Structured Information Extraction from Natural Disaster Events on Twitter

Sandeep Panem
IIIT, Hyderabad, India
sandeep.panem@research.iiit.ac.in

Manish Gupta*
Microsoft, Hyderabad, India
gmanish@microsoft.com

Vasudeva Varma
IIIT, Hyderabad, India
vv@iiit.ac.in

ABSTRACT

As soon as natural disaster events happen, users are eager to know more about them. However, search engines currently provide a ten blue links interface for queries related to such events. Relevance of results for such queries can be significantly improved if users are shown a structured summary of the fresh events related to such queries. This would not just reduce the number of user clicks to get the relevant information but would also help users get updated with more fine grained attribute-level information.

Twitter is a great source that can be exploited for obtaining such fine-grained structured information for fresh natural disaster events. Such events are often reported on Twitter much earlier than on other news media. However, extracting such structured information from tweets is challenging because: 1. tweets are noisy and ambiguous; 2. there is no well defined schema for various types of natural disaster events; 3. it is not trivial to extract attribute-value pairs and facts from unstructured text; and 4. it is difficult to find good mappings between extracted attributes and attributes in the event schema.

We propose algorithms to extract attribute-value pairs, and also devise novel mechanisms to map such pairs to manually generated schemas for natural disaster events. Besides the tweet text, we also leverage text from URL links in the tweets to fill such schemas. Our schemas are temporal in nature and the values are updated whenever fresh information flows in from human sensors on Twitter. Evaluation on ∼58000 tweets for 20 events shows that our system can fill such event schemas with an F1 of ∼0.6.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Data Mining*; H.3.4 [**Information Storage and Retrieval**]; H.4.0 [**Information Systems Applications**]: General

General Terms

Algorithms, Design, Experimentation

Keywords

Natural Disaster Events, Structured Event Mining, Twitter, Event Infoboxes, Attribute-Value Extraction, Fact Triplet Extraction, Natural Calamities

*The author is also affiliated with IIIT-Hyderabad

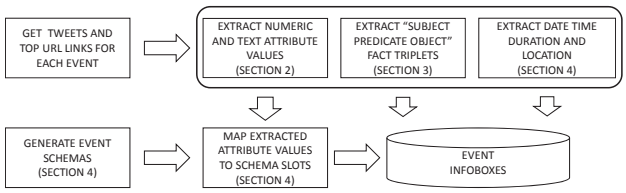

Figure 1: System Diagram

1. INTRODUCTION

As soon as natural disaster events happen, users are eager to know more about them. Often times they look out for related facts. For example, what is the severity of the storm or the magnitude of the earthquake. Searchers are also interested in knowing about the damage caused by these natural calamities, e.g., number of people dead or number of homes destroyed. In 2012, there were 905 natural catastrophes worldwide, 93% of which were weather-related disasters. Overall costs were US $170 billion and insured losses $70 billion. 45% were meteorological (storms), 36% were hydrological (floods), 12% were climatological (heat waves, cold waves, droughts, wildfires) and 7% were geophysical events (earthquakes and volcanic eruptions). Between 1980 and 2011 geophysical events accounted for 14% of all natural catastrophes [13].

Search engines currently provide a ten blue links interface for queries related to such events. Relevance of results for such queries can be significantly improved if users are shown a structured summary of the fresh event related to the query. This would not just reduce the number of user clicks to get the relevant information but would also help users get updated with more fine grained attribute-level information. This is especially useful in cases of disaster events because it can help users obtain information quickly and thereby help in reducing anxiety levels.

To show a structured summary of such events, one needs to obtain fresh information about such events. News media, blogs, Twitter are all good sources of information. However, for natural disasters, information is found fastest on Twitter [26]. But Twitter text is highly unstructured and noisy. Hence, extracting useful information from tweets is challenging. First, one needs to identify all tweets related to the query disaster event. Next, these tweets need to be linguistically analyzed to extract useful structured information. How to obtain semantic attribute-value pairs and facts from tweets? Besides that, there is no standard schema available for such events. How to generate such a schema? Given a schema and extracted structured information, how to map attributes from

extracted information to standard attributes in event schemas?

In this paper, we deal with a few of the above challenges as follows. We use Stanford Typed Dependencies [6] and the CMU Tweet POS Tagger [8] for linguistic parsing of the tweet text. Next, we design novel algorithms for extraction of both numeric as well as textual attribute-value pairs from tweets. We also provide a novel algorithm for extracting fact triplets from tweets. We generate schemas for five different event types by leveraging Wikipedia Infoboxes along with some manual efforts. Finally, we present a novel algorithm to map extracted information to standard structured fields in the event schemas. To the best of our knowledge, the proposed system is the first to focus on extraction of structured event Infoboxes from Twitter for natural calamity events. Figure 1 shows the system diagram for the proposed system.

In short, we make the following contributions in this paper.

- We propose the problem of automatically generating structured Infoboxes for events from social media.
- Specifically, we consider natural disaster events and propose novel algorithms for extracting attribute-value pairs and facts.
- Through experiments on ∼58000 tweets related to natural disaster events of five different types, we show the effectiveness of the proposed system.

This paper is organized as follows. In Section 2, we present an introduction to Stanford dependencies and then present our novel algorithms for extracting both numeric and textual attribute-value pairs. In Section 3, we present a novel algorithm for extracting fact triplets. In Section 4, we first discuss event schema generation and then provide a novel mechanism to map attribute-value pairs extracted in Section 2 to event schemas. We present results of our experiments on 20 events in Section 5. In Section 6, we discuss related work in the areas of structured extraction from free text on the web as well as on Twitter. Finally, we conclude with a summary in Section 7.

2. EXTRACTION OF ATTRIBUTE-VALUE PAIRS

We use linguistic tools like the Stanford Typed Dependencies [6] and the CMU Tweet POS Tagger [8] to understand the semantics of the tweets. We remove user mentions, URL links and retweet symbols from the tweet before performing linguistic analysis.

The aim of this section is to obtain attribute-value pairs from tweets. When the units of the values are also mentioned next to the value, we extract the units too. In this section, we first provide a basic introduction to Stanford typed dependencies. After that we propose two novel algorithms: one for extracting numeric attribute-value pairs from tweets and the other for extracting textual attribute-value pairs.

2.1 Basic Introduction to Stanford Dependencies

The Stanford typed dependencies representation [6] was designed to provide a simple description of the grammatical relationships in a sentence that can be used to extract textual relations. The representation contains approximately 50 grammatical relations. For example, consider the tweet "Arizona struggles to contain blaze: Conflagration engulfs 110,000 acres... http://bit.ly/RpMYv4". The dependencies obtained after parsing the cleaned tweet are as follows: "root(ROOT-0, struggles-2); nsubj(struggles-2, Arizona-1); aux(contain-4, to-3); xcomp(struggles-2, contain-4); dobj(contain-4, blaze-5); nsubj(engulfs-8, Conflagration-7); parataxis(struggles-2, engulfs-8); num(acres-10, 110,000-9); dobj(engulfs-8, acres-10); prep_of(acres-10, land-12)". Here, nsubj, amod, nn, etc. are all dependencies.

The dependencies are all binary relations: a grammatical relation holds between a governor (also known as a regent or a head) and a dependent. For example, for the dependency "nsubj(struggles-2, Arizona-1)", "struggles" is the governor, and "Arizona" is the dependent.

The following dependencies are important with respect to this paper and so we define them here.

- $root$: The root grammatical relation points to the root of the sentence.
- $nsubj$: A nominal subject is a noun phrase which is the syntactic subject of a clause.
- $dobj$: The direct object is the noun phrase which is the (accusative) object of the verb.
- $pobj$: The object of a preposition is the head of a noun phrase following the preposition, or the adverbs "here" and "there".
- nn: A noun compound modifier is any noun that serves to modify the head noun.
- $prep\_*$: A prepositional modifier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition.
- num: A numeric modifier of a noun is any number phrase that serves to modify the meaning of the noun with a quantity.
- $number$: An element of compound number is a part of a number phrase or currency amount.
- $amod$: An adjectival modifier of a noun phrase is any adjectival phrase that serves to modify the meaning of the noun phrase.
- dep: A dependency is labeled as dep when the system is unable to determine a more precise dependency relation between two words.

For detailed understanding of these dependencies we redirect the reader to [6].

2.2 Numeric Attribute-Value Extraction

Identifying numeric attribute-value pairs from tweets is challenging. Naïve approaches like considering the neighboring words close to numeric literals as attribute names do not always work. For example consider the tweet: "Death toll rises to 123 in Mexico following Tropical Storm Ingrid". Here the attribute is "Death toll" and the value is "123". We can observe that "123" cannot be linked to the previous word or the next word. It needs to be linked to the phrase which actually describes it by understanding the relation.

Our approach to numeric attribute-value extraction consists of the following sub-modules: splitting sentences into self-complete sub-units, handling of special cases of attribute-value mentions, extracting attribute-value pairs using dependencies, and extracting complete attribute names.

2.2.1 Splitting Sentences into Self-complete Sub-units

If the tweet has two or more sentences or multiple subjects then we split the tweet into separate sentences, each containing a single subject. Since each subject gives additional description about a particular attribute, so mapping a subject to its corresponding attribute

is critical. For example, for the tweet "Arizona struggles to contain blaze: Conflagration engulfs 110,000 acres... http://bit.ly/RpMYv4". There are two subjects, but the appropriate one for the attribute "land" is "nsubj(engulfs-8, Conflagration-7)". The splitting is done based on a set of delimiters, or by considering subtrees of the parse tree.

2.2.2 Handling of Special Cases of Attribute-Value Mentions

Here we discuss two interesting cases as follows.

Case 1 - Numeric Values are mentioned side-by-side: Consider the tweet: "#USGS M 1.9 - 4km N of Hydesville, California: Time 2014-07-03 02:31:00 UTC 2014-07-02 19:31:00 -07:00 at ep..". Here, the attributes "M", "N" and values "1.9", "4km" are side by side. We handle this case using the following simple rule: Given any word marked as "numeric", if the next word is tagged as a noun (common noun, or proper noun) then we consider the next word as the attribute else we consider the previous word as the attribute name.

Case 2 - Attribute-Value pairs mentioned in a Sequence: In some tweets, attribute-value pairs are listed as a sequence with some delimitier. Repeated occurrences of ("numeric", "noun") pairs helps us detect such cases and identify the attribute and its value appropriately. For example, consider the tweet "Mag: 3 - Depth: 116 km - UTC 8:07 AM - Tarapaca, Chile - EMSC". Here attributes "Mag", "Depth", "UTC" are extracted along with the values "3", "116km", "8:07 AM" respectively.

2.2.3 Extracting Attribute-Value Pairs using Dependencies

After handling the special cases mentioned above, the remaining sentence (after removing the already extracted attribute-value pairs) is analyzed to extract the dependencies. We propose the following rules to exploit these dependencies to extract attribute-value pairs, the subject and the object.

- The subject and the object are extracted from the dependent parts of the $nsubj$ and $dobj$ dependencies respectively.
- The (governor, dependent) pair of every num dependency provides an attribute-value pair.
- The (governor, dependent) pair of every nn dependency provides an attribute-value pair if the dependent contains digits.

2.2.4 Extracting Complete Attribute Names

Since the dependencies provide relationships between pairs of words only, one needs to combine a few dependencies to extract complete attribute names. The detailed pseudo-code for the algorithm is presented in Algorithm 1. We propose the following rules for exploiting various dependencies to obtain complete phrases. Let AV be the set of extracted attribute-value pairs so far, where $AV[i].a$ and $AV[i].v$ denote the attribute and the value part of the i^{th} attribute-value pair.

- If the dependent of an $nsubj$ matches $AV[i].a$, expand $AV[i].a$ to include the governor too. (Lines 4 to 6)
- If the dependent or the governor of an nn matches $AV[i].a$, expand $AV[i].a$. (Lines 7 to 11)
- If the governor of a $prep\_*$ matches $AV[i].a$, expand $AV[i].a$ to include the preposition and the dependent. (Lines 12 to 15)
- If the governor of any $nsubj$, nn or $prep\_*$ matches the subject, expand the subject to include the dependent of the dependency too. (Lines 16 to 17)

Algorithm 1 Extraction of Complete Attribute Names

Input: (1) Dependency list DL (2) Set of extracted attribute-value pairs AV
Output: Set of attribute-value pairs AV with complete attribute names

```
1:  for all d ∈ DL do
2:      if d.rel ∈ {nn, prep_*, nsubj} then
3:          for all av ∈ AV do
4:              if d.rel == nsubj then
5:                  if av.a contains d.dep then
6:                      av.a ← d.gov + d.dep
7:              else if d.rel == nn then
8:                  if av.a contains d.dep then
9:                      av.a ← d.gov
10:                 else if av.a contains d.gov then
11:                     av.a ← d.dep + d.gov
12:             else if d.rel == prep_* then
13:                 Let p be the preposition.
14:                 if av.a contains d.gov then
15:                     av.a ← d.dep + p + av.a
16:         if Subject == d.gov then
17:             Subject ← d.dep + d.gov
```

2.3 Textual Attribute-Value Extraction

Textual attribute-value pairs are ones in which the value is non-numeric text. Compared to numeric attribute-value pairs, it is more challenging to mine textual attribute-value pairs due to the lack of any numeric clues. For example, consider the tweet: "Hurricane Sandy cancels many flights at Orlando Airport". Here, "(Airport, Orlando)" and "(Hurricane, Sandy)" are the two attribute-value pairs.

Given a tweet, there are three ways to obtain attribute-value pairs: (1) a central attribute-value pair related to the subject of the tweet ($CentralAV$), (2) attribute-value pairs related to the root word of the tweet ($RootAV$), and (3) attribute-value pairs connected to preposition dependencies ($PrepAV$).

For example, in the tweet above, "(Hurricane, Sandy)" is the central attribute-value pair. Given the dependencies for the tweet, we can leverage them to identify the above three types of textual attribute-value pairs.

Algorithm 2 explains about the process of extraction of the three types of textual attribute-value pairs. Lines 2 to 9 relate to extraction of $RootAV$. First the root word is extracted using the root dependency. Next, other words dependent on the root word are extracted. Further, the $dobj$, $pobj$ and $amod$ dependencies are exploited to obtain $RootAV$ where the root word is the attribute.

Lines 10 to 17 relate to the extraction of $CentralAV$. First the subject and the verb related to the attribute-value pair are extracted using the $nsubj$ dependency. Then the nn dependency is used to extract the $CentralAV$ pair where the subject forms the major part of the attribute name.

Lines 18 to 33 relate to the extraction of $PrepAV$. We refer to the (governor, dependent) pair of a $prep*$ dependency as a prepositional pair. Prepositional pairs could themselves be used as attribute-value pairs. First we obtain the prepositional pairs. Next, we use nn dependencies to enhance the dependents and governors of the prepositional pairs. If the nn dependency does not contain a noun and its dependent matches the governor of a prepositional pair, it is used to obtain an attribute-value pair.

Finally, the sets $RootAV$ and $PrepAV$, and $CentralAV$ are merged to get the set of all textual attribute-value pairs.

3. EXTRACTION OF FACT TRIPLETS

A fact triplet consists of three main parts: Subject, Predicate and the Object. For example consider the tweet "Volvo Ocean Race set for raft of changes to boats, teams and route in bid to appease sailors and sponsors via @Telgraph http://soc.li/AenbU9M". The extracted fact triplet for this tweet is "(Volvo Ocean Race : appease

Algorithm 2 Extraction of Textual Attribute-Value Pairs

Input: Dependency list DL
Output: Set of textual attribute-value pairs AV

1: Let $CentralAV$ be the central attribute-value pair, $RootAV$ be the set of attribute-value pairs related to the root word of the tweet, $PrepAV$ be the set of attribute-value pairs connected to preposition dependencies.
2: $rootWord \leftarrow d.dep$ where $d \in DL$ and $d == root$.
3: $rootWordValues \leftarrow \{d.dep | d.rel = dep$ and $d.gov = rootWord$ and $d \in DL\}$
4: **for all** $d \in DL$ **do**
5: **for all** $v \in rootWordValues$ **do**
6: **if** ($d.rel == dobj$ or $d.rel == pobj$) and ($d.gov == v$) **then**
7: $RootAV.Add((rootWord, v))$
8: **if** $d.rel == amod$ and $d.gov == rootWord$ **then**
9: $RootAV.Add((rootWord, d.dep))$
10: $subject \leftarrow d.dep$ where $d \in DL$ and $d == nsubj$.
11: $subjectVerb \leftarrow d.gov$ where $d \in DL$ and $d == nsubj$.
12: **for all** $d \in DL$ **do**
13: **if** $d.rel == nn$ and $d.gov == subject$ **then**
14: **if** {root,dobj} $\notin DL$ **then**
15: $CentralAV \leftarrow (d.dep + subject, subjectVerb)$
16: **else**
17: $CentralAV \leftarrow (subject, d.dep)$
18: Let PP be the set of all prepositional pairs.
19: **for all** $d \in DL$ **do**
20: **if** $d.rel == prep*$ **then**
21: $PP.Add(d.gov, d.dep)$
22: **for all** $d \in DL$ **do**
23: **for all** $pp \in PP$ **do**
24: **if** $d.rel == nn$ **then**
25: **if** $d.gov == pp.gov$ **then**
26: Update pp to $(d.dep + pp.gov, pp.dep)$.
27: **if** $d.gov == pp.dep$ **then**
28: **if** $d.gov$ or $d.dep$ is a noun **then**
29: Update pp to $(pp.gov, d.dep + pp.dep)$.
30: **else**
31: $PrepAV.Add(d.dep, pp.dep)$
32: Remove pp from PP.
33: $PrepAV \leftarrow PrepAV \cup PP$
34: $AV \leftarrow RootAV \cup \{CentralAV\} \cup PrepAV$

: sailors, sponsors)". In this section, we discuss a mechanism to extract such fact triplets using Stanford dependencies from any tweet. We extract fact triplets only from those tweets in which at least two entities occur.

Algorithm 3 presents our mechanism for extraction of fact triplets. First we obtain the subjects and objects in the tweet using various dependencies (Lines 1 and 2). Next, we obtain the root word and its index (Lines 3 and 4). Here index refers to the word position in the tweet. If there is no subject in the tweet, we use the root word to form a subject (Lines 6 to 7). Similarly, if there is no object in the tweet, we use the *dep* dependency to obtain an object (Lines 9 to 10). Further, we use various dependencies to expand the subjects and objects to get their complete forms (Lines 11 and 12). Subjects and objects are then matched using the verbs that appear with them in the dependencies (Lines 13 and 14). These verbs form the predicates, and are expanded using the prepositional modifiers (Line 15). Finally matching expanded (subject, predicate, object) are returned as fact triplets.

4. FILLING OF EVENT SCHEMAS

In this section, we discuss the challenges in generation of event schemas and how we created schemas for natural disaster events. Next, we discuss our algorithm to map extracted attributes to schema slots (or attributes).

4.1 Generation of Event Schemas

One can generate schemas for natural disaster events automatically as follows. For any event type (e.g., earthquakes), gather Infoboxes for a few events of that type from Wikipedia. Consider the top most frequent fields as attributes for the schema for that event

Algorithm 3 Extraction of Fact Triplets

Input: Dependency list DL
Output: Fact Triplets

1: $numSubjects \leftarrow$Number of $nsubj$ and $nsubjpass$.
2: $numObjects \leftarrow$Number of $dobj$ and $pobj$.
3: $rootWord \leftarrow d.dep$ where $d \in DL$ and $d == root$.
4: $rootIndex \leftarrow d.dep.index$ where $d \in DL$ and $d == root$.
5: Obtain list of subjects using $nsubj$ and $nsubjpass$.
6: **if** $numSubjects == 0$ **then**
7: Add $d.dep + d.gov$ as the subject where $d.gov == rootWord$ and ($d.rel == amod$ or $d.dep.index < rootIndex$).
8: Obtain list of objects using $dobj$ and $pobj$.
9: **if** $numObjects == 0$ **then**
10: Add $d.dep$ as the object where $d.rel == dep$.
11: Use dependents of nn and $amod$ with governor matching the object to expand the object.
12: Use dependents of dep and nn with governor matching the subject to expand the subject.
13: Match subjects with objects using matching verbs which are governor/dependent of dependencies containing the subject or object.
14: Set the matching verbs as predicates for (subject, object) pairs.
15: Expand predicates using prepositional modifiers.
16: Generate a fact triplet for every subject-predicate-object.

type. This technique does not work for events which do not have Infoboxes on Wikipedia (e.g., "Justin Bieber's birthday"). Also for general events, identifying the most relevant Infobox type is difficult. However, for natural disaster events, this is expected to work well.

But, there is a large mismatch between the Wikipedia Infobox attribute names and the attributes extracted from Twitter. Most of the attribute-value pairs extracted from Twitter events are quite new and are not present in Wikipedia Infoboxes. Hence, we had to resort to manual generation of event schemas with guidance from Wikipedia Infoboxes.

For each event type we use the tweets of one event in the training phase to manually learn the attributes for the event schema. Thus, we grow our schema beyond the one that could be obtained using Wikipedia Infoboxes.

Besides the attribute names, the event schemas contain more metadata information for every attribute described as follows. (1) We extract ranges for the Wikipedia Infobox attributes. For each event type, we determine the minimum and maximum value an attribute of that type can hold and define the range for each attribute. For attributes which are not present in Wikipedia, we manually assign attribute value ranges. Thus, each attribute of every event type is associated with a range of values it can take. (2) Next, we define the data type for each attribute of each event type. The event schema attributes can be of the following types: integer, float, string, date, time. (3) We also define the units for each event attribute. For example, for the attribute $wind\_speed$ the units will be 'mph','km/h' and for $funds\_donated$ would be '$' or 'euros' etc. (4) Finally, for each schema attribute for each event type, we identify a set of synonyms. For example, "total_cost, total_loss, money_loss" are synonyms for "total_economic_impact". Similarly, "body_wave_magnitude" is synonym for "mb" for the earthquake event schema.

4.2 Populating Values of Schema Attributes

As described in Section 2, for each attribute-value pair, we also extract the subject and object which are useful in mapping an extracted attribute-value pair to the event schema attribute. Often times, an attribute takes multiple values across various tweets for the event. We assign the most frequent value to each attribute. After that, we map the attribute-value pair to a schema attribute as described in the following. Algorithm 4 presents the pseudo-code for the mapping algorithm.

For each extracted attribute-value pair (a, v) and each schema

attribute s, we compute a match score (Lines 3 to 28). The match score depends on the following: (1) does v lie within the range of attribute s, (2) similarity between units of s and v, (3) similarity between units of s and subject of a, (4) similarity between units of s and object of a, (5) similarity between s and subject of a, (6) similarity between s and object of a, (7) similarity between s and a.

When computing similarity values, the score is incremented dependent on whether the similarity is greater than a threshold T or not. The similarity values lie between 0 and 1. When computing the score, various factors are given appropriate weights based on their importance. Based on this match score, we find the best schema attribute for each extracted attribute (Line 29). This also gives us a list of candidate extracted attributes for every schema attribute. We sort this list by score and map the schema attribute to the extracted attribute with the highest score (Line 34). Ties are resolved using frequency of occurrence of the candidate extracted attributes.

Some extracted attributes do not get mapped to any schema attribute. If they are frequent enough, we also list such attribute-value pairs in the structured event summary. Also, we use the SUTIME library [5] for extracting date, time and duration values from tweets. Based on the frequency across all tweets for an event, the final date, time and duration for the event occurrence are determined. Finally, fact triplets extracted using Algorithm 3 are also included in the event summary.

Algorithm 4 Mapping Attribute-Value Pairs to Event Schemas

Input: (1) Event Schema (2) Attribute-Value Pairs for an Event
Output: Mapping between Attribute-Value Pairs and Attributes in Event Schema, $Mapping$

1: Get the most frequent value for each extracted attribute.
2: $schemaAttributeToCandidates \leftarrow \phi$
3: **for all** $(a, v) \in AV$ **do**
4: $subject \leftarrow$ Subject related to (a, v).
5: $object \leftarrow$ Object related to (a, v).
6: $schemaAttributeToScore \leftarrow \phi$
7: **for all** attribute $s \in$ schema attributes **do**
8: $score \leftarrow 0$
9: $units \leftarrow$ units for attribute s
10: $range \leftarrow$ range for attribute s
11: $type \leftarrow$ type for attribute s
12: **if** v has type $type$ **then**
13: $score \leftarrow score + 1$
14: **if** v lies within the range $range$ **then**
15: $score \leftarrow score + 1$
16: **if** $sim(units, v) \geq T$ **then**
17: $score \leftarrow score + 2 \times sim(units, v)$
18: **if** $sim(units, subject) \geq T$ **then**
19: $score \leftarrow score + 2 \times sim(units, subject)$
20: **if** $sim(units, object) \geq T$ **then**
21: $score \leftarrow score + 2 \times sim(units, object)$
22: **if** $sim(a, s) \geq 0.8$ **then**
23: $score \leftarrow score + 3 \times sim(a, s)$
24: **if** $sim(s, subject) \geq T$ **then**
25: $score \leftarrow score + 2 \times sim(s, subject)$
26: **if** $sim(s, object) \geq T$ **then**
27: $score \leftarrow score + 2 \times sim(s, object)$
28: $schemaAttributeToScore.Add(s, score)$
29: Find the schema attribute K with max score S
30: **if** types of a and K do not match **then**
31: $S \leftarrow 0$
32: $schemaAttributeToCandidates.Add(K, (a, S))$
33: **for all** attribute $s \in$ schema attributes **do**
34: $f \leftarrow$ the candidate attribute with max score using $schemaAttributeToCandidates[s]$.
35: $Mapping.Add(s, f)$

5. EXPERIMENTS

In this section, we present details of our dataset, our experiment design, results of experiments conducted, and analysis of results.

5.1 Dataset

We selected five natural disaster event types: earthquakes, hurricanes (or typhoons), floods, wildfires and landslides. For each event type, we crawled tweets of 3–5 recent events listed as follows.

- Earthquakes: Chile Earthquake, Visayas Earthquake, Mexico Earthquake, Solomon Earthquake, Vizag Earthquake
- Hurricanes: Hurricane Sandy, Hurricane Amanda, Hurricane Ingrid Manuel, Typhoon Haiyan, Typhoon Phailin
- Floods: Balkan Floods, Serbia Floods, US Colorado Floods, Uttarakhand Floods
- Wildfires: California Wildfire, Alaska Wildfire, Arizona Wildfire
- Landslides: Washington Landslide, Zambales Landslide, Bolivia Landslide

We obtained related tweets using the Twitter search API. On an average the dataset consists of ∼3000 tweets per event. We use one event for each type to learn the event schema and then use that schema for all the events of the same type.

5.2 Accuracy of Filling the Event Schemas

For schema filling, we use threshold $T = 0.8$. Table 1 shows the precision, recall and F1 for the task of filling schemas for different events, except for the events used for learning the schema itself. As shown in the table, the event schemas contain around 30 attributes per event on an average. We measured the precision with which the proposed algorithms could fill the event schemas, and also the recall, i.e., the number of event attributes that could be filled.

Tweets may not always contain all information. We extract URLs mentioned in tweets for an event. We expand the shortened URLs and filter out links which relate to images, videos and Twitter or Facebook posts. For each event, we extract top 20 URLs and crawl the text of the URL links. We run the algorithms proposed in Sections 2 and 3 to extract attribute-value pairs and facts respectively on URL text too. Extracted attribute-value pairs from URLs are again mapped to event schemas.

We summarize the results shown in Table 1 across all events in Table 2 for all the three settings: "Only Web-links", "Only Tweets", and "Tweets + Web-links". The precision for "Only Web-links" is more because web text is more structured compared to the tweets. This results in accurate mapping of a value to a particular attribute. However, the recall is less as much of the information was not expressed in the top few Web-links, so less number of attribute-value pairs were discovered. The recall increased when both tweets and web-links were used to extract attribute-value pairs as expected. This is because some attributes were expressed in tweets and some were expressed in web-links, so in combination it resulted in more attribute-value pairs. Overall, we obtained best F1 with "Tweets + Web-links".

5.3 A Case Study: "Chile Earthquake"

Table 3 shows the attribute-value pairs extracted and mapped to the earthquake schema for the event "Chile Earthquake".

Note the variety of attributes that can be extracted from tweets. We could obtain magnitude of the earthquake on various scales including (1) local magnitude (ML), commonly referred to as "Richter

			Only Tweets			Only Web-links			Tweets + Web-links		
Event-Name	#Tweets	#Attributes	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Earthquake_Mexico	5675	35	0.961	0.714	0.819	1.0	0.2857	0.444	0.962	0.742	0.837
Earthquake_Solomon	976	35	0.823	0.4	0.538	0.615	0.228	0.332	0.833	0.428	0.565
Earthquake_Visayas	177	35	0.923	0.371	0.529	0.833	0.285	0.424	0.866	0.4	0.547
Earthquake_Vizag	199	35	0.8	0.228	0.354	1.0	0.142	0.248	0.818	0.257	0.391
Hurricane_Amanda	4390	42	0.833	0.476	0.605	1.0	0.166	0.284	0.869	0.50	0.634
Hurricane_IngridManuel	2253	42	0.857	0.285	0.427	0.875	0.333	0.482	0.863	0.452	0.593
Typhoon_Haiyan	5376	42	0.809	0.404	0.538	1.0	0.214	0.352	0.695	0.380	0.491
Typhoon_Phailin	345	42	0.818	0.261	0.395	0.909	0.238	0.377	0.846	0.261	0.398
Floods_Serbia	5835	35	0.75	0.428	0.544	0.8	0.228	0.354	0.75	0.428	0.544
Floods_USColorado	370	35	0.88	0.228	0.362	0.928	0.371	0.530	0.933	0.4	0.559
Floods_Uttarakhand	5115	35	0.764	0.371	0.499	0.916	0.314	0.467	0.947	0.514	0.666
Wildfire_Alaska	3118	29	1.0	0.448	0.618	0.833	0.517	0.638	0.894	0.551	0.681
Wildfire_Arizona	5765	29	0.875	0.482	0.621	0.866	0.448	0.590	0.944	0.586	0.723
Landslide_bolivia	721	19	0.88	0.421	0.569	0.8	0.21	0.332	0.9	0.473	0.620
Landslide_Zambales	31	19	0.80	0.21	0.332	1.0	0.421	0.592	1.0	0.526	0.689

Table 1: Accuracy of Filling the Event Schemas with Structured Information from Tweets, Web-links and Tweets + Web-links

	Avg. P	Avg. R	Avg. F1
Only Tweets	0.851	0.385	0.516
Only Web-links	0.891	0.293	0.429
Tweets + Web-links	0.874	0.460	0.595

Table 2: Average Precision (P), Recall (R), and F1 for the three Variations

Attribute	Value
areas_affected	chile iquique antofagasta
distance_miles	6.6
magnitude	5.0
mw (moment magnitude)	5.9
mb (body-wave magnitude)	4.7
ml (local magnitude)	4.0
death_toll	1,655
people_evacuated	300
missing_people	40k
date	2014-05-05
duration	1 minute (P1M)
time	05:00
tsunami_warning	3
direction@e	98km
direction@ne	47km
direction@n	73km
direction@se	34km
direction@sw	67km
direction@s	20.1km
direction@nw	19km
depth	10.0

Table 3: Event Schema filled with Attribute-Values for "Chile Earthquake"

magnitude," (2) body-wave magnitude (Mb), and (3) moment magnitude (Mw). We could also obtain drilled down numbers about people affected: people dead, people evaluated and people missing. We could also obtain severity of the tsunami warning and the impact distances in various directions. Showing such structured information for the query "Chile earthquake" would surely be better than what popular search engines show today.

5.4 Temporal Analysis of Attribute-Value Pairs

We also performed temporal analysis regarding how the event schemas get populated and how the attribute-value pairs evolve over time. Table 4 shows the variation in the values of attributes over time from five different events. Each line describes an attribute from one of the five events. The columns correspond to hours (H) or days (D) after the event. Each table cell represents the value of the attribute at that time point and the number of tweets from which that value was extracted. From the temporal analysis, we make the following observations.

- People talk more about attributes like number of people died, magnitude, direction, number of people affected compared to other attributes. Technical attributes like "mb, ml, etc." also appear on Twitter but they are usually put up by news agencies.
- Usually technical attributes like the magnitude, depth of the epicenter, etc. appear first on Twitter. After some time, when field analysis gets done, people start tweeting about the damage. This is when we observe attributes like people affected, schools affected, people injured getting populated.
- Attribute values that appear in the beginning are not very trustworthy. Initially people tweet various values for an attribute from the sources they have access to. Slowly over time the attribute values become stable. For example, as shown in Table 4, the value of the "magnitude" attribute fluctuates a lot in the initial hours and becomes stable only around the 11^{th} hour after the event.
- Some attributes are inherently temporal in nature. For example, as can be seen in Table 4, the value for the attribute '$people\_dead$' gradually increases. As more and more people are confirmed dead, this number keeps on increasing.

5.5 News vs. Tweets

We also performed an analysis of what attributes get filled using news URL text versus tweets. We observed that on Twitter news agencies often put out very technical information about events like body-wave magnitude, moment magnitude, etc. Indeed this information is not found on news webpages. We hypothesize that this is because although the news agencies have this information, reporters prefer to publish only non-technical information which appeals the mass. Hence, they refrain from publishing very technical

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	12-24 H	2-5 D	6-10 D
direction@w	84km 35	84km 2	84km 1	-	-	84km 1	-	-	-	-	-	-	84km 2		
direction@e	64km 2	-	-	-	-	-	-	-	-	-	-	-	-	-	60km 16
depth	-	-	-	-	102km 12	-	-	-	-	-	-	-	-	-	-
mb	8.0 175	8.0 114	-	8.0 266	8.0 107	7.9 63	8.0 25	7.9 66	7.9 46	-	-	-	-	-	-
magnitude	6.2 189	5.1 28	5.5 86	5.3 25	4.6 20	6.9 2	6.3 2	5.0 28	4.9 55	7.2 421	7.2 421	-	-	-	-
people_affected	-	150 42	-	150 10	150 5	150 1	-	150 2	-	-	-	-	-	-	-
direction@n	-	-	-	-	-	-	62km 2	-	-	-	-	-	-	-	-
people_dead	24 4	-	43 3	43 1	70 7	36 7	58 1	80 5	-	123 3	180 11	-	2,100 3	-	-
people_evacuated	-	-	2 1	-	-	-	-	-	-	-	-	-	-	115 1	250 6
people_injured	-	-	-	-	58 1	-	-	-	-	-	-	-	100 19	-	-
schools_affected	-	-	-	-	-	-	-	-	-	-	-	-	four 14	-	150 1

Table 4: Temporal Analysis for five different Events

information, but concentrate on location of the events, number of people dead or injured. These are the aspects of events that appeal the mass in general.

5.6 Fact Triplets

Along with the attribute-value pairs, we also display fact triplets as part of the event summary. Table 5 shows a set of few such triplets extracted for the event "Hurricane Sandy".

6. RELATED WORK

There has been a lot of work on extracting structured content from news articles, Wikipedia pages, queries and general web documents. Also there has been work on extracting structured content from tweets. While our work also leverages linguistic analysis similar in philosophy to other previous works, our work focuses on extracting attribute-value pairs from tweets and map them to standard event schemas which has not been done earlier.

6.1 Extracting Structured Content from the Web

Sarawagi [21] provides a great survey on automatic extraction of information from unstructured sources. We describe a few other works here. Nakashole et al. [16] developed a system 'PATTY' which is a large resource for textual patterns that denote binary relations between entities. Fader et al. [7] developed a system 'REVERB' which is based on syntactic and lexical constraints on binary relations expressed by verbs from English sentences. Wu et al. [25] developed a system 'Kylin Ontology Generator' which extracts structured data from Wikipedia raw texts and builds an ontology by combining Wikipedia Infoboxes with Wordnet using statistical relational learning. Wikipedia text follows a good grammatical structure and is easy to extract structured info than compared with noisy and short text information contained in tweets. Rusu et al. [19] extracted triplets from general English sentences using Tree-bank and link grammar parsers. They have proposed an algorithm to extract subject, object and predicate for grammatically correct English sentences, which fail to work on tweets. Bellare et al. [4] proposed a lightly supervised method to extract attributes from different entities from natural language corpus like Web. They trained on a fixed entities like company, country, etc. and extracted attributes pertaining to that entities itself. Reisinger et al. [17] proposed an approach 'Bootstrapped Web-Search Extraction' to extract class attributes simultaneously from Web documents and query logs. But they do not map the attributes to an existing event schema, neither do they detect the event type. Wong et al. [24] proposed a methodology for extracting attribute-value pairs from web pages. In the first phase they generate candidate attributes and in second phase they do candidate filtering. Even in this case the attribute-values were not mapped to any event schema. Enrique et al. [2] proposed a hierarchical topic model for automatically identifying syntactical and lexical patterns to detect relations from Web text. They leverage distant supervision using relations from knowledge base Freebase. Enrique et al. [3] presented a method for increasing the quality of automatically extracted instance attributes by exploiting weakly supervised and unsupervised instance relatedness data. The method organizes text derived data into graph and propogates attributes among related instances through random walks over the graph. Lee et al. [12] extract attributes for concepts and entities by integrating concept and instance based patterns into probabilistic typicality scores that scale to broad concept space. All these methods have been shown to perform well on Web text. But tweets are more noisy and much shorter than sentences on the Web, and hence these methods do not usually perform well on tweets.

6.2 Extracting Structured Content from Tweets

There is a large body of work on event detection from Twitter [9, 10, 15, 18]. But our proposed system goes much beyond simple event detection. Our focus is to display as much structured event content as possible. The closest to our work is the work by Marcus et al. [14] which focuses on providing an SQL-like interface to Twitter API. While their focus is on querying the event stream database, our focus is on populating such a database with highly structured data.

TwiCal [18] extracts a 4-tuple representation of events which in-

Tweet	Fact Triplet
Nearly 69,000 Con Edison customers in NYC and Westchester County have already lost power	(Edison customers, lost, power)
Waves begin crashing over Chelsea Piers in Manhattan	(Waves, crashing, Chelsea Piers)
Hurricane Sandy Hits NJ	(Hurricane Sandy, Hits, NJ)
Hurricane Sandy halts international #flights, too - from @LAtimes : http://tinyurl.com/93k6c3o	(Hurricane Sandy, halts, international flights)
Florida Crews Sent North To Assist Hurricane Sandy Victims http://cbsloc.al/RjDMSA	(Florida Crews, Assist, Hurricane Sandy Victims)

Table 5: Fact Triplets for the Event "Hurricane Sandy"

cludes a named entity, event phrase, calendar date, and event type. It represents the events in a calendar format. It does not leverage the global context of tweets and does not focus on extracting attribute-values and facts from tweets but rather focuses on extracting 4-tuple representation. Abel et al. [1] inferred facets and facet values by enriching the semantics of tweets. They tried to tag person, location and organization etc. in the tweets which help in faceted search. Their system does not focus on extracting attribute-value pairs and facts triplets which is the main contribution of this paper.

Twitter has also been used to detect and track natural disaster events: locating wildfires [23], hurricanes, floods [22], earthquakes [20, 11] and tornados. In this paper, we use such detected events as input and extract a drilled down structured summary.

7. CONCLUSIONS

In this paper, we studied the problem of extracting structured information for natural disaster events from Twitter. Specifically we focused on extracting attribute-value pairs and fact triplets. We solved this problem by first performing linguistic analysis like part of speech tagging and dependency parsing. After that we proposed three novel algorithms for numeric attribute-value extraction, textual attribute-value extraction, and fact triplet extraction. We also proposed an algorithm to map the extracted attributes to a schema for the corresponding event type. Experiments on $\sim$58000 tweets for 20 events show the effectiveness of the proposed approach. Such a structured event summary can significantly improve the relevance of the displayed results by providing key information about the event to the user without any extra clicks. In the future, we would like to generalize this approach to other events on Twitter.

8. REFERENCES

[1] F. Abel, I. Celik, G.-J. Houben, and P. Siehndel. Leveraging the Semantics of Tweets for Adaptive Faceted Search on Twitter. In *International Semantic Web Conference*, pages 1–17, 2011.

[2] E. Alfonseca, K. Filippova, J.-Y. Delort, and G. Garrido. Pattern Learning for Relation Extraction with a Hierarchical Topic Model. In *Proc. of the* 50^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 54–59. Association for Computational Linguistics, 2012.

[3] E. Alfonseca, M. Pasca, and E. Robledo-Arnuncio. Acquisition of Instance Attributes via Labeled and Related Instances. In *Proc. of the* 33^{rd} *Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 58–65. ACM, 2010.

[4] K. Bellare, P. P. Talukdar, G. Kumaran, F. Pereira, M. Liberman, A. McCallum, and M. Dredze. Lightly-Supervised Attribute Extraction. In *Proc. of the Neural Information Processing Systems (NIPS) 2007 Workshop on Machine Learning for Web Search*, 2007.

[5] A. X. Chang and C. D. Manning. SUTime: A Library for Recognizing and Normalizing Time Expressions. In *Proc. of the 2012 Intl. Conf. on Language Resources and Evaluation (LREC)*, pages 3735–3740, 2012.

[6] M.-C. de Marneffe and C. D. Manning. The Stanford Typed Dependencies Representation. In *Proc. of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, 2008.

[7] A. Fader, S. Soderland, and O. Etzioni. Identifying Relations for Open Information Extraction. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545. Association for Computational Linguistics, 2011.

[8] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proc. of the* 49^{th} *Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, pages 42–47, 2011.

[9] M. Gupta, R. Li, and K. Chang. Tutorial: Towards a Social Media Analytics Platform: Event Detection and User Profiling for Microblogs. In *Proc. of the* 23^{rd} *Intl. Conf. on World Wide Web (WWW)*, 2014.

[10] T. Hua, F. Chen, L. Zhao, C.-T. Lu, and N. Ramakrishnan. STED: Semi-supervised Targeted-interest Event Detection in Twitter. In *Proc. of the 19th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 1466–1469, 2013.

[11] K. Kireyev, L. Palen, and K. Anderson. Applications of Topics Models to Analysis of Disaster-Related Twitter Data. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, Dec 2009.

[12] T. Lee, Z. Wang, H. Wang, and S. won Hwang. Attribute Extraction and Scoring: A Probabilistic Approach. In *Proc. of the 2013 IEEE* 29^{th} *Intl. Conf. on Data Engineering (ICDE)*, pages 194–205, 2013.

[13] P. Löw. Natural Catastrophes in 2012 Dominated by U.S. Weather Extremes. http://www.worldwatch.org/natural-catastrophes-2012-dominated-us-weather-extremes-0, 2013.

[14] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Processing and Visualizing the Data in Tweets. *ACM SIGMOD Record*, 40(4):21–27, 2012.

[15] M. Mathioudakis and N. Koudas. Twittermonitor: Trend Detection over the Twitter Stream. In *Proc. of the 2010 ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD)*, pages 1155–1158, 2010.

[16] N. Nakashole, G. Weikum, and F. Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proc. of the 2012 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1135–1145. Association for Computational Linguistics, 2012.

[17] J. Reisinger and M. Pasca. Low-Cost Supervision for Multiple-Source Attribute Extraction. In *Proc. of the 2009 Conf. on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 382–393, 2009.

[18] A. Ritter, O. Etzioni, S. Clark, et al. Open Domain Event Extraction from Twitter. In *Proc. of the 18th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 1104–1112, 2012.

[19] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic. Triplet Extraction From Sentences. In *Proc. of the* 10^{th} *Intl. Multiconf. "Information Society - IS 2007"*, volume A, pages 218–222, 2007.

[20] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake Shakes Twitter Users: Real-Time Event Detection by Social Sensors. In *Intl. World Wide Web Conference (WWW)*, pages 851–860, 2010.

[21] S. Sarawagi. Information Extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[22] K. Starbird, L. Palen, A. L. Hughes, and S. Vieweg. Chatter on the Red: What Hazards Threat Reveals about the Social Life of Microblogged Information. In *Computer Supported Cooperative Work (CSCW)*, pages 241–250, 2010.

[23] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging During Two Natural Hazards Events: What Twitter may Contribute to Situational Awareness. In *Intl. Conf. on Human Factors in Computing Systems (CHI)*, pages 1079–1088, 2010.

[24] Y. W. Wong, D. Widdows, T. Lokovic, and K. Nigam. Scalable Attribute-Value Extraction from Semi-structured Text. In *Proc. of the 2009 IEEE Intl. Conf. on Data Mining (ICDM) Workshops*, pages 302–307, 2009.

[25] F. Wu and D. S. Weld. Automatically Refining the Wikipedia Infobox Ontology. In *Proc. of the* 17^{th} *Intl. Conf. on World Wide Web (WWW)*, pages 635–644. ACM, 2008.

[26] J. Yang and J. Leskovec. Patterns of Temporal Variation in Online Media. In *Proc. of the* 4^{th} *ACM Intl. Conf. on Web Search and Data Mining (WSDM)*, pages 177–186. ACM, 2011.