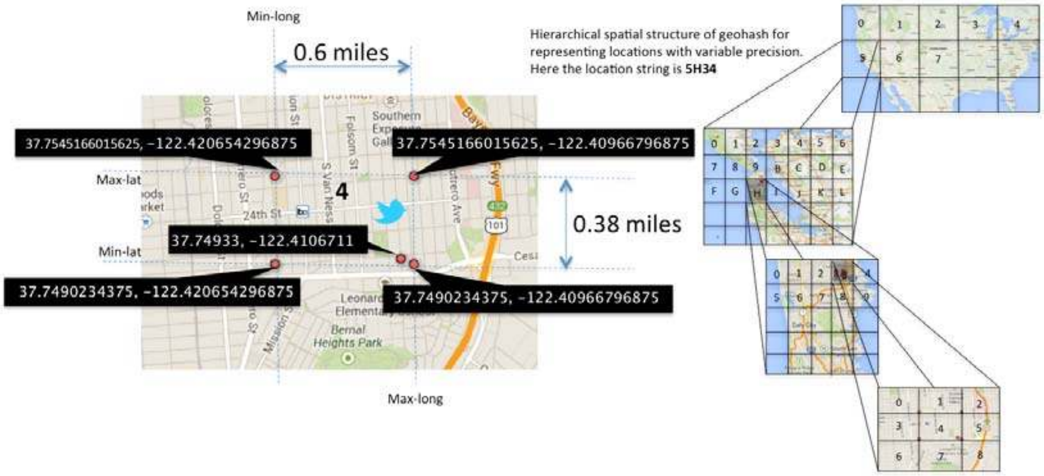
(1) Geohashing: We split the city into grids of a specified area using the geohashing algorithm[9]. These grids compartmentalize a city into various spatial regions. Different grids correspond to different levels of granularity. The spatial precision increases with the length of the string representing a location. We assign unique grid number to each grid. Figure 6 presents a geohashing example for San Francisco Bay Area and shows a tweet reported within the geohash. We associate a unique identifier to the location meta data of the tweet originating from that grid. Algorithm 1 transforms a raw tweet

[9] http://wiki.xkcd.com/geohashing/The_Algorithm

ALGORITHM 1: Populating metadata for each tweet

```
Input: $tweet_{ts,lat,long}, CRF_{model}$
Output: $\langle n, e, t, d, l \rangle$
n := spotEntities($tweet_{ts,lat,long}, CRF_{model}$);
e := spotEventTerms($tweet_{ts,lat,long}, CRF_{model}$);
t := getTimeOfDay($tweet_{ts,lat,long}$);
d := getDayOfWeek($tweet_{ts,lat,long}$);
l := getGridNumberFromGeohash($tweet_{ts,lat,long}$);
return $\langle n, e, t, d, l \rangle$ ;
```

Fig. 6. Spatial region bounded by a box which is part of the geohashing scheme to split a huge geographical area into smaller addressable units. A tweet posted within this box is shown.



with timestamp (ts) and geo-location (lat and long) to feature vectors in the form of a quintuple $\langle n, e, t, d, l \rangle$. The CRF model is trained on automatically generated training set (using dictionary based spotting).

(2) Event Aggregation: After careful consideration of event characteristics in a city, we make *three assumptions* to group messages with event terms and location annotations in a city: (a) *Spatial Coherence*: Events reported within a grid $g_i \in G$ (where G is a set of all grids in a city) in the same time interval are associated with the same event. (b) *Temporal Coherence*: Events reported within a time interval Δt (difference between end time and start time) in a grid g_i are associated with the same event. (c) *Thematic Coherence*: Events with similar entities reported within a grid g_i and time Δt are associated with the same event.

Algorithm 2 presents our approach to derive city traffic related events from the feature vectors generated by Algorithm 1. The input to the algorithm are the feature vectors generated within a time interval Δt. The algorithm utilize the set of grids (squares) in a city G generated using a geohashing implementation[10] and the event hierarchy from 511.org. Algorithm 2 has three steps. First, each feature vector $\langle n, e, t, d, l \rangle$ associated with individual tweet is assigned an event type based on the event hierarchy of 511.org. The event type is assigned based on the event term e in the feature vector $\langle n, e, t, d, l \rangle$. Since the CRF model is trained using the 511.org hierarchy, the

[10] https://github.com/kungfoo/geohash-java

ALGORITHM 2: Derive event descriptions from feature vectors generated by Algorithm 1

```
Input: Representation of tweet content using quintuple $\langle n, e, t, d, l \rangle_N$ where n represents
       location terms, e consists of the event terms, t represents the time of day, d represents
       the day of week, and l represents the geohash location, and N refers to the number of
       input quintuples, $\Delta$t representing the time step such as hour, day, or week used to step
       through starting time $t_s$ and ending time $t_e$
Output: Event quintuple corresponding to a collection of tweets giving the aggregated type,
        location, start time, end time, and impact represented by
        $\langle \hat{e}_{i,type}, \hat{e}_{i,loc}, \hat{e}_{i,st}, \hat{e}_{i,et}, \hat{e}_{i,impact} \rangle_n$ where n represents the number of events
while $\exists \langle n, e, t, d, l \rangle$ within a time step $\Delta t$ do
    // Associate a type with each quintuple utilizing the 511.org event hierarchy
    for i := 1:N do
        $v_i := \langle n, e, t, d, l \rangle_i$ ;
        type := 511.org hierarchy term associated with e ;
        Assign event type type to $v_i$ ;
    end
    // Create event type buckets with corresponding feature vectors
    for i := 1:N do
        Collect all the feature vectors $v_i$ with the same type into an event type bucket $E[type_k]$
        where $k$ is the number of event types ;
    end
    // Filter cluster items based on grid information
    for i := 1:k do
        Find location with highest number of occurrence in $E[type_i]$ represented by $l_{max}$ ;
        Remove all the members of the set $E[type_i]$ whose location is not $l_{max}$ ;
    end
    // Derive event metadata from event clusters
    for i = 1:k do
        $\hat{e}_{i,type} := type_i$;
        $\hat{e}_{i,loc} := l_{max}$ associated with $E[type_i]$;
        $\hat{e}_{i,impact}$ := number of items in the set $E[type_i]$;
        $\hat{e}_{i,st}$ := smallest time stamp in the cluster $E[type_i]$ ;
        $\hat{e}_{i,et}$ := largest time stamp in the cluster $E[type_i]$ ;
        emit $\langle \hat{e}_{i,type}, \hat{e}_{i,loc}, \hat{e}_{i,st}, \hat{e}_{i,et}, \hat{e}_{i,impact} \rangle$ ;
    end
end
```

event term e would correspond to a type in the hierarchy. Second, the feature vectors are grouped together based on space, time, and theme information. Finally, in the third step, each event type cluster (set of feature vectors) is processed for gleaning the start and the end time, location, and impact of the event. Each feature vector has a unique grid associated with it and each grid is assumed to have a unique event. Start time of the event is approximated using the time stamp associated with the first tweet (determined by timestamp) in the cluster. End time of the event is estimated using the timestamp associated with the last tweet (determined by timestamp) in the cluster. For estimating event location, we count the maximum number of occurrences of location l_{max} in the event type cluster.

(3) Impact Assessment: Events may have varying impact on the functioning of a city. City authorities need to prioritize these events based on the severity level. For example, a pot hole on a major road can be more critical to fix than a pot hole on a smaller road that is used much less. City authorities have realized the importance

of impact assessment of city events[11]. Unlike formal incident reports, tweets do not contain easily accessible/decipherable information. We approximate the seriousness of an event is by the number of people reporting the event. Algorithm 2 captures this intuition for estimating the event impact.

Overall, we presented an approach for city event extraction in two steps: (a) *Annotation*: we use the CRF model trained using the automatically generated training data (using dictionary based spotting of OSM and 511.org entities) to determine city locations and event terms. (b) *Extraction*: Using the three key assumptions of spatial, temporal, and thematic coherence characterizing city events, we aggregate feature vectors to glean event meta data.

4. EXPERIMENTAL SETUP

To evaluate our approach, we need to prepare training and test datasets, and train a CRF model for annotating tweets.

4.1. Dataset Description and Evaluation Metric

To make the evaluation tractable, we constrain our experiments to the domain of traffic related events. This was motivated by the availability of ground truth data from city authorities of San Francisco Bay Area[12]. The proposed approach is generic enough that it can be applied to any other domain for which the ground truth is available. We propose a novel approach to create massive training data with minimum manual intervention. We leverage two external sources in the work: (1) Open domain knowledge available for a city, specifically, vocabulary related to traffic from 511.org, and (2) OSM for city locations. For those domains not covered by the 511.org hierarchy, a vocabulary of event terms should be augmented. We have collected data from 511.org and twitter for a period of four months (Aug 2013 to Nov 2013). We utilized the Java Messaging Service (JMS) to receive the traffic data in the form of an XML stream from 511.org. For collecting twitter data, we used the twitter streaming API with location bounding box as San Francisco Bay Area. There are over 8 million tweets collected for this time period, augmented with 162 million sensor data points, 180 scheduled events, and 335 active events. The total dataset size is around 7 GB. Incident reports and sensor data from 511.org may serve as the ground truth (though we use only incident reports in this work). Table II summarizes active and scheduled events along with various subtypes. Temporal distribution of events over the period of four months (Aug-Nov, 2013) is shown in Figure 7. There are more active events compared to scheduled events and the distribution is non-uniform (unpredictable). Scheduled events are distributed uniformly throughout four months. A spatial distribution of events is also presented in Figure 7. Traffic events are concentrated on major roads and central part of the city. Our objective of the evaluation is to quantify the extent to which our approach can recover traffic incidents from tweets. We compare our approach with a state-of-the-art baseline [Ritter et al. 2012] using precision, recall, and F-measure along with confusion matrix.

4.2. Training Data Creation

We use a novel approach to create city specific training data for sequence labeling task by utilizing the domain knowledge of city locations and event vocabulary.

4.2.1. Data Preprocessing. We create training data with location name annotations utilizing locations from Open Street Maps (OSM) [Haklay and Weber 2008]. We create

[11] http://www.kaggle.com/c/see-click-predict-fix
[12] http://511.org/developer-resources_traffic-data-feed.asp

Fig. 7. Spatio-temporal distribution of ground truth data consisting of Active and Scheduled events over four months obtained from 511.org

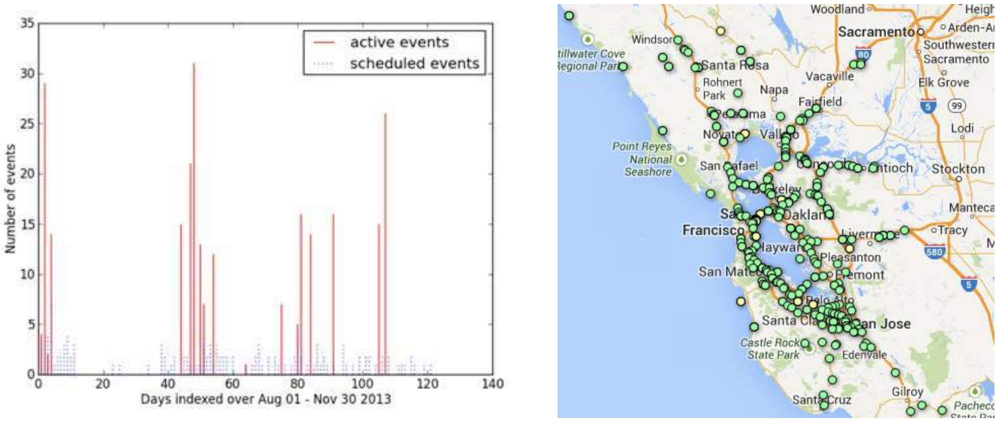


Table II. Ground truth events collected from 511.org along with their number of occurrence between August 2013 and November 2013

Active Events

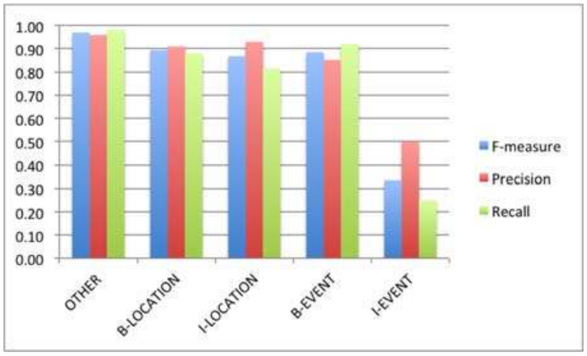
incident;truck-fire	2
special-events;festival	1
incident;emergency-maintenance	1
incident;accident-involving-a-motorcycle	1
obstructions;downed-power-lines	2
traffic-conditions;residual-delays	3
sporting-events;race-event	1
incident;disabled-semi-trailer	6
incident;accident	73
disasters;grass-fire	1
incident-response-status;police-department-activity	1
incident;injury-accident	13
obstructions;debris-on-roadway	3
incident;disabled-bus	1
incident;overturned-semi-trailer	1
visibility-air-quality;fog	1
incident;disabled-truck	3
disasters;fire	1
incident;multi-vehicle-accident	1
incident;spilled-load	1
sporting-events;baseball-game	3
incident;vehicle-on-fire	1
special-events;fair	9
roadwork;long-term-road-construction	7
incident;disabled-vehicle	9
incident;single-vehicle-accident	2
incident;road-construction	92
incident;spinout	1
obstructions;obstruction-on-roadway	38
winds;strong-winds	6
device-status;signal-problem	1
special-events;major-event	1
Total Number of Active Events	*311*

Schedule Events

race-event	5
fair	24
movie-filming	1
festival	5
long-term-road-construction	5
football-game	26
hockey-game	14
basketball-game	10
concert	25
race-eventmarathon	2
road-constructionpaving-operations	2
major-event	6
weekend-long-construction	1
soccer-game	1
concertfestival	1
baseball-game	42
Total Number of Scheduled Events	*170*

training data containing event term annotations using the hierarchical knowledge of traffic events from 511.org. There are two levels of filtering: (a) *Location based filtering* using the latitude-longitude of a bounding box around the city to filter tweets from the city, and (2) *Content based filtering* using location names on OSM and traffic related concepts on 511.org to filter tweets related to the domain of traffic events in the city. We propose a novel scalable solution for creating training data that utilizes available knowledge as a dictionary to annotate real-world data collected from twitter. We use the Aho-Corasick [Commentz-Walter 1979] string matching algorithm implemented by LingPipe [Alias-i 2008] to perform annotation of locations/event terms in linear time. The Aho-Corasick algorithm utilizes the dictionary of locations and event terms to spot

Fig. 8. Plot of Precision, Recall, and F-measure for the dictionary based training data creation process



entities in twitter text. Annotated tweets containing location and event terms are then used as a training sample for building a CRF model.

Table III. Evaluation results of the dictionary based training data creation process using precision, recall, and F-measure

		Actual Labels					Total	Precision
		OTHER	B-LOCATION	I-LOCATION	B-EVENT	I-EVENT		
Dictionary Annotation	OTHER	4267	62	113	6	3	**4451**	0.96
	B-LOCATION	37	451	7	1	0	**496**	0.91
	I-LOCATION	39	0	525	0	1	**565**	0.93
	B-EVENT	12	0	0	80	2	**94**	0.85
	I-EVENT	2	0	0	0	3	**4**	0.50
Total		**4357**	**513**	**645**	**87**	**8**		
Recall		0.98	0.88	0.81	0.92	0.25		
F-measure		0.97	0.89	0.87	0.88	0.33		

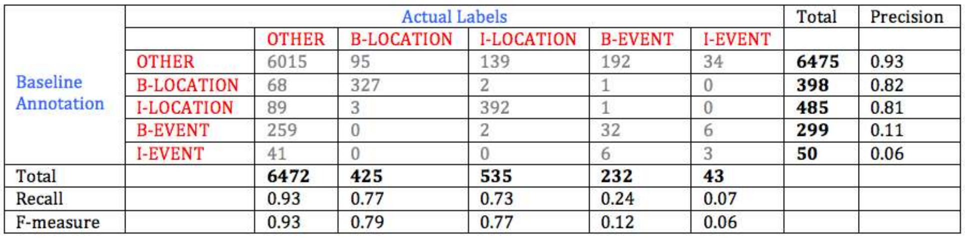
4.2.2. Preprocessing Evaluation. To understand the quality of training data, we evaluate the autonomous training data creation process. The evaluation is carried out by random sampling of annotated tweets. We select 500 random samples with 5,616 tags for evaluation. These samples were obtained from tweets generated during three months from Aug 2013 to Oct 2013. The results of evaluation are presented in Table III. We define *precision* as the ratio of number of correctly classified instances (tags) to the total number of instances. The total number of instances is the sum of correctly classified instances, N_C (sum of diagonal elements in Table III) and incorrectly classified instances, N_I (sum of non-diagonal elements in Table III). Our annotation process exhibits high quality with precision of around 94%. With such an accuracy, human intervention can be minimized or even eliminated. Since precision alone cannot provide insights into the annotation process, we present our results in the form of a confusion matrix in Table III. Recall that B- and I- refer to the beginning and intermediate tags respectively when there are multiple words in an entity name. *Location* and *event* are the two types of entities that contribute to event metadata. The *Other* tag is assigned to any other category of tokens. The loss of precision is due to the following challenges: (1) Subtle change in context results in varied interpretation of words, e.g., “Bed bath and beyond aha” can refer to a location or it is a casual remark not related to any location. This lack of context caused our annotator to mark this as location while in the evaluation we took a conservative approach of penalizing such annotations. (2) Intertwined space and time references cause loss of precision, e.g., “All them people from middle school and high school I don’t even talk to them anymore just shows me a lot” in which the author is not really referring to any location but merely referring to temporal dimension of life. Our annotator cannot differentiate between such references. (3) Subtle difference in location and event references, e.g., Twin Peaks in “Twin Peaks Summit” is labeled as location since Twin Peaks is actually a location. The word “Summit” makes the interpretation of the entire phrase as an event. Since our dictionary based annotation process is stateless, it cannot catch such subtle differences. All these limitations motivated us to move toward a tagger that can capture such dependencies between words. We utilized 8,074 annotated tweets as a training set for building a CRF model which addresses some of the limitations we described in this section.

4.3. Model Creation and Evaluation

We compare the CRF model created using our approach (with no manual intervention in creating the training data) with the baseline [Ritter et al. 2012] which was trained on a manually crafted dataset. A quantitative comparison of the two approaches for the annotation task is presented here.

We used 8,074 annotated tweets to train a linear chain CRF model. The trained CRF model, created using 8,074 annotated tweets, is used in Algorithm 1 for annotating location and event tokens. We use the CRF implementation provided by LingPipe [Alias-i 2008] for our experiments. We evaluate the tagging process on the data collected for the month of Nov 2013 (test data) which is not used in any of the previous experiments. This temporal separation of data is natural in the context of temporal streams such as microblogs. For scalability of our approach it is necessary to create training data in an autonomous manner. To explore this, we compare two scenarios. First, we evaluate the CRF model created by using the annotated data from the previous section as is. Evaluation is done with manual inspection in which location and event annotations are examined for correctness. Second, we annotate the microblog text using the baseline approach [Ritter et al. 2012] and evaluate the quality of annotation. We carry out the two experiments on 500 randomly chosen tweets from Nov 2013. The performance of annotation is evaluated using a confusion matrix for a deeper insight along with the precision, recall, and F-measure scores.

Table IV. Evaluation of annotation based on precision, recall, and F-measure metrics for baseline

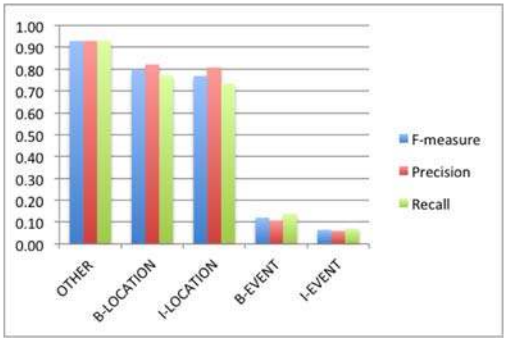


		Actual Labels					Total	Precision
		OTHER	B-LOCATION	I-LOCATION	B-EVENT	I-EVENT		
Baseline Annotation	OTHER	6015	95	139	192	34	**6475**	0.93
	B-LOCATION	68	327	2	1	0	**398**	0.82
	I-LOCATION	89	3	392	1	0	**485**	0.81
	B-EVENT	259	0	2	32	6	**299**	0.11
	I-EVENT	41	0	0	6	3	**50**	0.06
Total		**6472**	**425**	**535**	**232**	**43**		
Recall		0.93	0.77	0.73	0.24	0.07		
F-measure		0.93	0.79	0.77	0.12	0.06		

The baseline model [Ritter et al. 2012] is trained on a carefully annotated tweet corpus in three different categories of annotation. The first training dataset is for Part Of Speech (POS) tagged tweets. This data consists of tweets annotated with POS tags. Second training dataset consists of the tweet chunking information which has tags grouping the beginning and end of POS tags in a tweet e.g., to capture multi-word nouns. Third training dataset consists of the named entities. These datasets[13] are annotated using the BIO notation. The baseline dataset is created meticulously by manual inspection of tweets. This is an arduous task given the volume of tweets and the

[13] https://github.com/aritter/twitter_nlp

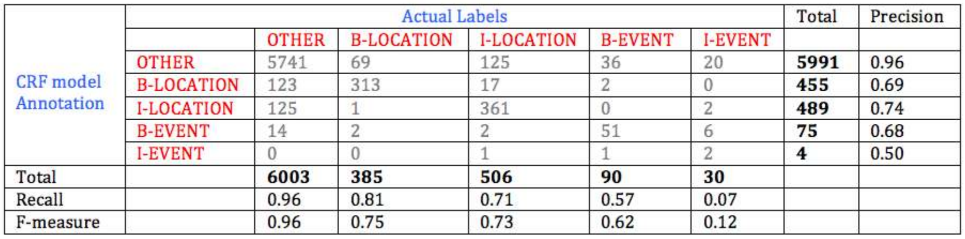
Fig. 9. Plot of Precision, Recall, and F-measure for the baseline annotation



challenges in understanding tweet content. Sometimes the lack of context is so serious that it can confound manual annotation. The precision of the baseline annotation task is shown in Table IV. The model suffers loss of precision mostly for the event term annotation justified by the lack of background knowledge of event terms. The baseline takes a high-recall low-precision approach so that the events can be ranked based on the rarity of events (done during the event aggregation phase). The precision, recall, and F-measure of the baseline approach is plotted in Figure 9. To understand the impact of event term annotation on the overall precision, we need to change the denominator term N$_I$, where N$_I$ is the sum of all the non-diagonal entries which constitutes the mis-classified instances. Consider the first column of Table IV. Number of instances that belonged to 'other' category but were classified as B-EVENT is given by the fourth entry in the first column. If we retain this, since baseline is based on the high-recall philosophy, we are unnecessarily penalizing the baseline. We can pretend that the baseline did not provide us with these event tags instead, it was tagged as other. This results in the modified precision for the baseline computed as

$$Precision_{baseline}^{ignore-event-annotation} = \frac{N_C}{N_C + N_I} = \frac{6769 + 259}{6769 + 259 + (938 - 259)} \approx 91\%$$

Table V. Evaluation of annotation based on precision, recall, and F-measure metrics for our approach



		Actual Labels					Total	Precision
		OTHER	B-LOCATION	I-LOCATION	B-EVENT	I-EVENT		
CRF model Annotation	OTHER	5741	69	125	36	20	**5991**	0.96
	B-LOCATION	123	313	17	2	0	**455**	0.69
	I-LOCATION	125	1	361	0	2	**489**	0.74
	B-EVENT	14	2	2	51	6	**75**	0.68
	I-EVENT	0	0	1	1	2	**4**	0.50
Total		**6003**	**385**	**506**	**90**	**30**		
Recall		0.96	0.81	0.71	0.57	0.07		
F-measure		0.96	0.75	0.73	0.62	0.12		

The evaluation of our approach is presented in the form of a confusion matrix in Table V. The proposed approach essentially takes a knowledge base as input for creating a training data without any manual intervention. The knowledge base used here are Open Street Maps and the event related knowledge from 511.org. Such a knowledge base consists of the vocabulary used in referring to various concepts in the domain and such a vocabulary is readily available for most of the cities. We did not perform any cleaning of the dataset before training the CRF model. Precision, recall, and F-measure of our approach is shown in Figure 10. This is on a par with the precision of

Fig. 10. Plot of Precision, Recall, and F-measure for our annotation process

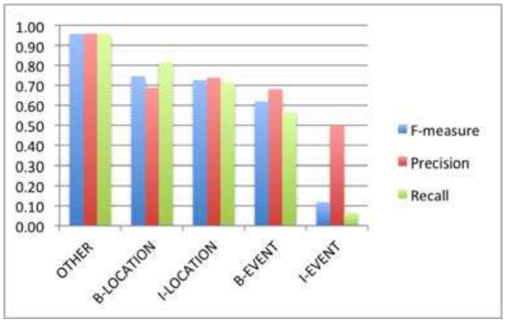


Table VI. Normalization of tags with baseline tag and the corresponding normalizing tag

Baseline tags	*Normalizing tag*
B-facility, B-company, B-geo-loc	B-LOCATION
I-facility, I-company, I-geo-loc	I-LOCATION
B-event	B-EVENT
I-event	I-EVENT
B-other, I-other, B-person, I-person, B-tvshow, I-tvshow, B-sportsteam, I-sportsteam, B-movie, I-movie, B-product, I-product, B-musicartist, I-musicartist	OTHER

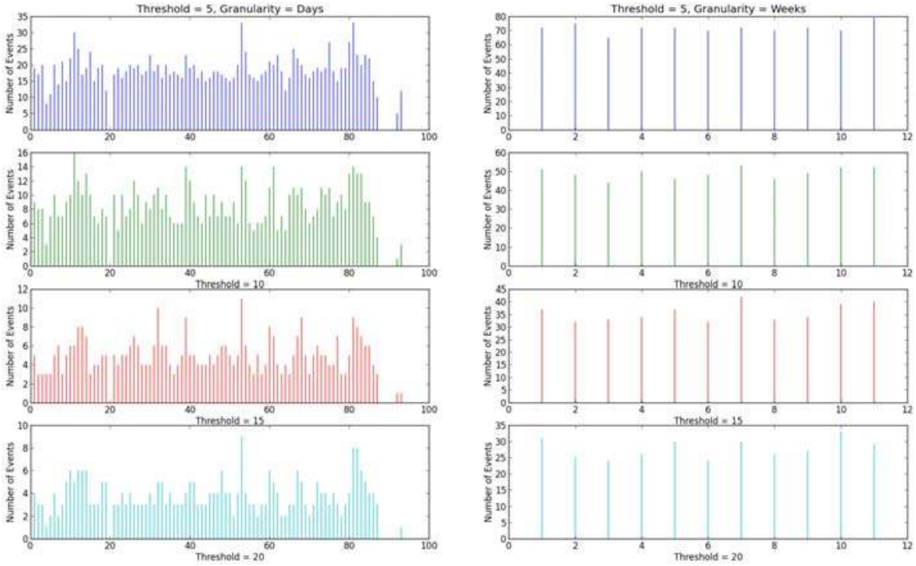
the baseline while reducing the tremendous effort involved for manually created training data. A city may have many other events of interest. We need a scalable approach that can leverage existing knowledge of the domain to feed the statistical NLP models (e.g., CRF) with automatically created training data. Our results show that this is indeed possible and we will explore this in our future research as well.

4.3.1. Principles for annotation of ground truth. We use the same ground truth to compare CRF model built on manual and automatically annotated corpus. The ground truth was created as follows: (1) *Normalization*: We perform tag normalization associated with locations and events by making the transformations as shown in Table VI. (2) *Minimality*: When we need to decide on including or excluding neighboring words within the location annotation, we look for the minimum words that provide a unique location hit on google maps. We stop including words that do not really contribute to location uniqueness, e.g., if *Bay Area Medical Academy San Francisco* appears in text, we believe that the first four words are enough to get a hit on google maps. (3) *Specificity*: Annotating specific locations and missing general locations is acceptable. We do this since we can infer the generic location from the specific location e.g., if *Bay Area Medical Academy* and *San Francisco* appear in a tweet, annotating the first location will allow us to infer the second location.

4.4. Scalability Challenges

We started by exploring the use of existing tools for building the CRF models. MALLET [McCallum 2002] is a comprehensive tool for Natural Language Processing with a suit of machine learning libraries. The learning technique used is the Limited-memory BFGS (L-BFGS) which does not scale for the dataset we have. LingPipe [Alias-i 2008] provides a CRF library which utilizes Stochastic Gradient Descent (SGD) as the learning mechanism and scales well for our dataset size. Our training set data generation

Fig. 11. Sensivity of event extraction to thresholds presented for the time granularity of days and weeks

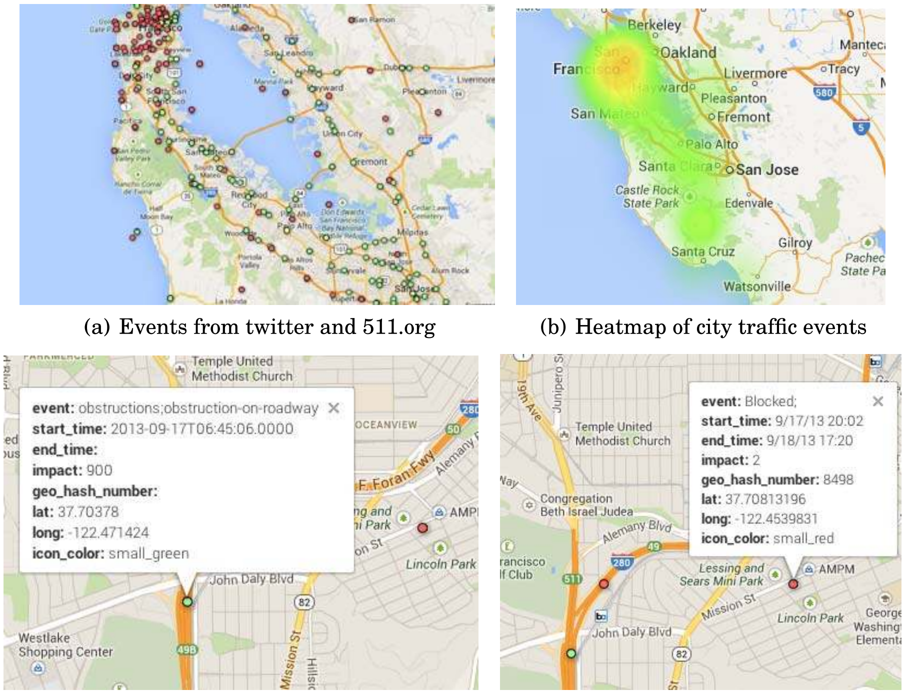


capability is massive and we will explore the scalable training of sequence models as a future work.

5. EVALUATION OF EVENT EXTRACTION

An implementation of all the algorithms presented in this paper along with complete dataset is available as an Open Science Framework project[14]. We evaluate the effectiveness of the events extracted by investigating if they are corroborative, complementary, or timely compared to the incident reports from 511.org. We believe that microblog alerts such as tweets are related to conventional sources such as news and incident reports by city authorities and sensor data in several ways. We call the event extracted from twitter as *corroborative* to the event from 511.org if they are reporting exactly the same event. We call the extracted event to be *complementary* if it provides additional information to 511.org events, e.g., extracted event may be traffic jam that further adds to the construction event from 511.org. The extracted event (complementary or corroborative) is called *timely* if it precedes the event reported on 511.org. They may provide additional information and may even help us explain some observations reported on conventional sources. For example, we extracted 'traffic' event from a textual stream and a 'baseball game' observation as reported from 511.org, and both these events have same space and time extent. Traffic information is complementary to the baseball game. If we extract baseball game event from textual stream, then the event will be corroborative (one supporting the other). If we extract any of the two events (traffic or baseball game in this example) before the incident report from 511.org then the extracted event is timely. We use these "characteristics" to manually verify each extracted event. The bottom-up nature of citizen sensing has both positive and negative

[14] https://osf.io/b4q2t/



(a) Events from twitter and 511.org

(b) Heatmap of city traffic events

event: obstructions;obstruction-on-roadway
start_time: 2013-09-17T06:45:06.0000
end_time:
impact: 900
geo_hash_number:
lat: 37.70378
long: -122.471424
icon_color: small_green

(c) Event from 511.org reporting obstruction

event: Blocked;
start_time: 9/17/13 20:02
end_time: 9/18/13 17:20
impact: 2
geo_hash_number: 8498
lat: 37.70813196
long: -122.4539831
icon_color: small_red

(d) Event from twitter reporting blockage

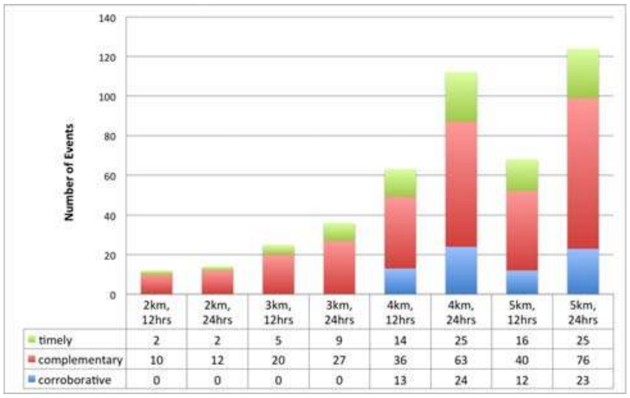
Fig. 12. Distribution of city events that were extracted from tweets along with the scheduled and active events from 511.org

implications. A positive implication is that such near real-time reports may surpass the conventional sources in terms of timeliness. A negative implication is that they can be contradictory or misleading.

For better understanding of the distribution of extracted events over time, we vary the granularity of time slices (days and weeks) and the threshold of minimum concurring tuples that constitute an event (5, 10, 15, and 20). In other words, our confidence on event occurrence depends on the cluster size. We study the variation of number of event tuples emitted based on a threshold of 5, 10, 15, and 20. We perform this study over days and weeks by varying granularity of time as shown in Figure 11. We chose days and weeks as time granularity and exclude months as it may not provide any additional insight. The x-axis is the time granularity and the y-axis is the number of events. An important thing to note is that the variation of threshold from 5 to 20 has not created any major change in the distribution of events over time. When threshold is low, there are lot more events reported per day. From the first row of Figure 11 we see that an average of 20 events are detected per day and 70 events per week. This is still small compared to the total number of tweets that are generated every day (which is in the thousands). There is a 50% drop in the number of events over days for every 5 unit increase in the threshold. For a threshold of 20, there are around 4 events per day.

A distribution of extracted events from twitter and 511.org events over the city of San Francisco are shown in Figure 12. Recall that the data was collected from Aug 2013 to Nov 2013. Our approach has been complementary to the standard sources of events in a city testified by the red dots (extracted event) in areas where green and

Fig. 13. Distribution of corroborative, complementary, and timely events across all the eight sets of event pairs



	2km, 12hrs	2km, 24hrs	3km, 12hrs	3km, 24hrs	4km, 12hrs	4km, 24hrs	5km, 12hrs	5km, 24hrs
🟩 timely	2	2	5	9	14	25	16	25
🟥 complementary	10	12	20	27	36	63	40	76
🟦 corroborative	0	0	0	0	13	24	12	23

yellow dots (511.org events) are sparse. Green dots indicate active events (e.g., accidents, breakdowns, blocked roads) while the yellow dots indicate scheduled events (e.g., baseball game, concert, maintenance). Also, the extracted events appear on highways or near major roads. Further, the events are concentrated in the city center where we expect a lot of activity as summarized by the heatmap.
Figure 12(c) has an active event reported by 511.org and Figure 12(d) is the city event extracted from twitter. The 511.org event is related to obstruction on roadway reported at 2013-09-17 06:45. The extracted event provides insight that the event has lasted from 2013-09-17 20:02 to 2013-09-18 17:20 beyond the event occurrence time. Such complementary information will be useful for understanding the impact of an event.

We summarize the result of evaluation for 1,042 city events extracted from all the tweets collected over four months by considering 26 of them for clarity in this section. The event extraction was done using a threshold of 10 as we wanted to strike a balance between number of events and the manual effort involved in examining the results. The extracted city events and a comparison to ground truth (corresponding entries in Table VIII), in terms of whether they are complementary (CP), corroborative (C), or timely (T) events are presented in the Table VII. Each tuple in Table VII represents an event and the subscript of the tuple refers to the relationship of the event to the corresponding ground truth in Table VIII containing scheduled (s) and active (a) events. Note that there are 26 lines in both the tables with line to line correspondence. Table IX lists all the event-types for the extracted events. Events 1, 3, 4, 7, 8, 12, 15, 16, 20, 21, 22, 23, 25, and 26 provide a complementary view of bad traffic conditions during events of road constructions, baseball games, football games, and hockey games. Events 2, 5, 6, 9, 10, 14, 17, 18, and 19 are corroborative, reporting events that are consistent with events from 511.org. Events 5, 10, 13, 14, 15, 18, 19, 22-25 are reported before the events from incident reports of 511.org making twitter a timely source of information for city authorities to react. The dataset[15] used in this evaluation, consisting of twitter data and 511.org data, are available as an Open Science Framework project[16] for the research community.

[15] http://harp.cs.wright.edu/cityevents/
[16] https://osf.io/b4q2t/

Table VII. Events extracted from textual stream compared with ground truth of 511.org categorized as C = corroborative, CP = complementary, and T = timely

1	⟨ traffic, [37.642231, -122.426173], 2013-07-31 19:48:33, 2013-08-01 19:02:46, 31⟩$_{CP}$
2	⟨festival, [37.35676, -122.117852], 2013-08-04 19:14:34, 2013-08-05 03:21:51, 30⟩$_{C}$
3	⟨traffic, [37.77752898, -122.4034819], 2013-08-25 19:35:26, 2013-08-26 18:06:21, 14⟩$_{CP}$
4	⟨traffic, [37.78151103, -122.4189619], 2013-09-03 20:03:19, 2013-09-04 18:53:31, 41⟩$_{CP}$
5	⟨concert, [37.35676, -122.117852], 2013-09-11 21:09:20, 2013-09-12 18:49:57, 29⟩$_{C,T}$
6	⟨football game, [37.39611, -121.931096], 2013-09-13 19:20:54, 2013-09-14 18:58:32, 14⟩$_{C}$
7	⟨festival, [37.35676, -122.117852], 2013-09-15 19:13:52, 2013-09-16 08:24:00, 35⟩$_{CP}$
8	⟨blocked, [37.77322896, -122.4254819], 2013-09-18 20:00:41, 2013-09-19 19:03:12, 18⟩$_{CP}$
9	⟨concert, [37.561391, -122.096567], 2013-10-07 19:29:54, 2013-10-08 18:40:41, 22⟩$_{C}$
10	⟨concert, [37.35676, -122.117852], 2013-10-09 19:21:01, 2013-10-10 18:22:25, 28⟩$_{C,T}$
11	⟨accident, [37.517208, -121.948119], 2013-10-10 21:57:43, 2013-10-11 18:12:01, 14⟩$_{CP}$
12	⟨traffic, [37.707621, -122.340281], 2013-10-13 19:27:10, 2013-10-14 16:45:49, 27⟩$_{CP}$
13	⟨fog, [37.77578298, -122.5136819], 2013-10-18 19:45:36, 2013-10-19 15:23:28, 14⟩$_{C,T}$
14	⟨festival, [37.604053, -122.472817], 2013-10-18 21:43:32, 2013-10-19 18:37:36, 30⟩$_{C,T}$
15	⟨traffic, [37.39611, -121.931096], 2013-10-18 19:15:36, 2013-10-19 15:40:18, 11⟩$_{CP,T}$
16	⟨traffic, [37.77994998, -122.4591199], 2013-10-18 19:16:12, 2013-10-19 18:32:52, 46⟩$_{CP}$
17	⟨fog, [37.561391, -122.096567], 2013-10-21 20:01:34, 2013-10-22 18:37:42, 45⟩$_{C}$
18	⟨concert, [37.329895, -122.065265], 2013-10-21 19:32:02, 2013-10-22 19:07:38, 43⟩$_{C,T}$
19	⟨accident, [37.77322896, -122.4254819], 2013-10-22 19:23:43, 2013-10-23 18:08:55, 14⟩$_{C,T}$
20	⟨traffic, [37.77322896, -122.4254819], 2013-10-30 20:22:54, 2013-10-31 11:14:44, 21⟩$_{CP}$
21	⟨traffic, [37.79262801, -122.4063839], 2013-11-03 13:03:42, 2013-11-03 21:43:06, 11⟩$_{CP}$
22	⟨traffic, [37.474743, -122.303362], 2013-11-13 00:49:38, 22013-11-13 22:57:06, 27⟩$_{CP,T}$
23	⟨traffic, [37.200495, -122.202653], 2013-11-14 01:15:10, 2013-11-14 23:29:47, 24⟩$_{CP,T}$
24	⟨tornado, [37.77502896, -122.4384818], 2013-11-17 01:38:36, 2013-11-17 19:03:57, 11⟩$_{CP,T}$
25	⟨traffic, [37.76405301, -122.4066841], 2013-11-22 02:51:30, 2013-11-22 22:12:16, 10⟩$_{CP,T}$
26	⟨traffic, [37.39611, -121.931096], 2013-11-27 00:40:34, 2013-11-28 00:06:56, 89⟩$_{CP,T}$

5.1. Global Evaluation

We extend the evaluation to include all the 1,042 city events extracted from twitter and present a detailed evaluation by comparing them with all the 481 events from 511.org. Our evaluation strategy involves chunking the events based on location and time. We find out event pairs $\langle e_t, e_{511} \rangle$ that coexist within a radius of 2, 3, 4, and 5 kilometers where e_t is the event extracted from twitter and e_{511} is the event from 511.org. For each radius value, we find event pairs that coexist within the 12 hours and 24 hours window. Thus we have eight sets containing event pairs $\langle e_t, e_{511} \rangle$ each of which is evaluated for being complementary, corroborative, or timely. Figure 13 presents a distribution of extracted events from twitter as complementary, corroborative, or timely when compared to events from 511.org. The evaluation over all the eight sets containing event pairs is summarized. Although we have extracted many (1,042) city traffic events, around 40% of them (454) co-existed (based on location and time constraints stated above) with ground truth data. Our approach has potential to discover lot more city traffic events unreported on 511.org but we did not have the ground truth for verification.

6. DISCUSSION

Use of microblogs such as tweets for city related event extraction is indeed feasible. Citizens form an important part of a city and tapping directly into citizen observations provide a fine-grained view of city infrastructure. Sometimes the cost of setting up hardware infrastructure may hinder city authorities from gaining access to city events. In such situations, as demonstrated in this work, social streams can be used as an important source of city related events. Further, city events are strongly tied to space and time that can be exploited for event extraction and aggregation.

Table VIII. Incident reports from 511.org that corresponds to the extracted events in Table VII with subscript a = active events, s = scheduled events

1	$\langle$incident;road-construction, [37.628892, -122.41652], 2013-07-31T09:19:46.0000, 1800$\rangle_a$
2	$\langle$fair, [38.433036, -122.703], 2013-08-04T10:00:00.0000, 2013-08-04T23:00:00.0000$\rangle_s$
3	$\langle$football-game, [37.715272, -122.387296], 2013-08-25T13:00:00.0000, 2013-08-25T21:00:00.0000$\rangle_s$
4	$\langle$baseball-game, [37.778752, -122.390288], 2013-09-03T18:15:00.0000, 2013-09-09T23:00:00.0000$\rangle_s$
5	$\langle$concert, [37.423516, -122.07812], 2013-09-14T09:00:00.0000, 2013-09-14T23:00:00.0000$\rangle_s$
6	$\langle$football-game, [37.87112, -122.251824], 2013-09-14T11:59:00.0000, 2013-09-14T20:00:00.0000$\rangle_s$
7	$\langle$concert, [37.423516, -122.07812], 2013-09-15T10:00:00.0000, 2013-09-15T23:00:00.0000$\rangle_s$
8	$\langle$incident;accident, [37.768712, -122.407712], 2013-09-17T17:53:53.0000, 900$\rangle_a$
9	$\langle$concert, [37.332192, -121.900544], 2013-10-07T18:30:00.0000, 2013-10-07T23:00:00.0000$\rangle_s$
10	$\langle$concert, [37.423516, -122.07812], 2013-10-12T18:00:00.0000, 2013-10-12T23:00:00.0000$\rangle_s$
11	$\langle$baseball-game, [37.750956, -122.202232], 2013-10-10T16:00:00.0000, 2013-10-10T21:15:00.0000$\rangle_s$
12	$\langle$football-game, [37.715272, -122.387296], 2013-10-13T09:30:00.0000, 2013-10-13T18:00:00.0000$\rangle_s$
13	$\langle$visibility-air-quality;fog, [37.810832, -122.477416], 2013-10-19T22:55:47.0000, 1800$\rangle_a$, $\langle$visibility-air-quality;fog, [37.818596, -122.478584], 2013-10-19T22:57:05.0000, 1800$\rangle_s$
14	$\langle$festival, [37.43372, -122.468288], 2013-10-19T08:00:00.0000, 2013-10-19T18:00:00.0000$\rangle_s$
15	$\langle$incident;road-construction, [37.395324, -121.873672], 2013-10-19T22:16:50.0000, 1800$\rangle_a$
16	$\langle$visibility-air-quality;fog, [37.810832, -122.477416], 2013-10-19T22:55:47.0000, 1800$\rangle_a$, $\langle$visibility-air-quality;fog,[37.818596 -122.478584], 2013-10-19T22:57:05.0000, 1800$\rangle_a$
17	$\langle$visibility-air-quality;fog, [37.5402, -122.06688], 2013-10-20T08:00:30.0000, 1800$\rangle_a$
18	$\langle$concert, [37.332192, -121.900544], 2013-10-18T18:30:00.0000, 2013-10-18T23:00:00.0000$\rangle_s$
19	$\langle$incident;accident, [37.749184, -122.4038], 2013-10-23T08:32:18.0000, 1800$\rangle_a$
20	$\langle$incident;road-construction, [37.788776, -122.387808], 2013-10-30T00:00:57.0000, 28800$\rangle_a$
21	$\langle$football-game, [37.750956, -122.202232], 2013-11-03T09:00:00.0000, 2013-11-03T17:30:00.0000$\rangle_s$
22	$\langle$incident;road-construction, [37.324488, -122.399984], 2013-11-13T07:05:06.0000, 28800$\rangle_a$
23	$\langle$incident;road-construction, [37.258392, -122.122008], 2013-11-15T09:54:44.0000, 28800$\rangle_a$
24	$\langle$winds;strong-winds, [37.779968, -122.398416], 2013-11-21T20:36:36.0000, 14400$\rangle_a$
25	$\langle$incident;disabled-semi-trailer, [37.810656, -122.364336], 2013-11-22T08:44:45.0000, 1800$\rangle_a$
26	$\langle$hockey-game, [37.332192, -121.900544], 2013-11-27T18:30:00.0000, 2013-11-27T23:00:00.0000$\rangle_s$

Table IX. Types of events extracted from textual stream (originally from the 511.org hierarchy of traffic related events)

accident, blocked, left lane blocked, right lane blocked, baseball game, circus, cleared, concert, construction, crime, crowded, delay, dew, festival, fog, football game, frost, hurricane, incident, marathon, olympics, parade, performing arts, protest, rain, road construction, shooting, showers, snow, soccer game, toll plaza, tornado, tournament, traffic, weather

6.1. Techniques for Annotation

Existing approaches to event extraction from twitter that rely on n-gram based techniques are limited because City-related events are often too sparse and N-gram based techniques will obscure such events. The N-gram based techniques do not distinguish between semantic types such as location, person, or event terms which are crucial for extracting event metadata. N-gram based techniques cannot capture the subtle structure in the text messages which may help us to understand and disambiguate entities. Sequence labeling for identifying entities proved to be a good solution in extracting city related events. Automated generation of training data using existing knowledge of a domain is feasible allowing us to use similar technique across different domains relevant to a city.

6.2. Techniques for Aggregation

The principled approach of event aggregation process (discussed in detail in solution components) is based on the three coherence dimensions: spatial, temporal, and thematic. Algorithm 2 utilize Δt as the time granularity for aggregation, which has currently been set to a single day. Though we have provided a rationale for choosing Δt as a single day in evaluation, there are reasons to use more fine-grained or more coarse-grained granularity for specific events. In the future, we propose to explore adaptive techniques that decide time window based on the event type, location and time, or combine/abstract events happening in a single grid.

6.3. Validation of Extracted Events

The ground truth validation revealed the complementary, corroborative, and timely nature of events extracted from textual streams. Incident reports from city authorities are not fine grained enough to validate all the extracted events. We would like to explore the use of sensor data from 511.org that has fine grained speed, volume, and travel time for various road segments in San Francisco Bay Area to validate extracted events in the future work. At least a part of these events may influence traffic and we would like to investigate the correlation between events and change in travel time in the road segments surrounding the event. Study of such dependencies would allow us to profile various city events to obtain actionable insights helpful to city authorities.

6.4. Scalability Challenges

The amount of training data we could generate using our approach could not be completely utilized in the training of the sequence labeling model. We had to select a random subset of data for training the sequence labeling model. A future research direction is to investigate scalable training of sequence labeling models such as CRFs. Further, event aggregation should run in near real-time for timely decision support. We would like to explore the inherent parallelism in the problem of city event extraction, e.g., events are localized to a geohash grid. There could be a dedicated instance of event extraction algorithm processing data for each grid.

6.5. Application Scenarios

The city traffic events extracted from textual streams can have dual benefits. First, the decision makers or city planners can tap into these events for resource allocation and planning. Second, the citizens can leverage this information for smooth functioning of their daily activities. This work was motivated by some of the requirements of city partners in the CityPulse[17] project from a comprehensive list of scenarios[18] for city traffic event extraction.

7. CONCLUSIONS

Entity identification techniques such as sequence labeling are helpful in deciphering microblogs. The training data creation process that leveraged knowledge base of locations and event terms generated good quality training data. As such, the CRF model trained on this data performed on a par with a CRF model trained over manually created dataset. Furthermore, city events are open domain, so we need automated ways of creating training data for developing annotation models. The extracted events proved to be complementary, corroborative, or timely compared to the incident reports (from 511.org). As such, microblogs can serve as valuable enhancement to 511.org for analyzing and understanding road traffic.

Scalable training of sequence labeling models will be required for utilizing the automatically created training data. For better impact assessment, considering time of day, day of week, type of incident, etc., may be significant. Event extraction will help us know events, but to reveal valuable insights, we need to understand the relationships between various events. We will pursue this as future work and will explore techniques for understanding intricate relationships between city events. We believe that declarative knowledge such as domain ontology or commonsense knowledge such

[17] http://www.ict-citypulse.eu/page/

[18] http://www.ict-citypulse.eu/scenarios/

as ConceptNet[19] would provide valuable support for understanding relationships between various events. Thus, we are also exploring declarative knowledge driven statistical model creation for understanding city events.

ACKNOWLEDGMENTS

This paper describes work undertaken in the context of the EU FP7 Citypulse project contract number: 609035. We thank NSF for sponsoring SoCS grant IIS-1111182 titled as 'Social Media Enhanced Organizational Sensemaking in Emergency Response' for partially enabling this work.

References

Charu C Aggarwal and Karthik Subbian. 2012. Event Detection in Social Streams.. In *SDM*, Vol. 12. SIAM, 624–635.

Alias-i. 2008. LingPipe 4.1.0. (2008). http://alias-i.com/lingpipe

Pramod Anantharam and Biplav Srivastava. 2013. City Notifications as a Data Source for Traffic Management. In *Proceedings of the 20th ITS World Congress 2013*.

Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond Trending Topics: Real-World Event Identification on Twitter.. In *ICWSM*.

Jennifer Bélissent. 2010. Getting clever about smart cities: new opportunities require new business models. (2010).

Jennifer Bélissent. 2013. Service Providers Accelerate Smart City Projects. (2013). http://www.forrester.com/pimages/rws/reprints/document/82981/oid/1-LTEQ9N

Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. 2006. Participatory sensing. (2006).

Edwin Chen. 2012. Introduction to Conditional Random Fields. (2012). http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

Beate Commentz-Walter. 1979. *A string matching algorithm fast on the average*. Springer.

Wenwen Dou, K Wang, William Ribarsky, and Michelle Zhou. 2012. Event Detection in Social Media Data. In *IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content*. 971–980.

Charles Elkan. 2008. Log-linear Models and Conditional Random Fields, Video lectures link:. (2008). http://videolectures.net/cikm08_elkan_llmacrf/

Luca Filipponi, Andrea Vitaletti, Giada Landi, Vincenzo Memeo, Giorgio Laura, and Paolo Pucci. 2010. Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors. In *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*. IEEE, 281–286.

Ralph Grishman, Silja Huttunen, and Roman Yangarber. 2002. Real-time event extraction for infectious disease outbreaks. In *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 366–369.

Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE* 7, 4 (2008), 12–18.

Michael Kehoe, Michael Cosgrove, SD Gennaro, Colin Harrison, Wim Harthoorn, John Hogan, John Meegan, Pam Nesbitt, and Christina Peters. 2011. Smarter cities series: a foundation for understanding IBM smarter cities. *An IBM Redguide publication* (2011).

Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Giridhar Kumaran and James Allan. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 297–304.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a Social Network or a News Media?. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. ACM, New York, NY, USA, 591–600. DOI:http://dx.doi.org/10.1145/1772690.1772751

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[19] http://conceptnet5.media.mit.edu/

Vasileios Lampos and Nello Cristianini. 2012. Nowcasting events from the social web with statistical learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 4 (2012), 72.

Greg Lindsay. 2010. Cisco's big bet on New Songdo: creating cities from scratch. *Fast Company* 1 (2010).

Mingrong Liu, Yicen Liu, Liang Xiang, Xing Chen, and Qing Yang. 2008. Extracting key entities and significant events from online daily news. In *Intelligent Data Engineering and Automated Learning–IDEAL 2008*. Springer, 201–209.

Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. 2010. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 211–224.

Lluís Màrquez, Xavier Carreras, Kenneth C Litkowski, and Suzanne Stevenson. 2008. Semantic role labeling: an introduction to the special issue. *Computational linguistics* 34, 2 (2008), 145–159.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. (2002).

Dunja Mladenić and Alexandra Moraru. 2012. Complex event processing and data mining for smart cities. (2012).

Meenakshi Nagarajan, Karthik Gomadam, Amit P Sheth, Ajith Ranabahu, Raghava Mutharaju, and Ashutosh Jadhav. 2009. Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences. In *Web Information Systems Engineering-WISE 2009*. Springer, 539–553.

Milind Naphade, Guruduth Banavar, Colin Harrison, Jurij Paraszczak, and Robert Morris. 2011. Smarter cities and their innovation challenges. *Computer* 44, 6 (2011), 32–39.

Martina Naughton, Nicholas Kushmerick, and Joseph Carthy. 2006. Event extraction from heterogeneous news sources. In *Proceedings of the AAAI Workshop Event Extraction and Synthesis*. 1–6.

Masayuki Okamoto and Masaaki Kikuchi. 2009. Discovering Volatile Events in Your Neighborhood: Local-Area Topic Extraction from Blog Entries. In *Information Retrieval Technology*. Springer, 181–192.

John Pucher, Nisha Korattyswaroopam, and Neenu Ittyerah. 2004. The crisis of public transport in India: overwhelming needs but limited resources. *Journal of Public Transportation* 7 (2004), 95–113.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, 157–176.

Alan Ritter, Oren Etzioni, Sam Clark, and others. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1104–1112.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, 851–860.

Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. 2009. Event Detection and Tracking in Social Streams.. In *ICWSM*.

Amit Sheth. 2009. Citizen sensing, social signals, and enriching human experience. *Internet Computing, IEEE* 13, 4 (2009), 87–92.

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. 2008. Real-time news event extraction for global crisis monitoring. In *Natural Language and Information Systems*. Springer, 207–218.

Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. 2012. Automatic crime prediction using events extracted from twitter posts. In *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, 231–238.