Challenges of Computational Verification in Social Multimedia

Christina Boididou
CERTH-ITI
boididou@iti.gr

Symeon Papadopoulos
CERTH-ITI
papadop@iti.gr

Yiannis Kompatsiaris
CERTH-ITI
ikom@iti.gr

Steve Schifferes
City University London
Journalism Department
steve.schifferes.1@city.ac.uk

Nic Newman
City University London
Journalism Department
nic.newman@gmail.com

ABSTRACT

Fake or misleading multimedia content and its distribution through social networks such as Twitter constitutes an increasingly important and challenging problem, especially in the context of emergencies and critical situations. In this paper, the aim is to explore the challenges involved in applying a computational verification framework to automatically classify tweets with unreliable media content as *fake* or *real*. We created a data corpus of tweets around big events focusing on the ones linking to images (fake or real) of which the reliability could be verified by independent online sources. Extracting content and user features for each tweet, we explored the fake prediction accuracy performance using each set of features separately and in combination. We considered three approaches for evaluating the performance of the classifier, ranging from the use of standard cross-validation, to independent groups of tweets and to cross-event training. The obtained results included a 81% for tweet features and 75% for user ones in the case of cross-validation. When using different events for training and testing, the accuracy is much lower (up to %58) demonstrating that the generalization of the predictor is a very challenging issue.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]

Keywords

Social media; Twitter; fake images; credibility

1. INTRODUCTION

With the extended use of online social media, a large volume of multimedia content is circulated on the Web. Twitter, as a news-oriented platform, suffers from significant amounts of misinformation and spam. Fake images, rumours and unreliable information about news and events often appear and get *viral*, sometimes leading to severe consequences. For example, during the Boston Marathon bombings suspect hunt, two innocent spectators were falsely portrayed as suspects of the explosions, while photos of them appeared in the cover of newspapers and in numerous tweet posts. This false alarm caused to them emotional distress and invasion of privacy even near loss of their jobs[1]. A similar case was also a student's disappearance allegedly linked to the Boston tragedy: his photo was posted on Twitter with the rumour of being a suspect[2]. His family had to cope, not only with the pain from his loss, as he was found dead days after, but also with the horrific smear provoked by the false information. Posting fake pictures can also cause panic and chaos among people, as exemplified during the Hurricane Sandy storm, when fake images of sharks inside New York were posted. Additionally, posts of a fake image of a giant creature found on a California beach, putting the blame on Fukushima radiation is an example of unreliable information that misinformed the public[3].

Taking into consideration the harmful consequences of false media content, there is a profound need to detect and control false information and prevent its spread. To this end, this paper contributes an open framework for computational verification of social media content and explores a set of challenges arising when applying the framework in real-world datasets. Compared to previous similar works in terms of approach [8], the paper aims at highlighting several practical complexities, such as the data collection, and methodological pitfalls, such as the selection of appropriate training and test sets. Finally, the paper aims at generating a set of open resources that will be constantly updated and be reusable by the interested scientific community.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the employed computational verification framework. Section 4 presents several experiments that explore the challenges of the problem at hand, and Section 5 contains a critical discussion of the problem and hints to the future work.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04
http://dx.doi.org/10.1145/2567948.2579323.

[1] http://www.nydailynews.com/news/national/boston-bombing-bag-men-sue-new-york-post-article-1.1365190

[2] https://twitter.com/NewsBreaker/status/325142081599332353

[3] https://twitter.com/EleChild/status/421255310683013120

2. RELATED WORK

Rumors, web and social network spamming. The problem of misinformation has been extensively studied before. In particular, the topic of Web spam detection was discussed by Gyongyi et al. who organized Web spamming techniques into a taxonomy and provided a framework for combating them [9]. Link-based and content-based dependencies among Web pages were used by Castillo et al. to develop an automated way to predict and identify Web spam [3]. Given the rise of social networks as an interaction platform, Seo et al. studied how rumours spread on them [13]. Considering that false claims come from a small number of sources, they tried to identify these rumours and their sources. Mendoza et al. studied the propagation of tweets that carry false rumours during emergencies, concluding that rumours are more questioned by users than the true information [11]. Benevenuto et al. leveraged user characteristics related to tweet content and social behaviour to classify them as spammers or non-spammers [1]. Automated identification of user spam accounts was also performed by Stringhini et al. who tried to spot a large number of spam accounts on Twitter by detecting anomalous behaviour [15].

User activity during crises. Additional research has been conducted on the field of social media activity analysis in the context of crises. Using as an example the Mumbai blasts, Gupta et al. studied the patterns of activity of users during the crisis and demonstrated that (a) non-authority users post more than other groups of users and (b) inaccurate information is more dominant [7]. The same authors, in another work [6], provided an automated ranking framework to predict rank of tweets according to their credibility. Cheong et al. used social network analysis to study the interactions between Twitter users during natural disasters and in particular floods. They studied the online communities formed during the flood to identify active players and their role in spreading information [5]. Furthermore, Yang et al. studied the inner and outer relationships of criminal user accounts [16]. Using the properties of their social relationships, they designed an algorithm for inferring criminal accounts, starting from a set of known ones.

Content credibility on social networks. Recent research has also been oriented towards the discovery of credible information sources in social networks. For instance, Canini et al. proposed a method, that given a particular topic, identifies users relevant to it based on a combination of their expertise and trust [2]. Similarly, Castillo et al. focused on automatic methods for assessing the credibility of a given set of tweets [4]. A recent work that is similar to ours was presented by Gupta et al. [8]. To distinguish between the shared fake and real images, the authors attempt to capture the patterns of fake twitter content, by using classification models on tweet text and user features. In our work, we build upon a similar machine learning framework in terms of the employed features and classifiers. However, motivated by the challenges we faced in replicating the reported results, we shift our focus on the issues of collecting reference fake media corpora and exploring different training/testing approaches in order to obtain more realistic estimates of the fake prediction accuracy performance.

3. FRAMEWORK

Here, we describe the data collection, feature extraction and classification methodology of the proposed framework.

3.1 Corpus Creation

We collect historical data from Twitter using the *Topsy*[4] API, which supports text-based search based on keywords, phrases or hashtags. In addition, it offers the option to filter the results according to type (*tweet, image or link*) and the *time period* they were posted.

For each case (event), we define a set of keywords K and the appropriate time period in order to gather a set of tweets T. Moreover, we define a set of unique fake pictures that spread on Twitter, as verified from online resources (articles and blogs), and we create the fake image set I_F. By following the same procedure, we form the real image set I_R. We use these sets as seeds to create our reference fake media corpus $T_C \subset T$. This corpus includes only those tweets that contain at least one image of the predefined sets of images I_F, I_R. However, in order not to restrict the tweets to only those that point to the exact seed image URLs, we also employ an optimized visual near-duplicate search strategy as described in [14]. More specifically, we extract compact descriptors from the collected images using a VLAD+SURF descriptor-aggregator combination. The extracted vectors are further encoded using Product Quantization, thus making the following Nearest Neighbour (NN) search more efficient. We use the sets of tweet images T_C as visual queries to the NN algorithm and for each query we check whether it exists as an image item or a near-duplicate image item of the I_F or the I_R set. With the help of similarity search, we extend the coverage of the dataset, taking into account the images that were not identical but very similar to the ones included in the seed sets. To ensure near-duplicity, we empirically set a minimum threshold of similarity tuned for high precision. However, a small amount of the images exceeding the threshold are eventually irrelevant to the ones in the seed set. To remove those, we conduct a manual verification step on the extended set of images, removing the irrelevant ones.

3.2 Feature extraction

After the data collection, the framework involves a feature extraction step, based on the content and the user information of each collected item. Those are summarized in Table 1 and described in the following paragraphs.

Table 1: Content and User features.

Content features F_T	
length of tweet	num of words
contains question mark	contains exclamation mark
num of question marks	num of exclamation marks
contains happy emoticon	contains sad emoticon
contains 1st order pronoun	contains 2nd order pronoun
contains 3rd order pronoun	num of uppercase characters
num of negative senti words	num of positive senti words
num of mentions	num of hashtags
num of URLs	num of retweets

User Features F_U	
num of friends	num of followers
follower-friend ratio	num of times listed
user has a URL	user is a verified user
num of tweets	

[4] http://www.topsy.com

3.2.1 Content features

These features are solely based on the content of tweets. We rely on the features used by Gupta et al. [8], to which we add the number of retweets for each tweet. The features are listed in Table 1. Beginning from the characteristics of the tweet, we compute features such as the length of the tweet and the number of words it contains. Also, we include features such as the number of question and exclamation marks or the number of uppercase characters included in the tweet text. To take into account the sentiment of the tweet, we compute the number of positive and negative words it contains, relying on a predefined list of sentiment words. We support three different languages English, Spanish and German, for each of which we make use of a different list of sentiment words. For English we use the list provided by Jeffrey Breen[5], for Spanish the Spanish adaptation for ANEW [12] and for German the Leipzig Affective Norms for German [10]. After detecting the text language of each of the tweets in T_C using an open language detection library[6], we form a new set of tweets $T_L \subset T_C$ that contains only those that were formulated in one of the defined languages. The result of feature extraction produces a list of tweet features F_T for each tweet of the T_L corpus.

3.2.2 User features

We also extract features from the Twitter user who made the post. The user's number of friends and followers, as well as whether the user is verified or not, are included in the list. The set of user features F_U is presented in Table 1.

3.3 Building the Classifier

The aim of this step is to assess the ability of the classifier to distinguish between fake and real tweets. We consider three main training approaches.

The first classification building approach is similar to the one of Gupta et al. [8]. According to it, a two-class 10-fold cross-validation scheme is used based on the corpus T_L of tweets. The approach is illustrated in Figure 1. To avoid class imbalance complications, we select equal number of fake and real tweets in the training set.

The second approach builds the classifier using tweets associated with independent groups of images in the training and test set. To make this clear, let us assume that the set of fake images I_F, defined above, consists of n images and the T_L set of t tweets. We first randomly select m of the n images $I_F^{train} = \{I_{F_1}, I_{F_2}, .., I_{F_m}\}$ and we collect f fake tweets of the T_L set that include one of the m images. This set of tweets forms the training dataset for the classifier. The remaining *(t-f)* tweets contain one of the rest $I_F^{test} = \{I_{F_{m+1}}, .., I_{F_n}\}$ images and are used to form the test set. We perform the same splitting procedure for the real images selecting k of them and separating the training and test dataset by the tweets containing the I_R^{train} images and those containing the I_R^{test} ones. By following this methodology, we test the performance of the classifier when the samples in the training and testing sets are completely independent, i.e. $(I_F^{train} \cup I_R^{train}) \cap (I_F^{test} \cup I_R^{test}) = \emptyset$. Figure 2 illustrates the idea.

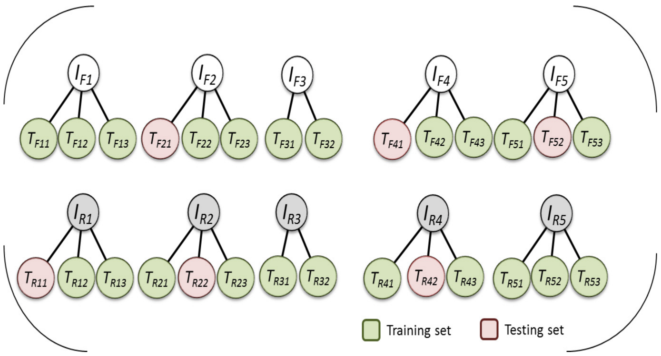


Figure 1: Cross-validation approach. Randomly include items to training and test sets.

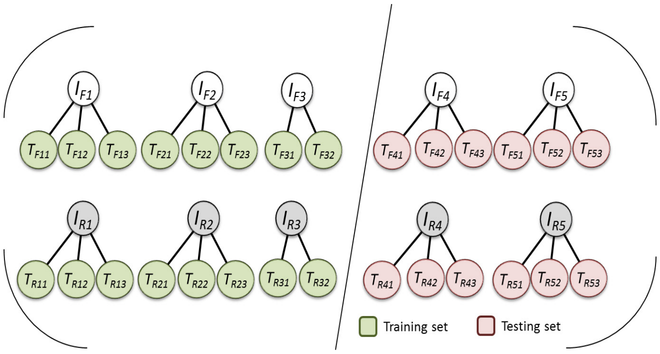


Figure 2: Independent training/test set creation approach. $I_{F1}, .., I_{F5}$ and $I_{R1}, .., I_{R5}$ are the fake and real images respectively, while $T_{F1}...T_{F5}$ and $T_{R1}...T_{R5}$ the corresponding tweets. The example uses n=5 fake images, m=3 selected fake images and k=3 selected real images for training.

In the last approach, we use for training a dataset of tweets T_{L1} collected around a specific event, and for testing a set of tweets T_{L2} collected around a different event. In this way, we attempt to assess the classifier accuracy when it is required to verify content from a different source than the one it was trained with.

4. EXPERIMENTS

In the following, we describe in detail the conducted experiments using the aforementioned approaches and the obtained results.

4.1 Corpus Creation

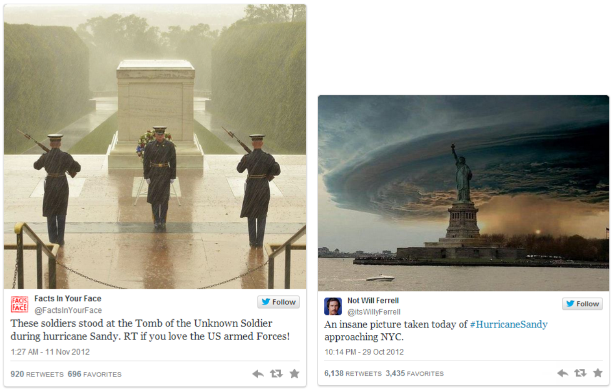
We collected sets of tweets around two big events, Hurricane Sandy (HS) and the Boston Marathon (BM) terrorist hunt. ***Hurricane Sandy***[7] was a natural disaster that caused destruction and turmoil around the US from October 22nd to 31st, 2012. According to NBC News, the death toll in the US was 109 and damages exceeded $50 billion. Social media such as Twitter helped people be informed of the latest updates of the hurricane, but it was also maliciously used to spread rumours and fake media. As mentioned in the introduction, images of sharks inside New York and the

[5] https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107

[6] https://code.google.com/p/language-detection/

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

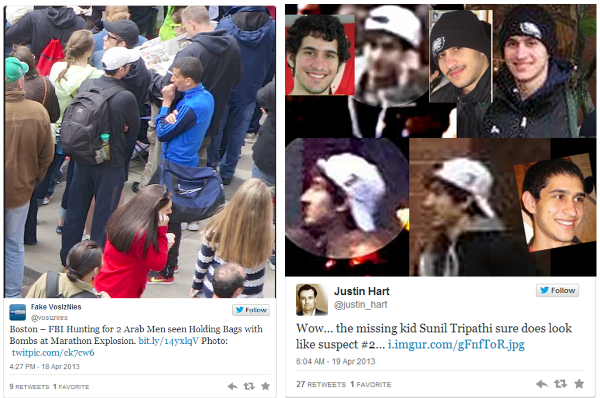
flooded Statue of Liberty went viral. Figure 3 shows two sample tweets containing two example fake images.



Facts In Your Face
@FactsInYourFace
These soldiers stood at the Tomb of the Unknown Soldier during hurricane Sandy. RT if you love the US armed Forces!
1:27 AM - 11 Nov 2012
920 RETWEETS 696 FAVORITES

Not Will Ferrell
@itsWillyFerrell
An insane picture taken today of #HurricaneSandy approaching NYC.
10:14 PM - 29 Oct 2012
6,138 RETWEETS 3,435 FAVORITES

Figure 3: Fake Hurricane Sandy tweets.



Fake VosIzNies
@vosiznies
Boston – FBI Hunting for 2 Arab Men seen Holding Bags with Bombs at Marathon Explosion. bit.ly/14yxlqV Photo: twitpic.com/ck7cw6
4:27 PM - 18 Apr 2013
9 RETWEETS 1 FAVORITE

Justin Hart
@justin_hart
Wow... the missing kid Sunil Tripathi sure does look like suspect #2... i.imgur.com/gFnfToR.jpg
6:04 AM - 19 Apr 2013
27 RETWEETS 1 FAVORITE

Figure 4: Fake Boston Marathon tweets.

Boston Marathon bombings[8] was also an event that gathered massive social media activity. It occurred on 15 April, 2013 during the Boston Marathon when two pressure cooker bombs exploded at 2:49 pm EDT, killing three people and injuring an estimated 264 others. The Federal Bureau of Investigation (FBI) took over the investigation, and on April 18, they released photographs and a surveillance video of two suspects. Apart from these two suspects, a lot of images of possible fake suspects were also shared online. Tweets were posted, claiming possible suspects and publishing images of the time of explosion from private and street cameras. Two fake tweets concerning the event are illustrated in Figure 4.

To proceed with the dataset collection, we identified a set of online resources for HS, such as official articles and blogs, that marked images posted about the storm as fake or real. The result of this manual research was to identify 16 unique fake images and 70 unique real images. Using Topsy, we collected tweets with keywords and hashtags that are listed in Table 2. The hurricane occurred from 20^{th} October to 1^{st} November, so we set this as the period of interest for our data collection. The gathered tweets were filtered to keep only those containing at least one image from our predefined fake or real seed sets. To avoid manually selecting the set of tweets, we applied the NN algorithm described before to select the similar images. From the extended set of images, only a small percentage (approximately 5%) were manually found to be irrelevant to the seed sets and were therefore removed. Eventually, applying the framework of Section 3, we managed to maintain only the tweets that contained either a fake or a real picture. Table 3 shows the statistics for the collected data.

Table 2: Keywords and Hashtags for data collection

Hurricane Sandy	
Hurricane Sandy	#hurricaneSandy
Hurricane	#hurricane
Sandy	#sandy
Boston Marathon	
Boston Marathon	#bostonMarathon
Boston bombings	#bostonbombings
Boston suspect	#bostonSuspect
manhunt	#manhunt
watertown	#watertown
Tsarnaev	#Tsarnaev
4chan	#4chan
Sunil Tripathi	#prayforBoston

Table 3: Statistics of tweets for each corpus

	HS	BM
Total tweets with images	343,939	112,449
Total unique users	238,982	95,743
Tweets with fake images	10,757	281
Users with fake images	10,431	278
Tweets with real images	3,540	460
Users with real images	3,540	417

Similar to HS, we also defined list of seed images for BM. Here we focused on resources that pointed to suspects of the bombing. For identifying the real suspects we relied on The Independent as well as on additional resources presented in Table 4. In the end, we managed to gather 22 unique pictures showing real suspects and 18 unique pictures for fake suspects. In this case, to form our dataset, we used the keywords of Table 3 to collect tweets that were posted between 15^{th} April to 22^{nd} April, 2013.

Except for the words directly linked to Boston Marathon, we also used *4chan*, a blog that published images of fake suspects. Following the dataset collection approach of subsection 3.1, the images of the tweets collected, were filtered performing the similarity algorithm in order to maintain only tweets that contained images contained in the fake or real seed lists. Note that the majority of tweets gathered around the event were tagged with the #prayforBoston hashtag (intended to express support for people hit from the bombings). This is why, observing the statistics, there is a big difference between the total number of tweets and the number of the tweets containing images with suspects (Table 3).

4.2 Prediction accuracy assessment

The next step was to apply the classifier building approaches of subsection 3.3 to assess the performance of fake

[8] http://en.wikipedia.org/wiki/Boston_Marathon_bombings

Table 4: Resources used for each event.

	Hurricane Sandy	
1	*the Week*	http://theweek.com/article/index/235578/10-fake-photos-of-hurricane-sandy
2	*Mashable*	http://mashable.com/2012/10/29/fake-hurricane-sandy-photos/
3	*Atlantic*	http://www.theatlantic.com/technology/archive/2012/10/sorting-the-real-sandy-photos-from-the-fakes/264243/
4	*the Wire*	http://www.thewire.com/national/2012/10/most-unbelievable-real-pictures-sandys-destruction/58492/

	Boston Marathon	
1	*The Independent*	http://www.independent.co.uk/news/world/americas/boston-brothers-tamerlan-and-dzhokhar-tsarnaev-had-planned-4-july-attack-says-official-8601983.html
2	*Timeturk*	http://www.timeturk.com/en/2013/04/19/fbi-releases-photos-of-two-boston-bomb-suspects.html
3	*nowtheendbegins*	http://www.nowtheendbegins.com/blog/?p=13739
4	*4chan*	http://imgur.com/a/sUrnA
5	*Int'l Business Time*	http://www.ibtimes.com/sunil-tripathi-wrongly-identified-boston-marathon-bombing-suspect-missing-brown-university-student

image classification in different settings. For the HS dataset, we first applied the 10-fold cross validation approach using each of the selected classifiers listed in Table 5 in order to classify the tweets containing fake images and the tweets containing real images. Because of the unequal size of the real and fake tweets in the corpus (fake items were much more), we randomly selected equal number of fake tweets. We produced results using content and user features separately as well as in combination.

Table 5: Two-class 10-fold cross-validation results.

Hurricane Sandy

Classifier	Tweet	User	Total
J48 tree	81.41%	67.72%	80.68%
KStar	81.28%	71.16%	81.38%
Random Forest	80.59%	70.15%	80.94%

Boston Marathon

Classifier	Tweet	User	Total
J48 tree	76.45%	70.81%	81.25%
KStar	81.28%	74.12%	75.78%
Random Forest	78.59%	76.15%	79.10%

Comparing the results among the different cases, as Table 5 illustrates, it is obvious that the content features are generally more effective for the detection of fake content than the user-based ones. With the content features we obtained a percentage of 81%, while with the user features about 70%. That means that the way a user composes the tweet is more important for identifying the credibility of the tweets, than the user's characteristics and details. By combining the two kinds of features (total features), the results are really close to the ones coming from the content features.

In the case of BM, classifying the tweets containing images of fake or real suspects using the cross-validation approach, resulted in a 81.28% accuracy using the content features. Using only the user features, the correctly classified tweets was limited to 76% demonstrating once more that the content features are more effective for predicting fakes. Observing the results of total features classification, we note that they are similar to the ones obtained from using only content features (Table 5). Generally, the combination of the two kinds of features leads to similar or marginally better classification scores.

Table 6: Detection accuracy with different training and test set (HS).

Classifier	Tweet	User	Total
J48 tree	73.79%	51.06%	65.06%
KStar	75.30%	62.29%	53.31%
Random Forest	74.02%	63.10%	65.96%

Table 7: Detection accuracy using HS for training and BM for testing.

Classifier	Tweet	User	Total
J48 tree	55.05%	50.12%	54.10%
KStar	50.01%	50.10%	50.97%
Random Forest	58.75%	51.03%	58.78%

We also experimented in separating the corpora according to the second approach of subsection 3.3. We selected a part of the fake and a part of the real images and used the tweets that contained them to train the classifier. The rest of the tweets that also contained fake images but different ones than those of the training set were used to test the classifier. The aim of this experiment was to test our classification method when we fully separate the training and testing sets. For HS, applying this approach in combination with a J48 decision tree and content features led to a 73% detection accuracy. Similar scores, 74% and 75%, were achieved when we used Random Forest and KStar classifiers respectively. For the user features case, we observed lower accuracy scores, while somewhat higher scores were obtained when the combination of features was used (Table 6).

In the last experiment, we used for training the HS corpus and for testing the BM corpus. Testing the classifier with another dataset than the one used for training, was really challenging as the nature of the two events is different. Observing the scores of this experiment in Table 7, we remark that classification precision is not much higher than the random baseline (50-58%) in all implemented experiments. Despite the fact that both datasets included tweets posting content either real or fake, it is difficult to generalize from the features of the tweets of the first one to predict the veracity of the tweets around the second event.

5. CHALLENGES

This paper highlights the challenges involved in building a computational verification framework to detect fake multimedia content by use of content features, user features and their combination. Although we collected tweets based on a predefined set of images and for particular events, the data collection faced a series of practical issues. For the HS corpus, although massive amounts of fake content were posted during the event, it gradually started disappearing (either removed by its owners or suspended from Twitter). Additionally, the real content was sparse with much less elements, leading us to restrict the number of the tweets used for building the classifier. In a similar manner, the collected BM items were very few, in particular the fake ones. We believe this is a common challenge that researchers working on the problem are expected to face when trying to collect fake photos from past events. At the same time, copyright issues (in the case of media content) and terms of service limitations for the various social network services make dataset building and reuse a really complicated issue.

A further point for consideration stems from the large deviation we noted on the detection accuracy reported by Gupta et al. [8] and the one we could achieve. In [8], the claimed accuracy using cross-validation reaches the extremely high score of 97% when content features are used. In our case, we achieved a maximum detection accuracy of 81%. Although some deviation could be expected as a result of using a different corpus (yet around the same event), it becomes evident that research on such topics would greatly benefit from reproducible solutions.

Last but not least, we need to recognize that in real-world settings such as in the case when the classification is assessed on a completely different dataset than the one that the classifier was build on, the fake detection accuracy is far less impressive than in the "artificial" case of using cross-validation on a single-event dataset. The setting becomes even more complicated if the problem is formulated in the context of an ongoing event. This points to the need for collecting a much larger and varied set of fake content corpora in order to bridge the gap between the performance obtained in lab experimental settings and the one that would arise in real-world applications.

The work described in this paper attempts to create an experimental testbed for assessing the performance of computational verification approaches on social multimedia. A first version of the presented framework can be found on GitHub[9]. There are several issues that call for further work. More specifically, it is worth adding as a user feature the geographic location of the user. Building on the hypothesis that for a user, being closer to the event's location, increases his reliability for the information he spreads, we expect that this feature should carry considerable predictive power. Similarly, the incorporation of features from appropriately selected terms (based on statistical analysis of an independent set of fake and real tweets) should also carry considerable predictive power. Additionally, the time the tweet was posted and its distance from the beginning of the event is a content feature that could potentially offer more information about the veracity of the tweet. Finally, the experimentation with more datasets from different events, would help draw more reliable conclusions with respect to the effectiveness of different features and classifiers.

6. ACKNOWLEDGEMENTS

This work is supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

7. REFERENCES

[1] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *Collaboration, Electronic messaging, Anti-abuse and Spam conference (CEAS)*, volume 6, 2010.

[2] K. R. Canini, B. Suh, and P. L. Pirolli. Finding credible information sources in social networks based on content and social structure. In *IEEE Third International Conference on Social Computing (SocialCom)*, pages 1–8. IEEE, 2011.

[3] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference*, pages 423–430. ACM, 2007.

[4] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World Wide Web*, pages 675–684. ACM, 2011.

[5] F. Cheong and C. Cheong. Social media data mining: A social network analysis of tweets during the 2010-2011 australian floods. In *PACIS*, page 46, 2011.

[6] A. Gupta and P. Kumaraguru. @ twitter credibility ranking of tweets on events# breakingnews. 2012.

[7] A. Gupta and P. Kumaraguru. Twitter explodes with activity in mumbai blasts! a lifeline or an unmonitored daemon in the lurking? 2012.

[8] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 729–736. International World Wide Web Conferences Steering Committee, 2013.

[9] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *First international workshop on adversarial information retrieval on the web (AIRWeb 2005)*, 2005.

[10] P. Kanske and S. A. Kotz. Leipzig affective norms for german: A reliability study. *Behavior research methods*, 42(4):987–991, 2010.

[11] M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we rt? In *Proceedings of the first Workshop on Social Media Analytics*, pages 71–79. ACM, 2010.

[12] J. Redondo, I. Fraga, I. Padrón, and M. Comesaña. The spanish adaptation of anew (affective norms for english words). *Behavior research methods*, 39(3):600–605, 2007.

[13] E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *SPIE Defense, Security, and Sensing*, pages 83891I–83891I. International Society for Optics and Photonics, 2012.

[14] E. Spyromitros-Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, and I. Vlahavas. An empirical study on the combination of surf features with vlad vectors for image search. In *13th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4. IEEE, 2012.

[15] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 1–9. ACM, 2010.

[16] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 71–80. ACM, 2012.

[9] https://github.com/socialsensor/computational-verification