ClaimFinder: A Framework for Identifying Claims in Microblogs

Wee Yong Lim Mong Li Lee Wynne Hsu

Department of Computer Science
National University of Singapore
a0109697@u.nus, {leeml,whsu}@comp.nus.edu.sg

ABSTRACT

Twitter is a microblogging platform that allows users to post public short messages. Posts shared by users pertaining to real-world events or themes can provide a rich "on-the-ground" live update of the events for the benefit of everyone. Unfortunately, the posted information may not be all credible and rumours can spread over this platform. Existing credibility assessment work have focused on identifying features for discriminating the credibility of messages at the tweet level. However, they do not handle tweets that contain multiple pieces of information, each of which may have different level of credibility. In this work, we introduce the notion of a claim based on subject and predicate terms, and propose a framework to identify claims from a corpus of tweets related to some major event or theme. Specifically, we draw upon work done in open information extraction to extract from tweets, tuples that comprises of subjects and their predicate. Then we cluster these tuples to identify claims such that each claim refers to only one aspect of the event. Tweets corresponding to the tuples in each cluster serve as evidence supporting subsequent credibility assessment task. Extensive experiments on two real world datasets shows the effectiveness of the proposed approach in identifying claims.

1. INTRODUCTION

Communications over the web have increasingly become user-driven where there exist multiple platforms for users to post their messages that can be seen by the general public. Unfortunately, there is little or no mechanisms to ensure the credibility of the posted messages, unlike traditional news media. Take the popular microblogging platform Twitter as an example, where users can freely post or re-post any short messages, known as tweets, from their mobile accounts. Such a platform allows for the fast dissemination of first hand and repeated information. When a major event occurs, many tweets are generated or re-tweeted containing messages that may be true, false or speculative.

Published as part of the #Microposts2016 Workshop proceedings, available online as CEUR Vol-1691 (http://ceur-ws.org/Vol-1691)

#Microposts2016, Apr 11th, 2016, Montréal, Canada.

In fact, our observation of collected tweets related to major events indicate that a majority of tweets were forwarded (re-tweeted) by multiple users with little or no changes to the content of the message. Considering the minimal changes by the users, the primary motivation of these users stem from their desire to disseminate the information in the tweet. Such dissemination of information would indeed serve a social utility if the information is true, but would otherwise be detrimental if the information is false or even speculative.

Research in information credibility has been gaining momentum in recent years [4, 5, 18, 10]. Figure 1 shows the steps involved in a credibility assessment framework. Collecting a set of tweets related to a major event can be done manually using keywords relevant to natural disaster, terrorist or shooting incident events [10], or automatically via some event detection methods e.g. TwitterMonitor [12]. These tweets are then analyzed to identify topics for subsequent credibility classification [4, 5, 18]. Features used to help identify suspicious tweets include sentiment [15], location [22], message propagation characteristic [14] amongst others.

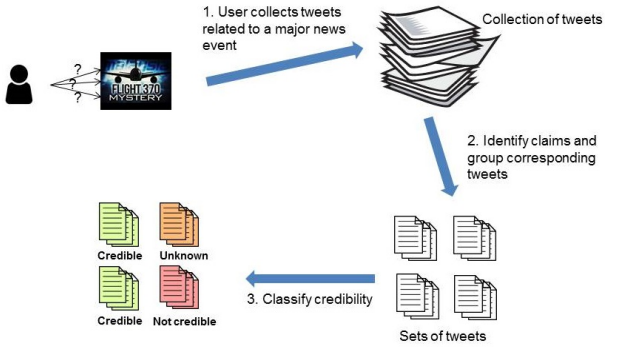

Figure 1: Credibility assessment framework involving tweet collection, claims identification and classification.

Methods to find topics in a corpus of tweets can be broadly divided into feature-based and topic modeling based approaches. The former extract features such as keywords from each tweet and clusters the tweets based on these features [2]. Each cluster of tweets defines a topic. For topic modeling based approaches, a topic is represented by a word distribution. The work in [23] observe that "a single tweet is usually about a single topic" and designed a TwitterLDA model where words in a tweet are chosen from a topic or the background noise words.

We observe that tweets typically contain multiple claims and advocate that current approaches which cluster tweets based on topics is too coarse-grained to identify all the claims in tweets. Take for example the following tweet on the Nashville flood:

> *"Middle TN (Nashville) has been hit by a terrible flood. Text 90999 to make $10 donation to the REDCROSS disaster relief. #nashvilleflood"*

This tweet has two claims: (1) Nashville has been hit by a flood, and (2) one can make a $10 donation by texting to 90999. It is important to identify both claims for subsequent credibility assessment. This is because while the first claim is likely to be true, the second claim appears highly suspicious. Existing credibility assessment work that utilizes tweet-level features will only give a single credibility score to this tweet and does not differentiate the two claims.

In this work, we formalize the concept of a *"claim"* in a corpus of tweets related to some major event. Our goal is to design a framework to identify the set of claims such that each claim refers to only one aspect of the event. Subsequently, the credibility of these claims can be verified against official sources. Note that the credibility assessment task is beyond the scope of this work.

We draw upon work done in the field of Open Information Extraction (IE) to extract entities in the tweets and the relationships between these entities. Then we construct tuples comprising of $<subject, predicate>$ from these entities/relationships. Finally, we cluster the tuples to form claims. The tweets that correspond to the cluster of tuples can be regarded as evidence supporting any subsequent credibility classification task. Extensive experiments on two real-world datasets of tweets demonstrate the effectiveness of our proposed approach to identify meaningful claims.

The paper is organized as follows. Section 2 defines the problem. Section 3 describes the proposed approach, and Section 4 gives an incremental method to identify claims. We present experiment results in Sections 5, followed by related work in Section 6 and conclude in Section 7.

2. PROBLEM DEFINITION

The objective of this work is to identify claims by grouping the tweets related to some major event such that tweets in each group refer to the same claim, of which can be true, false, speculative, conversational or simply spam in nature. We introduce the concept of a claim as follows:

Definition 1. A **claim** is the assertion of a subject and the corresponding predicate expression for the subject. It has the structure (S, P), where S is the set of words that refer to the same subject, P is the set of words that express the same predicate on S.

The set of words that refer to the same subject/predicate is very much context dependent. For example, in a corpus of tweets on the missing flight MH370 incident, the words "plane" and "MH370 aircraft" are likely to reflect the same subject whilst this may not be true in other context involving multiple planes such as news reports on manoeuvres between military planes[1]. Here, we assume that the major event provides the context for the claims and we would want to identify the claims within the event.

Since we do not assume that a tweet contains only one claim, we use an Open Information Extraction (OpenIE) tool [6] to extract from each tweet, zero or more triples of the form (E_1, R, E_2), where E_1 and E_2 are each a set of words refering to real world entities, while R is a set of words describing the relationship between the entities E_1 and E_2. Each triple is mapped to a subject-predicate tuple that has a structure similar to a claim, that is, $<S, P>$ where $S = E_1 \cup E_2$ and $P = R$. Thus, a tweet is associated with a set of subject-predicate tuples $\{t_1, t_2, ...\}$.

Problem Statement. Let $\mathcal{D}$ be a corpus of tweets related to a major event, and the i^{th} tweet in $\mathcal{D}$ is mapped to a set of tuples $\{t_{i1}, t_{i2}, ...\}$, $1 \leq i \leq |\mathcal{D}|$. Let $\mathcal{T}$ be the set of subject-predicate tuples obtained from all the tweets in $\mathcal{D}$. The goal is to obtain a partitioning $\mathcal{C}$ of the tuples in $\mathcal{T}$ such that $\mathcal{C}$ identifies the most number of claims in $\mathcal{D}$.

By partitioning the tuples, we obtain a soft clustering of the corresponding tweets since a tweet can contain more than a claim. The tweets that correspond to the tuples in each cluster provide evidence for the credibility assessment of the claim.

Example. To provide an intuition of the tuple clustering and claim identification process, Table 1 shows the OpenIE triples and the subject-predicate tuples obtained for 3 tweets. To simplify discussion, let us cluster these tuples based on the similarity of their subject words. For each cluster, we construct a claim by taking the union of the words in S and P respectively. Table 2 shows the clusters obtained and the corresponding claims. Note that our approach identifies the multiple claims contained in the tweets. For example, tweet 1 has two claims (c_1 and c_2), tweet 2 has two claims (c_2 and c_3), while tweet 3 has three claims (c_3, c_4 and c_5).

We will elaborate on our approach to identify claims in the next section.

3. CLAIMS IDENTIFICATION

Different from past tweets clustering work reviewed in Section 6, this work focuses on claim identification by clustering tuples mapped from OpenIE extractions of the tweets. We propose a 3-step *ClaimFinder* method (see Algorithm 1) which comprises of:

1. Preprocessing. We preprocess each tweet to remove known noise and tokenize the sentences prior to applying the OpenIE process.

2. Subject-predicate tuple extraction. We use the state-of-the-art OpenIE technique, ClausIE [6] to extract basic semantic units of information from the content of each tweet. Each extraction is mapped to a subject-predicate tuple $<S, P>$.

3. Clustering subject-predicate tuples. We define a similarity measure to compute the distance between the $<S, P>$ tuples. Then we can utilize methods such as agglomerative or spectral clustering [16] to cluster the tuples. Each cluster of tuples form a claim.

[1]http://edition.cnn.com/2014/08/22/world/asia/us-china-air-encounter/

	Tweet Content	Open IE Triples	Subject-Predicate Tuples
1	*MAS CEO confirms SAR ops and says airline is working to verify speculation that the mh370 may have landed in Nanning.*	(mas ceo, confirm, sar ops) (mh370, land, nanning)	<{mas,ceo,sar,ops}, {confirm}> <{mh370,nanning}, {land}>
2	*MH370 landing safely in Nanming is pure speculation. No distress signal or call was received at all*	(mh370, land, nanming) (distress signal call, receive)	<{mh370,nanming}, {land}> <{distress,signal,call}, {receive}>
3	*So you want me to believe that mh370 has crashed in water, Aussies found debris but still no signals captured*	(mh370, crash, water) (aussie, found, debris) (signal, capture)	<{mh370,water}, {crash}> <{aussie,debris}, {found}> <{signal}, {capture}>

Table 1: Subject-predicate tuples obtained from sample tweets.

	Cluster of tuples	Claim	Description
c_1	{ <{mas,ceo,sar,ops},{confirm}> }	({mas,ceo,sar,ops}, {confirm})	MAS CEO confirms SAR ops
c_2	{ <{mh370,nanning}, {land}>, <{mh370,nanming}, {land}> }	({mh370,nanning,nanming}, {land})	MH370 has landed in Nanning/Nanming
c_3	{ <{distress,signal,call}, {receive}>, <{signal}, {capture}> }	({distress,signal,call}, {receive,capture})	Signal received/captured
c_4	{ <{mh370,water}, {crash}> }	({mh370,water}, {crash})	MH370 crashed in water
c_5	{ <{aussie,debris}, {found}> }	({aussie,debris}, {found})	Australia found debris

Table 2: Claims obtained by clustering the tuples in Table 1.

Algorithm 1 *ClaimFinder*

Input: corpus $\mathcal{D}$ of tweets; number of clusters N
Output: set $\mathcal{C}$ of clusters of tuples
1: $\mathcal{T} = \emptyset$ // initialise set of tuples
2: **for** $twt \in \mathcal{D}$ **do**
3: $\mathcal{F} = OpenIE(Preprocess(twt))$
4: **for** $triple$ $(E_1, R, E_2) \in \mathcal{F}$ **do**
5: $\mathcal{T} \leftarrow \mathcal{T} \cup \{< (E_1 \cup E_2), R >\}$
6: **end for**
7: **end for**
8: $\mathcal{C} \leftarrow Cluster(\mathcal{T},\ N)$ // cluster the tuples
9: **return** $\mathcal{C}$

We describe each step in the following subsections.

3.1 Preprocessing

This phase corresponds to the function `Preprocess` in Algorithm 1 line 3. We preprocess each tweet via a series of data cleaning operations to reduce the noise that may affect subsequent OpenIE extraction. These include removing "rt" keywords (which indicate retweet message), URLs, user mentions, emoticons, colons, quote marks and hashtags' "#" signs. The tweet content is tokenized using the *twokenizer* tool designed for Twitter content [2]

3.2 Subject-Predicate Tuple Extraction

After preprocessing the tweets, each sentence is subsequently fed to an OpenIE tool to generate a list of relation triples. This step corresponds to the `OpenIE` function call in Algorithm 1 Line 3.

We chose to use ClausIE, the state-of-the-art OpenIE technique in this work. ClausIE takes as input each sentence in a tweet and identifies the entities E_1 and E_2, as well as their relationship R. The output is a triple (E_1, R, E_2). Then each triple (E_1, R, E_2) is mapped to a subject-predicate tuple (Algorithm 1 Lines 4-5).

[2] http://www.cs.cmu.edu/~ark/TweetNLP/

3.3 Clustering Subject-Predicate Tuples

At this juncture, we have obtained a set $\mathcal{T}$ of subject-predicate tuples from the original corpus of tweets $\mathcal{D}$. We use the popular Porter Stemmer [17] to stem the words in S and P, and filter the most frequent and infrequent words from the tuples.

We define the similarity between each pair of subject-predicate tuples $t_i = <S_i, P_i>$ and $t_j = <S_j, P_j>$ as follows:

$$similarity(t_i, t_j) = \left(w \cdot \frac{|S_i \cap S_j|}{|S_i \cup S_j|} + (1 - w) \cdot \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \right) \quad (1)$$

where w is a weight, $0 \leq w \leq 1$, which is empirically determined. Note that this similarity metric is based on the Jaccard index between sets from the respective tuples. This allows tuples comparison operations to be approximated and scaled up (see Section 4).

We can now apply existing clustering techniques to cluster the tuples in $\mathcal{T}$. Here, we choose two commonly used methods, namely, agglomerative or spectral clustering in our evaluation. Agglomerative clustering is a bottom-up hierarchical clustering approach, which initializes each subject-predicate tuple as a cluster by itself and successively merge the most similar pair of clusters at each step, till the specified number of clusters have been generated. Each cluster c is represented by a tuple t_c which is formed by taking the union of the respective S and P terms of the tuples in the cluster, that is,

$$t_c = < \{S_1 \cup \cdots \cup S_n\}, \{P_1 \cup \cdots \cup P_n\} > \ \forall < S_i, P_i > \in c$$

On the other hand, spectral clustering takes in a similarity matrix between all pairs of tuples and construct a Laplacian matrix. Then it performs an Eigen decomposition to obtain the top m eigenvectors, effectively reducing the dimensionality to m. Finally, we use k-means to cluster these eigenvectors to obtain the desired clusters.

The output of *ClaimFinder* is a set $\mathcal{C}$ of tuple clusters. This corresponds to Lines 8-9 in Algorithm 1. Each cluster corresponds to a claim. For each tuple in the cluster, we

can retrieve the corresponding tweets from which the tuple is derived. This forms a grouping of the tweets that can provide evidence to verify the credibility of the claim. Note that a tweet can belong to more than one grouping as it may contain multiple claims.

4. INCREMENTAL APPROACH

Considering the streaming nature of the tweets, especially for ongoing controversial major events rife with the propagation of rumours, we also propose an incremental approach to quickly identify claims from incoming tweets. Algorithm 2 gives the details of the $ClaimFinder^{INC}$ method.

Each incoming tweet is preprocessed and the tuples constructed as described in Sections 3.1 and 3.2. We create a set of empty buckets and assign a tuple to the bucket determined by a Locality Sensitive Hashing (LSH) function with MinHash (lines 2-6 of Algorithm 2). LSH allows us to quickly estimate the similarity between the set of subject and predicate words in the tuple and those in the bucket.

Let us first consider the subject term S in a tuple t. Since S is an arbitrary sized set of words, we choose its top n most frequent corpus words and apply m hash functions to this set of words S'. For each hash function h_i, we obtain the minimum hash value among the n words, denoted by $min(h_i(S'))$. With this, we form a vector

$$(\ min(h_1(S')), \ \cdots, \ min(h_m(S')) \ )$$

Similarly, we form a second vector based on the predicate term P as

$$(\ min(h_1(P')), \ \cdots, \ min(h_m(P')) \ )$$

where P' is the set of top n most frequent words in P. These two vectors form the MinHash signature of a tuple.

Next, we apply LSH on the MinHash signatures. Tuples with similar subject and predicate terms will be hashed to the same bucket. This is because if there exist some word that is present in both sets S_i and S_j, then $min(h(S_i)) = min(h(S_j))$. This eliminates the need for performing pairwise similarity computation between a tuple from an incoming tweet and each cluster. The corresponding tuples whose MinHash signatures have been mapped to the same bucket are subsequently merged into a cluster by taking the union of their S and P terms respectively.

Our incremental approach provides a mechanism to readjust the clusters should the size of a cluster increases beyond some threshold (lines 7-15 of Algorithm 2). This is achieved by treating the cluster as a mini-corpus to be further partitioned via standard clustering methods based on the similarity measure defined in Equation 1. After the adjustment, a merging operation may be applied to re-group clusters to specified number of clusters.

Algorithm 2 $ClaimFinder^{INC}$

Input: incoming tweet twt; split threshold $thres$
Output: set of buckets $B = \{b_1, b_2, ...\}$

```
1: F = OpenIE(Preprocess(twt))
2: for triple ∈ F do
3:     extract < S, P > tuple from triple
4:     i = LSH(MinHash(< S, P >))
5:     b_i ← b_i ∪ {< S, P >}
6: end for
7: if |b_i| ≥ thres then
8:     Split(b_i) into c_1 and c_2
9:     Let t_c1 and t_c2 be the representative tuples
           of c_1 and c_2 respectively
10:    Initialize b_i = ∅
11:    j = LSH(minHash(t_c1))
12:    b_j ← b_j ∪ {c_1}
13:    k = LSH(minHash(t_c2))
14:    b_k ← b_k ∪ {c_2}
15: end if
```

5. PERFORMANCE STUDIES

We implement the proposed algorithms $ClaimFinder$ and $ClaimFinder^{INC}$ in Python, and carry out experiments on a 2.3 GHz CPU with 8 GB RAM running on Ubuntu 14.04.

Our concept of claims is based on **subject-predicate tuples**. We also compare with the following representations:

- **tweet**: full text of the tweet
- **keywords**: a bag-of-words containing nouns, verbs, hashtags and cardinal numbers present in a tweet. The Stanford POS tagger using a trained model for tweets [7] is used to identify these keywords.
- **ngrams**: set of n consecutive words in the tweet, ignoring stop words. We use $n = 3$ as it has been shown to best capture the semantics in a tweet [1] generating 7,691 ngrams for the MH370 dataset and 3,998 ngrams for the Castillo dataset. Note that the similarity between a pair of ngrams is based on the Jaccard index (like Equation 1) rather than the fraction of overlapping tweets that contains both ngrams used in [1].

5.1 Datasets

We try to identify the claims in the two real world datasets:

- **MH370 Dataset.** We crawled and collected tweets on the crash of Malaysian Airline MH370 in 2014 for our experiments. This event involve the mysterious disappearance of a Boeing 777 plane en route from Kuala Lumpur to Beijing on 8 March 2014. Perceived mishandling of the public communication of the situation created an unfortunate conducive environment for the proliferation of various rumours related to MH370 with sustained public interest in the status of the flight and the cause of the disappearance. Such rumours range from the absurd such as alien abduction to more plausible ones such as the plane's safe landing in China during the early stage of the crisis. The location of the plane and cause of the disappearance remains unknown today. The tweet corpus was collected using the keyword "MH370" via Twitter's REST API. In total, 510,433 tweets from 8 March to 9 April were collected.

  We extracted a subset of tweets from the MH370 dataset using keywords of 6 known rumour and credible claims. Overall, 3,764 tweets have been identified and manually labeled with the corresponding claims. Table 3 gives the details. These claims form the ground truth.

- **Castillo Dataset.** We also obtain a subset of tweets with specific claims from 6 annotated topics in the Castillo dataset [5]. Table 4 shows 6 claims pertaining to President Obama. There are altogether 1,336

Claim	Description	#tweets	#unique tweets
M1	MH370 landed in Nanning	1393	271
M2	Pilot commit suicide	312	242
M3	Plane change course	203	78
M4	MH370 off course	1070	207
M5	Alien abduct MH370	538	398
M6	MH370 sighted in Maldives	248	50

Table 3: Groundtruth claims in MH370 dataset.

Claim	Description	#tweets	#unique tweets
T269	President Obama visiting the Gulf of Mexico	168	85
T876	President Obama sending troops to the US-Mexico border	466	283
T1494	President Obama praising/hailing lawmakers for a bill	48	39
T2370	President Obama signing the bill related to border security	212	104
T2384	President Obama supporting/endorsing building of a mosque near ground zero	373	233
T2499	President Obama is Muslim	69	67

Table 4: Groundtruth claims in Castillo dataset.

tweets, of which 811 are unique. Nomenclature of the claims follows that of the original annotated topics in [5], but with the prefix "T" instead of "TM" to indicate a filtered subset. We use these claims as ground truth.

5.2 Evaluation Metric

We evaluate the performance of the algorithms based on the proportion of claims they are able to identify. Let G be the set of ground truth claims and D_g be the set of tweets corresponding to a claim $g \in G$. The output of our algorithm is a set of tuple clusters, denoted C, where each cluster $c \in C$ refers to a claim. In other words, C is the set of claims identified by an algorithm. For each tuple cluster $c \in C$, we retrieve all the tweets associated with the tuples in c, denoted by D_c.

We define a match function to compute the fraction of tweets common in both D_c and D_g as follows:

$$match(c, g) = \frac{2 \times |D_c \cap D_g|}{|D_c| + |D_g|} \quad (2)$$

Note that when C and G have identical sets of tweets, we have $match(c, g) = 1$. On the other hand, when C and G have totally different sets of tweets, then $match(c, g) = 0$. Given a claim c, we say that c sufficiently covers a ground truth claim g if $match(c, g) \geq 0.8$.

We introduce a metric called *Coverage* to measure the ability of a method to identify claims as follows:

$$Coverage = \frac{|C_{match}|}{|G|} \quad (3)$$

where $C_{match} = \{g \in G \mid \exists\ c \in C, \quad match(c, g) \geq 0.8\}$

The set C_{match} contains the ground truth claims that have been covered by some cluster in C.

5.3 Performance of ClaimFinder

We have two versions of *ClaimFinder* depending on the clustering technique used. *ClaimFinder*(*Agglomerative*) implements the bottom-up agglomerative clustering in Line 8 of Algorithm 1, while *ClaimFinder*(*Spectral*) utilizes spectral clustering.

We run an initial set of experiments on each of the datasets to find the optimal settings for the parameters to achieve the best coverage results in Figures 2, 3 for *ClaimFinder*. These parameters are the input number of clusters N and the weight w in Equation 1 that controls the relative importance of the S and P terms when computing the similarity scores between tuples. For the MH370 dataset, we have $N = 18$ and $w = 0.6$, whereas for the Castillo dataset, $N = 6$ and $w = 0.8$. In addition, words less than 3% or more than 30% of the number of tweets are filtered prior to clustering the MH370 dataset. For the smaller Castillo dataset, a higher minimum threshold of 4% is used. These thresholds are determined empirically based on the frequencies of words in the groundtruth claims.

Figures 2 and 3 show the coverage for *ClaimFinder* using the different representations and clustering techniques. Spectral clustering gives better performance in both datasets, while keywords and ngrams generally gives lower coverage regardless of the clustering techniques employed.

We observe that the proposed subject-predicate tuples consistently identify more claims in both datasets and argue that its effectiveness indicates merit in discriminating the entity and relation terms using different weights for the different types of terms. This is not possible using keywords or ngrams. In addition, it is not effective to discriminate between the subject and object entities obtained directly from the OpenIE triple due to the interchangeability of the positions of the entities in the sentence (e.g.*plane abducted by alien* vs *alien abducts plane*).

5.3.1 Comparison with TwitterLDA

TwitterLDA [23] is designed for identifying topics in tweets. These topics are used to cluster the tweets for credibility assessment. We compare the performance of TwitterLDA using various tweet representations, namely, full tweet, keywords, subject-predicate tuples.

In addition to the original TwitterLDA model, we also experimented with its variants using author pooling and temporal pooling. For the MH370 dataset, there are 3,764 tweets from 3,557 authors. These tweets are posted across a period of 15 days and thus, a daily (24 hour) time frame is chosen for its temporal pooling. For the Castillo dataset, there are 1,336 tweets from 1,100 authors, posted between 1 May to 20 August 2010. The longer timeframe motivates the use of a weekly (7 days) time frame for temporal pooling.

Implementation for the TwitterLDA based approaches is based on the publicly available code[3], ran with default 100 iterations. TwitterLDA requires the number of topics as an input parameter. Our initial experiments show that the best performance is achieved when the number of topics is 12 for both datasets. We use this setting to obtain the coverage of the various TwitterLDA models.

[3]https://github.com/minghui/TwitterLDA

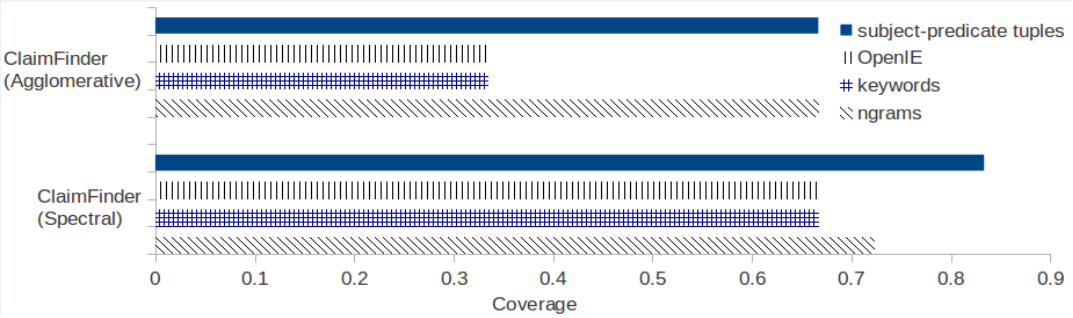

Figure 2: Performance of *ClaimFinder* (MH370).

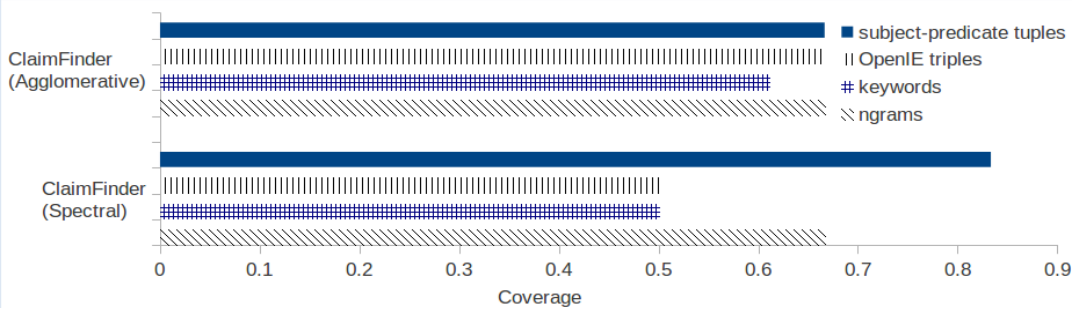

Figure 3: Performance of *ClaimFinder* (Castillo).

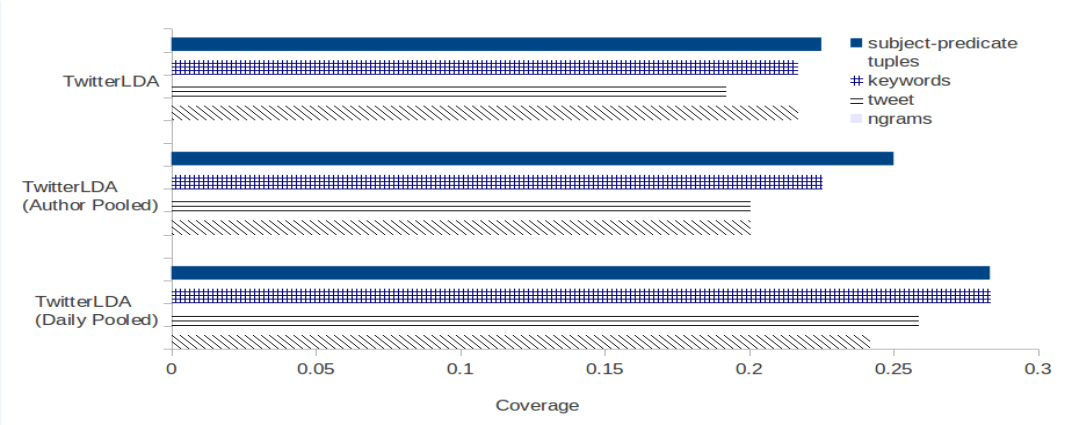

Figure 4: Performance of TwitterLDA (MH370).

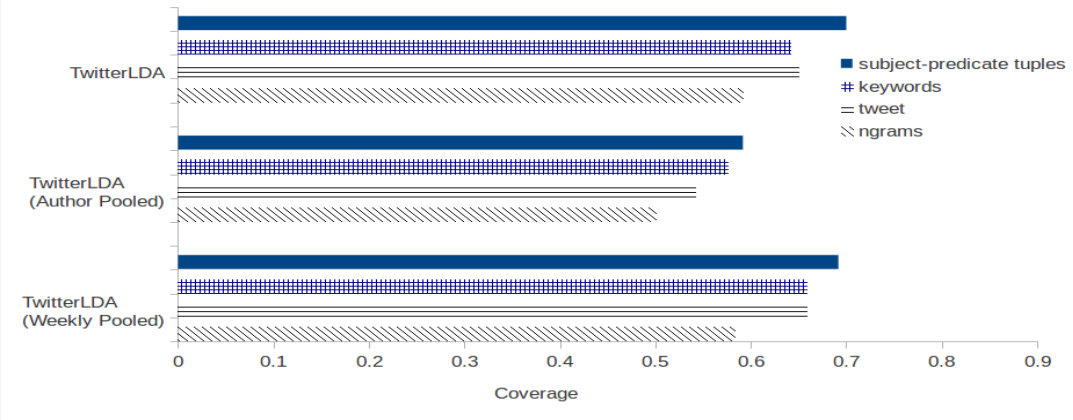

Figure 5: Performance of TwitterLDA (Castillo).

Figures 4 and 5 show the results. We observe that using the subject-predicate tuples representation always achieves the best coverage regardless of the TwitterLDA models used. This indicates that the subject-predicate tuples are able to capture the underlying semantics of a claim.

Using keywords generally yields better coverage compared to using ngrams or the full text of the tweet. Using the full tweet results in relatively bad coverage indicating that when there are multiple claims in a tweet, some of these claims may be missed.

Overall, the best performance is obtained when the proposed subject-predicate tuples is used in conjunction with TwitterLDA(Weekly Pooled). This is because there is a temporal correlation among the claims, that is, posts containing the same claims are likely to be sent within similar time windows. In contrast, TwitterLDA(Author Pooled) does not perform well due to the low tweet-to-author ratio for both datasets.

When we compare the coverage of the best performing variant of TwitterLDA, i.e. TwitterLDA(Weekly Pooled) in Figures 4 and 5, and the best performing *ClaimFinder* version, i.e. $ClaimFinder(Spectral)$ with subject-predicate tuples, we see that the latter significantly increases the number of claims identified in both datasets. We note that the MH370 dataset is noisier (more diverse set of words) than the Castillo dataset and believe that the larger improvement for the former is simply an indication of the weakness of TwitterLDA in dealing with the noise.

5.4 Effectiveness of ClaimFinder

As a case study on the effectiveness of the proposed claim identification approach, we retrieve the sets of subject-predicate tuples in the cluster that match some ground truth claim, as well as their corresponding tweets.

The identified claims and sample tweets obtained using $ClaimFinder(Spectral)$ are shown in Tables 5 and 6 for the MH370 and Castillo dataset respectively. We see that the tweets retrieved based on the clusters by *ClaimFinder* closely match the description of the ground truth claim, indicating that the subject-predicate tuples are able to capture the semantics of a claim.

5.5 Scalability of ClaimFinder^{INC}

Finally, we evaluate the scalability of the proposed incremental method $ClaimFinder^{INC}$ to identify claims.

We use 100 hash functions to generate the MinHash values, and spectral clustering for the splitting and merging operations. There are two parameters in $ClaimFinder^{INC}$, namely the number of LSH vectors and the threshold to split a cluster. We use 50 LSH vectors for both the MH370 and Castillo datasets. The split threshold is 10 and 30 tuples for MH370 and Castillo dataset respectively.

Figure 6 shows the runtime of $ClaimFinder^{INC}$ compared to *ClaimFinder* (in log scale) under spectral clustering and *ClaimFinder* under agglomerative clustering. We observe $ClaimFinder^{INC}$ is several orders of magnitude faster than both versions of *ClaimFinder* and remains scalable as the number of tweets increases.

Groundtruth claim	**Sample tweets**
M5 Alien abduct MH370	CNN has yet to rule out the theory that MH370 was abducted by aliens. Muldar, where are you?
	The #MH370 was abducted by aliens? How come?
	Rumors: Malaysia Airline MH370 Abducted by Aliens? - News - Bubblews
	What if the plane is abducted by the aliens? #MH370 if a mysterious island (Lost) can happen, so does an alien spaceship.
	Has somebody floated alien abduction theory for MH370?
M6 MH370 sighted in Maldives	BREAKING: Malaysia transport minister says reports of missing plane sighted over Maldives are untrue
	Minister: Maldives says it's not true that the plane was sighted in its airspace #MH370
	MH370: Reports that plane sighted in #Maldives not true
	RT Yahoo_MY: Plane sighted in Maldives? Not true, says Hishammuddin
	RT TODAYonline: #MH370 press con: Reports of plane sighted at Maldives are not true; forensic work underway to look at data deleted from...

Table 5: Sample claims found in MH370 dataset.

Groundtruth claim	**Sample tweets**
T269 President Obama visiting the Gulf of Mexico	President Obama will visit the Gulf of Mexico in the next 48 hours to check out the oil spill and response, per a White House official.
	RT @CNN: President Obama will visit the Gulf of Mexico in the next 48 hours to check out the oil spill and response.
	President Obama to visit Gulf of Mexico region in next 48 hours to check oil spill response, White House says.
	RT @GWPStudio: President Obama to visit site of oil spill in the Gulf of Mexico in next 48 hours http://bit.ly/cZ0q73 #oilspill
	RT @CNN: Just in: President Barack Obama will visit the Gulf of Mexico oil spill area on Sunday morning.
T2384 President Obama supports building of a mosque near ground zero	RT @croedemeierAP: WASHINGTON (AP) - President Obama supports allowing mosque to be built near ground zero in Manhattan.
	President Obama supports allowing mosque to be built near ground zero
	Obama backs Mosque near ground zero (AP): AP - President Barack Obama on Friday forcefully endorsed building ...
	Breaking news: President Obama backs mosque near ground zero
	Looks interesting: Obama backs mosque near ground zero: President
	Obama threw his support behind a controversial p...

Table 6: Sample claims found in Castillo dataset.

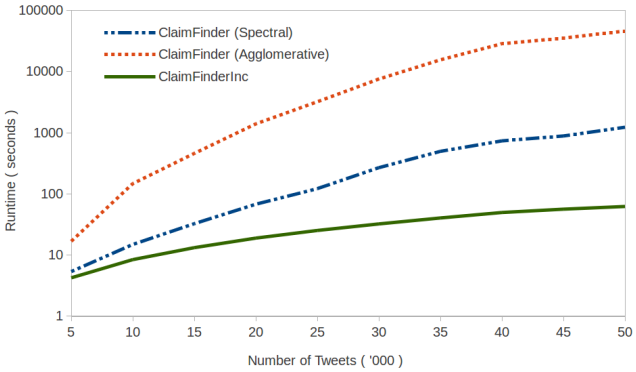

Figure 6: Scalability of $ClaimFinder^{INC}$ (MH370).

6. RELATED WORK

There are two main approaches to cluster tweets, namely features-based and topic modeling based clustering. Feature-based approach typically represent each tweet as a vector or set of features from which a similarity measure can then be used to quantify the distance between any given pair of tweets. A commonly used set of features is the TFIDF scores of the words present within the tweet content. Other features useful for differentiating individual tweet to their event include references to temporal, geographical and user information extracted from the tweet content [21]. These features are then used to cluster the tweets [9, 20, 8].

The alternative to features-based clustering is the generative topic modeling approaches, e.g., LDA [3]. However, the limited number of words present in microblog pose a major problem due to the lack of word co-occurrence within the tweets [11]. Empirical studies show that aggregating tweets such that each document is the concatenation of tweets from a user, hashtag or time window improves the topic clustering results [11][19][13]. The work in [23] assume that "a single tweet is usually about a single topic" and propose the TwitterLDA model where words in a tweet are either chosen from a topic or are background noise words. The TwitterLDA model is able to generate more coherent representative topic words compared to a standard LDA model.

To date, prior work on tweet or keywords clustering are designed mainly for topic or event detection, of which are overly encompassing in nature for the credibility assessment task. For example, an entity-oriented sample topic in [23] ''iphone6, #iphone, apple, app" correspond to tweets referring to the iPhone and/or the technology company while a event-oriented topic "health, flu, swine, #h1n1, #swineflu" correspond to tweets referring to the virus outbreak. The problem that there are multiple claims of varying credibility made within the tweets in each cluster remains unaddressed.

7. CONCLUSION

In this work, we observed that tweets may contain multiple claims and define a claim as comprising of subjects and predicates terms. We described a method called *ClaimFinder* to identify claims in a corpus of tweets related to some real world event. In particular, we use OpenIE techniques to identify entities and their relationships in tweets and map them to subject-predicate tuples. These tuples are then clustered such that each cluster refers to a claim. We further introduced an incremental approach to quickly process incoming tweets. Empirical evaluation on two real world datasets demonstrate the effectiveness of *ClaimFinder*, and scalability of $ClaimFinder^{INC}$. For future work, we plan to investigate existing features as well as information from other sources for credibility assessment.

8. REFERENCES

[1] L.M. Aiello, G. Petkos, and C. Martin et al. Sensing trending topics in twitter. *IEEE Transactions on Multimedia*, 15(6), 2013.

[2] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *AAAI Conference on Weblogs and Social Media*, 2011.

[3] D.M. Blei, A.Y. Ng, and M.I. Jordan et al. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[4] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *WWW*, 2011.

[5] C. Castillo, M. Mendoza, and B. Poblete. Predicting information credibility in time-sensitive social media. *Internet Research*, 2013.

[6] L.D. Corro and R. Gemulla. Clausie: Clause-based open information extraction. In *WWW*, 2013.

[7] L. Derczynski, A. Ritter, and S. Clark et. al. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Recent Advances in NLP*, 2013.

[8] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD*, 2001.

[9] E. Ferrara, M. JafariAsbagh, and O. Varol et. al. Clustering memes in social media. In *Advances in Social Networks Analysis and Mining*, 2013.

[10] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *Social Informatics*. 2014.

[11] L. Hong and B.D. Davison. Empirical study of topic modeling in twitter. In *SIGKDD Workshop on Social Media Analytics*, 2010.

[12] M. Mathioudakis and N. Koudas. Twittermonitor: Trend detection over the twitter stream. In *ACM SIGMOD*, 2010.

[13] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *ACM SIGIR*, 2013.

[14] M. Mendoza, B. Pobletey, and C. Castillo. Twitter Under Crisis: Can we trust what we RT? In *1st Workshop on Social Media Analytics,*, 2010.

[15] J. O'Donovan, B. Kang, and G. Meyer et. al. Credibility in context: An analysis of feature distributions in twitter. In *International Conference on Social Computing*, 2012.

[16] F. Pedregosa, G. Varoquaux, and A. Gramfort et. al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] M.F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.

[18] V. Qazvinian, E. Rosengren, D.R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *EMNLP*, 2011.

[19] Y. Wang, J. Liu, and J. Qu et. al. Hashtag graph based topic model for tweet mining. In *IEEE Data Mining*, 2013.

[20] C. Wartena and R. Brussee. Topic detection by clustering keywords. In *DEXA*, 2008.

[21] Y. Xia, X. Yang, and C. Wu et. al. Information credibility on twitter in emergency situation. In *Pacific Asia Conference on Intelligence and Security Informatics*, 2012.

[22] F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on sina weibo. In *ACM SIGKDD Workshop on Mining Data Semantics*, 2012.

[23] W. Zhao, J. Jiang, and J. Weng et. al. Comparing twitter and traditional media using topic models. In *European Conference on Advances in Information Retrieval*, 2011.