# Learning to Detect Misleading Content on Twitter

Christina Boididou, Symeon Papadopoulos, Lazaros Apostolidis, Yiannis Kompatsiaris
Information Technologies Institute, Centre for Research and Technology Hellas
Thessaloniki, Greece
{boididou,papadop,laaposto,ikom}@iti.gr

## ABSTRACT

The publication and spread of misleading content is a problem of increasing magnitude, complexity and consequences in a world that is increasingly relying on user-generated content for news sourcing. To this end, multimedia analysis techniques could serve as an assisting tool to spot and debunk misleading content on the Web. In this paper, we tackle the problem of misleading multimedia content detection on Twitter in real time. We propose a number of new features and a new semi-supervised learning event adaptation approach that helps generalize the detection capabilities of trained models to unseen content, even when the event of interest is of different nature compared to that used for training. Combined with bagging, the proposed approach manages to outperform previous systems by a significant margin in terms of accuracy. Moreover, in order to communicate the verification process to end users, we develop a web-based application for visualizing the results.

## CCS CONCEPTS

**•Information systems → Information retrieval; Test collections;** *Multimedia and multimodal retrieval;* **•Human-centered computing → Social networking sites;**

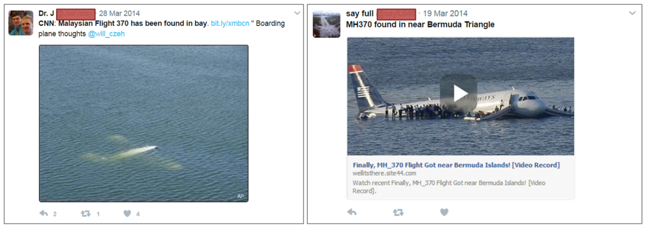
## KEYWORDS

social media, verification, fake detection, news mining

**ACM Reference format:**
Christina Boididou, Symeon Papadopoulos, Lazaros Apostolidis, Yiannis Kompatsiaris. 2017. Learning to Detect Misleading Content on Twitter. In *Proceedings of ICMR ’17, June 6–9, 2017, Bucharest, Romania, ,* 9 pages.
DOI: http://dx.doi.org/10.1145/3078971.3078979

## 1 INTRODUCTION

Recent years have seen tremendous increase in the use of social media platforms such as Twitter and Facebook as means of sharing news content and multimedia. The simplicity of the sharing process has led to large volumes of news content propagating over social networks and reaching huge numbers of readers in very short time. Especially multimedia content (images, videos) can rapidly reach massive audiences and become viral due to the fact that it is easily consumed and often carries a lot of entertainment value.



Dr. J [redacted] · 28 Mar 2014
CNN: Malaysian Flight 370 has been found in bay. bit.ly/xmbcn " Boarding plane thoughts @will_czeh

say full [redacted] · 19 Mar 2014
MH370 found in near Bermuda Triangle

Finally, MH_370 Flight Got near Bermuda Islands! [Video Record]
wellitsthere.site44.com
Watch recent Finally, MH_370 Flight Got near Bermuda Islands! [Video Record].

**Figure 1: Examples of fake imagery that spread on Twitter during the Malaysian airlines incident on March 2014.**

Given the speed of the news spreading process and the competition of news outlets and journalists to publish first, it is only natural that the verification of content is often carried out in a superficial manner or even neglected. This leads to the online appearance and spread of large amounts of misleading multimedia content. In particular, when a news event breaks (e.g., a natural disaster), and media coverage is of primary importance, news professionals often turn to social media to source newsworthy content. It is exactly this setting, when the risk of misleading content becoming viral is the highest. As misleading (or for the sake of brevity *fake*), we define any post that shares multimedia content that does not faithfully represent the event that it refers to. This could include: a) content from a past event that is reposted as referring to the current event, b) content that is deliberately manipulated, or c) content that is falsely used to represent an aspect of the current event. In a similar way, as *real*, we define posts that share content that faithfully represents the event in question. There are in-between cases: for instance, when a post acknowledges the misleading nature of the content it shares or refers to it with a sense of humour, it is hard to categorize as fake or real; these are out of the scope of this work.

The impact of fake content being widely disseminated can be quite severe. For example, after the Malaysia Airlines flight disappeared on March 2014 (Figure 1), numerous fake images that became soon viral on social media raised false alarms that the plane was detected. This deeply affected and caused emotional distress to people directly involved in the incident, such as the passengers’ families. Examples such as this point to the need for means of identifying and debunking fake media content on social media. One of the first such attempts [8] used a supervised learning approach, in which a set of known fake and real tweets were used to train a model to distinguish between the two classes; experiments were conducted on a dataset around the Hurricane Sandy and a very high detection accuracy was reported. Yet, the fact that content from the same event was used both for training and testing was found to give an overly optimistic sense of accuracy, questioning its generalization ability to content from different events [3].

*ICMR ’17, June 6–9, 2017, Bucharest, Romania*

ACM ISBN 978-1-4503-4701-3/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3078971.3078979

To address the limitations of state-of-the-art solutions on the problem, we present a robust approach for detecting in real time whether a media item shared by a tweet is fake or real. The proposed fake detection approach uses a variety of content-based and contextual features for the social media post in question, and leverages part of its own predictions for retraining, following a semi-supervised learning paradigm, to adapt the model to unseen content. Experiments on a public annotated corpus of multimedia tweets demonstrate the effectiveness of the proposed approach. Additionally, we propose a visualization method for communicating the result of automatic analysis to end users in an intuitive way.

## 2 RELATED WORK

**Multimedia forensics.** Although the field of multimedia forensics has led to a multitude of methods for detecting digital manipulation in digital content [18, 21], recent research has shown that tampered images found on the Web are very hard to detect [27, 28]. Moreover, in a lot of cases, forensics techniques are insufficient, e.g. when the multimedia item is just a reposting from a past event. Indeed, past studies have demonstrated that more than half of the videos around trending topics are repostings or remixes of past content [26]. Also, the verification methods employed by journalists [23], e.g. looking into the Exif metadata of content or getting in touch with the person that published it, are often not applicable due to the constraints of popular social media platforms. For instance, Twitter and Facebook remove the Exif metadata from posted content.

**Assessing content credibility in social media.** Castillo et al. focused on automatic methods for assessing the credibility of a given set of tweets. In particular, they analysed microblog posts related to trending topics and classified them as credible or not credible based on a number of features [5]. A similar approach was presented by Gupta et al. [8], demonstrating high classification accuracy on a dataset of tweets collected around Hurricane Sandy. A thorough experimental study of information credibility on Twitter was also based on information propagation processes in the context of news events [6]. Two models were developed, one that decides whether an information cascade corresponds to a newsworthy event and another one that evaluates the trustworthiness of the cascade. In contrast with the aforementioned approaches, Martinez-Romo et al. [13] conducted a study for detecting malicious tweets in trending topics focusing on statistical linguistic analysis, taking into account exclusively the tweets without considering any information from users. Finally, O'Donovan et al. [16] performed an analysis of the utility of various features when predicting content credibility.

**Verification services and systems.** Ratkiewicz et al. developed the Truthy system [17], a web service for tracking political memes and misinformation on Twitter, focusing on political astroturf. Truthy collects tweets, detects a number of memes in them, and offers a web interface that lets users annotate those memes they consider "truthy". In recent years, systems that are fully-automatic have been developed, such as TweetCred [7], a tool that computes credibility scores for a set of tweets, and Hoaxy [22], a platform for detecting and analysing online misinformation. Finally, semi-automatic systems have been also introduced, such as RumorLens [20], which combines human effort with computation to detect new rumours in Twitter, and TwitterTrails [14], which lets users investigate the propagation of a given rumour.

## 3 FAKE DETECTION FRAMEWORK

The proposed framework relies on two independent classification models built on the training data using two different sets of features, tweet-based (TB) and user-based features (UB). A bagging technique is used when building both models. At prediction time, an agreement-based retraining strategy is employed (`fusion`), which combines the outputs of the two models in a semi-supervised learning manner, to increase the generalization capabilities of the framework given tweets from a new unknown event. The outcome of the verification is then visualized to end users. A corpus of labelled posts is necessary (described in Section 4) in order to build the classification models and to generate the visualizations. Figure 2 depicts the main components of the proposed framework. The implementation of the framework is publicly available on GitHub[1].

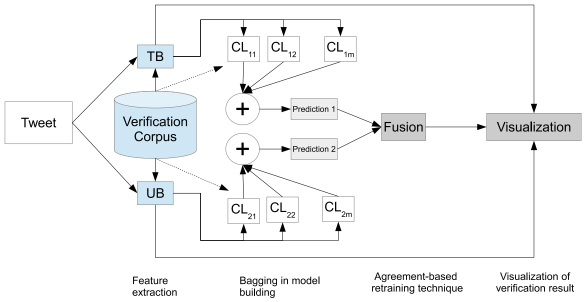


**Figure 2: Overview of the proposed framework.**

### 3.1 Feature Extraction

The selection of features used in our framework was carried out following a thorough study of the way in which news professionals, such as journalists, verify content on the Web. Based on relevant journalistic studies, such as [12], and the Verification Handbook [23], we have defined a set of features that are important for verification. These are not limited to the content itself, but also pertain to its source (Twitter account that posted the content) and to the location where it was posted. We decided to not use any image/video forensics features following the conclusion of our recent study [27] that Twitter media content is not amenable to image forensics. This was also confirmed by our recent MediaEval participations [2, 4], where the use of forensics features did not lead to consistent improvement. The feature extraction process produces a set of TB and UB features for each tweet (Table 1).

**Tweet-based features (TB)**: We consider four types of feature related to tweets: a) text-based, b) language-specific, c) Twitter-specific, and d) link-based.
***a) text-based***: These are extracted from the text of the tweet, and capture characteristics such as the length of a tweet text and the number of words in it. They also include characteristics such as the number of question and exclamation marks, uppercase characters, as well as binary features indicating the existence or not of emoticons, special words ("please") and punctuation (colon).

[1] https://github.com/MKLab-ITI/computational-verification

**Table 1: Overview of verification features. Link-based features are extracted in the TB case for external links that tweets may share, and in the UB case for the URL included in the account profile. Features with an asterisk were proposed in [3, 8] and will be denoted as Baseline Features (BF), while the full feature set (BF and newly proposed ones) as Total Features (TF).**

| Tweet-based Features (TB) | |
|---|---|
| **text-based** | |
| #words* | has ‘‘please’’ |
| length of text* | has colon |
| #question marks* | contains happy emoticon* |
| #exclamation marks* | contains sad emoticon* |
| contains question mark* | #uppercase chars* |
| contains exclamation mark* | |
| **language-specific** | |
| #pos senti words* | contains 1st pers.pron.* |
| #neg senti words* | contains 2nd pers.pron.* |
| #slangs | contains 3rd pers.pron.* |
| #nouns | readability |
| **twitter-specific** | |
| #retweets* | #mentions* |
| #hashtags* | #URLs* |
| has external link | |

| User-based Features (UB) | |
|---|---|
| **user-specific** | |
| #friends* | has location |
| #followers* | has existing location |
| follower-friend ratio* | has bio description |
| #media content | tweet ratio |
| has profile image | account age |
| has header image | is verified* |
| has a URL* | #times listed* |
| **link-based (common for TB and UB)** | |
| WOT score | alexa country rank |
| in-degree centrality | alexa delta rank |
| harmonic centrality | alexa popularity |
| | alexa reach rank |

***b) language-specific***: These are extracted for a predefined set of languages (English, Spanish, German), which are first detected using a language detection library[2]. They include the number of positive and negative sentiment words in the text. For English we use the list by Jeffrey Breen[3], for Spanish the adaptation of ANEW [19] and for German the Leipzig Affective Norms [10]. Additional binary features indicate whether the text contains personal pronouns (in the supported languages). An additional feature is the number of slang words in the tweet. This is extracted using slang words in English[4] and Spanish[5]. For German, no available slang list was found and hence no such feature is computed. Moreover, the number of nouns in the tweet text was also added as feature, and is computed based on the Stanford parser only for English [11]. Finally, to investigate whether the readability of the tweet text is related to its veracity, we use the *Flesch Reading Ease* method[6] to compute a readability score in the range [0, 100], with 0 representing the very hard-to-read text and 100 the very easy-to-read text. For the tweets written in a language, where the above features cannot be extracted, we consider the corresponding values missing.
***c) twitter-specific***: This set contains features related to the Twitter platform. These include the number of re-tweets, hashtags, mentions, URLs and a binary feature expressing whether any of the URLs points to external (non-Twitter) resources.
***d) link-based***: These include features that provide information about the links that are shared through the tweet. This set of features is common in both TB and UB sets, but in the latter it is defined in a different way (see link-based category in UB features). For TB, depending on the existence of an external URL in the tweet, its reliability is quantified based on a set of Web metrics: i) the WOT score[7], which is a way to assess the trust on a website using crowdsourced reputation ratings, ii) the in-degree and harmonic centralities[8], computed based on the links of the Web graph, and iii) four Alexa metrics (rank, popularity, delta rank and reach rank) based on the rankings API[9].

**User-based features (UB)**: These are related to the Twitter account posting the tweet. We divide them in a) user-specific and b) link-based features.
***a) user-specific***: These include the user’s number of friends and followers, the account age, the follower-friend ratio, tweet ratio (number of tweets/day divided by account age) and a number of binary features: whether the user is verified by Twitter, whether there is a biography in his/her profile, whether the user declares his/her location using a free text field, and whether the location text can be parsed into an actual location[10], whether the user has header or profile image, and whether a link is included in the profile.
***b) link-based***: In this case, depending on the existence of a URL in the Twitter profile description, we apply the same Web metrics as the ones used in the link-based TB features. If there is no link in the profile, the values of these features are considered to be missing.

After feature extraction, the next steps include pre-processing, cleaning and transformation. To handle the issue of missing values on some of the features, we use linear regression for estimating their values: we consider the attribute with the missing value as a dependent (class) variable and apply linear regression for numeric features. The method cannot support the prediction of boolean values and hence those are left missing. Only feature values from the training set are used in this process. Data normalization is also performed to scale the numeric feature values to the range [-1, 1].

## 3.2 Building the classification models

We use the TB and UB features to build two independent Random Forest classifiers ($\mathtt{CL}_1$, $\mathtt{CL}_2$), each of which is based on the respective set of features. To further increase classification accuracy, we make use of bagging: we create $m$ different subsets of tweets from the training set, including equal number of samples for each class (some samples may appear in multiple subsets), leading to the creation of $m$ instances of $\mathtt{CL}_1$ and $\mathtt{CL}_2$ ($m = 9$ in our experiments), as shown in Figure 2. The final prediction for each of the test samples is calculated using the majority vote among the $m$ predictions.

[2] https://code.google.com/p/language-detection/
[3] https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107
[4] http://onlineslangdictionary.com/word-list/0-a/
[5] http://www.languagerealm.com/spanish/spanishslang.php
[6] http://simple.wikipedia.org/wiki/Flesch_Reading_Ease
[7] https://www.mywot.com/
[8] http://wwwranking.webdatacommons.org/more.html
[9] http://data.alexa.com/data?cli=10&dat=snbamz&url=google.gr
[10] Based on https://github.com/socialsensor/geo-util project.

## 3.3 Agreement-based retraining

A key novelty in the proposed framework is an *agreement-based retraining* step (the fusion block in Figure 2) to improve the prediction accuracy for content associated with unseen events. This was motivated by an approach that was proposed for effectively tackling out-of-domain sentiment classification [25]. We combine the outputs of classifiers $\mathtt{CL}_1$, $\mathtt{CL}_2$ as follows: for each sample of the test set, we compare their predictions and depending on their agreement, we divide the test set in the *agreed* and *disagreed* subsets. The instances of the *agreed* set are assigned the agreed label (fake/real) assuming that it is correct with high likelihood, and are used to build a new classifier to handle the *disagreed* instances. To this end, we use two retraining techniques. First, we select the most effective of the independent classifiers $\mathtt{CL}_1$, $\mathtt{CL}_2$ based on their performance on the training set during cross-validation. Then, we either use just the agreed samples to train the $\mathtt{CL}$ classifier (denoted as $\mathtt{CL(i)}$), or we use the entire set of initial training samples extending it with the set of agreed samples (denoted as $\mathtt{CL(ii)}$). The goal is to adapt the initial model to the specific data characteristics of the new event. In that way, the model can predict more accurately the values of the samples for which $\mathtt{CL}_1$, $\mathtt{CL}_2$ did not initially agree. In the experimental section, we test both retraining variants.

## 3.4 Verification result visualization

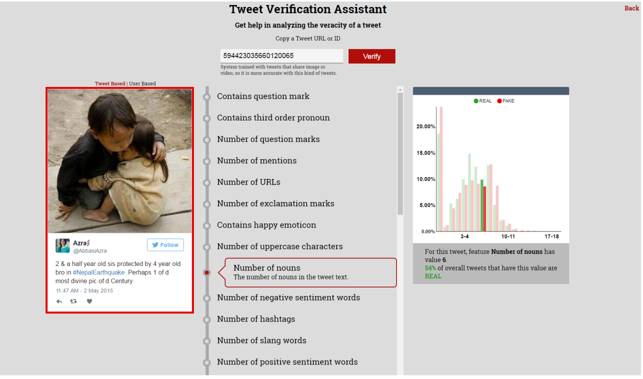
The key idea for visualizing the results of the proposed verification process is to present the list of extracted features for the input tweet, and then for a selected feature to present its value in relation to the distribution that this feature has for real versus fake tweets, as computed with respect to the verification corpus (Section 4).

Figure 3 illustrates a screenshot of this application, which is publicly available[11]. In terms of usage, the end user first provides the URL or id of a tweet of interest, and then the application presents the extracted tweet- and user-based features and the verification result (fake/real) for the tweet in the form of a color-coded frame (red/green respectively). It also offers the possibility of inspecting the feature values. By selecting a feature, its value distribution appears (right column), separately for the fake and real tweets (side-by-side). Moreover, a textual description informs the user about the percentage of tweets of this class (fake or real) that have the same value for this feature. In that way, the investigator may better understand how the verification result is justified based on the individual values of the features in relation to the "typical" values that these features have for fake versus real tweets.

# 4 VERIFICATION CORPUS

Our fake detection models are based on a publicly available a verification corpus ($VC$) of fake and real tweets. More specifically, this consists of tweets related to the 17 events of Table 2, comprising in total 193 cases of real images, 218 cases of misused (fake) images and two cases of misused videos, associated with 6,225 real and 9,596 fake tweets posted by 5,895 and 9,216 unique users respectively.

The corpus comprises a set of tweets $T$ that is collected with the help of a set of keywords $K$ for each of the 17 events. The ground truth labels (fake/real) of these tweets are based on a set of online resources, which discussed and debunked images and videos widely



594423035660120065

System trained with tweets that share image or video, so it is more accurate with this kind of tweets.

Azra @AbbasiAzra

2 & a half year old sis protected by 4 year old bro in #NepalEarthquake .Perhaps 1 of d most divine pic of d Century.

11:47 AM - 2 May 2015

Number of nouns
The number of nouns in the tweet text.

For this tweet, feature **Number of nouns** has value **6**.
54% of overall tweets that have this value are **REAL**

**Figure 3: Snapshot of the Tweet Verification Assistant interface. Given a tweet, a user can explore the verification result, including the extracted feature values and their distribution on the Verification Corpus.**

shared in the context of these events. Only resources were used that are reputable news providers and that adequately justified their decision about the veracity of each multimedia item. This led to a set of fake and real multimedia cases, denoted as $I_F$, $I_R$ respectively, which were then used as seeds to create the reference verification corpus $T_C \subset T$. This includes exclusively tweets that contain at least one item from the two sets. In order not to restrict the tweets to only those that point to the exact seed URLs, a visual near-duplicate search technique was employed [24]. More specifically, the sets of fake and real images were used as visual queries and for each query it was checked whether each image tweet from $T$ exists as an image item or a near-duplicate image item of the $I_F$ or the $I_R$ set. To ensure near-duplicity, a minimum threshold of similarity was empirically set, tuned for high precision. A small amount of the images exceeding the threshold were manually found to be irrelevant to the ones in the seed set and were then removed.

Several of the events, e.g., Columbian Chemicals, Passport Hoax and Rock Elephant, were actually hoaxes, hence all content associated with them is fake. Also, for several real events (e.g., MA flight 370) no real images (and hence no real tweets) were included in the dataset, since none came up as a result of the data collection.

As the aim of our work is to assess the generalization capability of the fake detection framework, we used every tweet in the corpus regardless of its language. The aim has been to use a comprehensive corpus, which contains the widest possible variety of fake tweets (even though this complicates the machine learning process due to missing feature values). Furthermore, we included content from different types of event. In terms of type of fake, we considered the following four categories: **a) reposting of real**: real photos from past events re-posted as being associated to a current event (Figure 4 (i)); **b) reposting of synthetic**: synthetic digital images, such as artworks or snapshots from movies, presented as real imagery about an event (Figure 4 (ii)); **c) speculations**: real photos from an ongoing event, expressing speculations regarding the association of persons or actions to the event (Figure 4 (iii)); **d) digital tampering**: digitally manipulated photos (Figure 4 (iv)).

[11] http://reveal-mklab.iti.gr/reveal/fake/

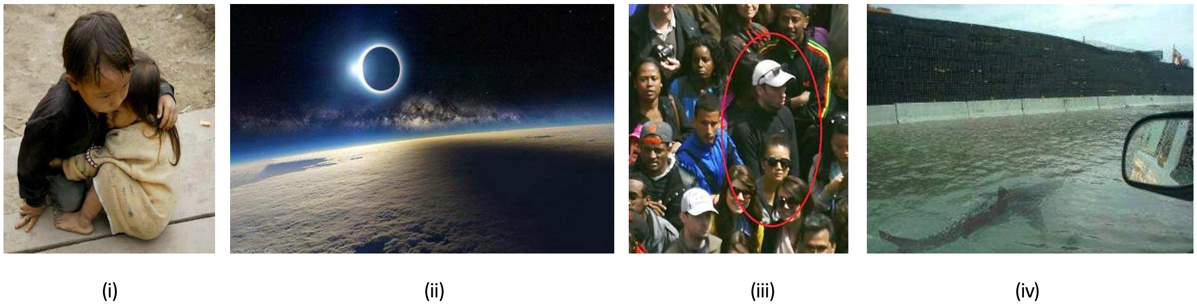


**Figure 4: Types of fake: (i) reposting of real photo depicting two Vietnamese siblings as being captured during the Nepal 2015 earthquakes; (ii) reposting of artwork as a photo from Solar Eclipse of March 2015; (iii) speculation of someone as being suspect of the Boston Marathon bombings in 2013; (iv) spliced sharks on a photo captured during Hurricane Sandy in 2012.**

**Table 2: List of events in $VC$: For each event, we report the number of unique real (if available) and fake images ($I_R$, $I_F$ respectively), unique tweets that shared those images ($T_R$, $T_F$) and Twitter accounts that posted those tweets ($U_R$, $U_F$).**

| ID | Name | $I_R$ | $T_R$ | $U_R$ | $I_F$ | $T_F$ | $U_F$ |
|---|---|---|---|---|---|---|---|
| E1 | Hurricane Sandy | 148 | 4,664 | 4,446 | 62 | 5,559 | 5,432 |
| E2 | Boston Marathon bombing | 28 | 344 | 310 | 35 | 189 | 187 |
| E3 | Sochi Olympics | - | - | - | 26 | 274 | 252 |
| E4 | Bring Back Our Girls | - | - | - | 7 | 131 | 126 |
| E5 | MA flight 370 | - | - | - | 29 | 501 | 493 |
| E6 | Columbian Chemicals | - | - | - | 15 | 185 | 87 |
| E7 | Passport hoax | - | - | - | 2 | 44 | 44 |
| E8 | Rock Elephant | - | - | - | 1 | 13 | 13 |
| E9 | Underwater bedroom | - | - | - | 3 | 113 | 112 |
| E10 | Livr mobile app | - | - | - | 4 | 9 | 9 |
| E11 | Pig fish | - | - | - | 1 | 14 | 14 |
| E12 | Nepal earthquake | 11 | 1004 | 934 | 21 | 356 | 343 |
| E13 | Solar Eclipse | 4 | 140 | 133 | 6 | 137 | 135 |
| E14 | Garissa Attack | 2 | 73 | 72 | 2 | 6 | 6 |
| E15 | Samurai and Girl | - | - | - | 4 | 218 | 212 |
| E16 | Syrian Boy | - | - | - | 1 | 1786 | 1692 |
| E17 | Varoufakis and ZDF | - | - | - | 1 | 61 | 59 |
| | Total | 193 | 6225 | 5895 | 220 | 9596 | 9216 |

From the corpus, we considered only unique posts by eliminating re-tweets. Finally, by manually checking the content of tweets, we ensured that no posts were included that featured funny/humorous content, nor posts that declared that their content is fake (both of which cases would be hard to classify as either real or fake).

# 5 EXPERIMENTAL STUDY

## 5.1 Overview

The aim of the conducted experiments was to evaluate the fake detection accuracy of different models on samples from new (unseen) events. We consider this an important aspect of a verification framework, as the nature of the untrustworthy (fake) tweets posted may vary across different events. Accuracy is computed as the ratio of correctly classified samples ($N_c$) over total number of test samples ($N$): $a = N_c/N$. The employed evaluation scheme can be thought of as a kind of *event-based cross-validation*: for each event $E_i$ of the 17 events in the $VC$, we use the remaining 16 events for training. and $E_i$ for testing. We denote each of these 17 splits as $T_i$. All models are built using Random Forests of 100 trees.

In addition, to compare the performance of our framework, with methods that participated in the recently organized Verifying Multimedia Use task in the context of *MediaEval* [1], we use the split proposed by the task organizers (denoted as $T18$): events $E1$-$E11$ are used for training, and events $E12$-$E17$ for testing.

## 5.2 New Features and Bagging

We first assess the contribution of the new features and bagging to the method's accuracy. To this end, we build the $\mathtt{CL}_1$, $\mathtt{CL}_2$ classifiers with and without the bagging technique. To create the models without bagging, we selected each time an equal number of random fake and real samples for training. We applied this procedure both for the Baseline (BF) and Total Features (TF) (cf. Table 1 caption). Table 3 presents the average accuracy for each setting.

We observe that the use of bagging led to considerably improved accuracy for both $\mathtt{CL}_1$ and $\mathtt{CL}_2$. In addition, further improvements are achieved when using the TF features over BF. We see that bagging led to an absolute improvement of approximately 10% and 15% in the accuracy of $\mathtt{CL}_1$ and $\mathtt{CL}_2$ respectively (when using the TF features), while the use of TF features over BF to an absolute improvement of approximately 18% on both classifiers (when bagging is used). Combined, the use of bagging and the newly proposed features led to an absolute improvement of more than 24% for both $\mathtt{CL}_1$ and $\mathtt{CL}_2$. Given the clear benefits of using bagging, in subsequent experiments, all reported results refer to classifiers with bagging.

**Table 3: Comparison between $CL_1$, $CL_2$, and the effect of bagging and Total Features ($TF$) over Baseline Features ($BF$).**

| | $\mathtt{CL}_1$ | $\mathtt{CL}_1$-bagging | $\mathtt{CL}_2$ | $\mathtt{CL}_2$-bagging |
|---|---|---|---|---|
| TF | 78.01 | 88.34 | 60.89 | 75.70 |
| BF | 64.14 | 70.57 | 51.15 | 57.40 |

**Table 4: Accuracy for the entire set of features $TF$. Agreement levels between the $\mathtt{CL}_1$ and $\mathtt{CL}_2$ classifiers, agreed, disagreed and overall accuracy for each model (`CL(i)`, `CL(ii)`) and each split.**

| Trial | Agreement percentage | Agreed accuracy | $CL(i)$ disagreed accuracy | $CL(ii)$ disagreed accuracy | $CL(i)$ overall | $CL(ii)$ overall | $CL_1$ bagging | $CL_2$ bagging |
|---|---|---|---|---|---|---|---|---|
| T1 | 71.54 | 95.11 | 57.89 | 62.75 | 84.54 | 86.05 | 74.10 | 90.56 |
| T2 | 57.12 | 88.64 | 81.63 | 61.36 | 85.87 | 77.71 | 68.79 | 76.06 |
| T3 | 67.73 | 93.73 | 83.40 | 85.20 | 90.36 | 91.09 | 90.58 | 68.83 |
| T4 | 93.43 | 99.72 | 96.66 | 75.24 | 99.23 | 98.54 | 96.10 | 96.87 |
| T5 | 78.82 | 96.81 | 76.76 | 82.58 | 92.33 | 93.77 | 90.07 | 83.79 |
| T6 | 95.45 | 100 | 100 | 82.38 | 100 | 98.79 | 95.67 | 99.78 |
| T7 | 57.95 | 90.19 | 66.59 | 96.87 | 89.54 | 93.18 | 92.95 | 54.09 |
| T8 | 76.92 | 99 | 100 | 100 | 99.23 | 99.23 | 99.23 | 76.15 |
| T9 | 89.38 | 99.29 | 100 | 94.81 | 99.38 | 99.02 | 96.01 | 92.12 |
| T10 | 80 | 97.32 | 100 | 100 | 97.78 | 97.78 | 96.67 | 78.88 |
| T11 | 82.85 | 100 | 100 | 60 | 100 | 93.57 | 91.42 | 91.42 |
| T12 | 50.83 | 79.11 | 84.42 | 86.78 | 81.94 | 83.10 | 82.26 | 47.60 |
| T13 | 54.90 | 91.89 | 69.16 | 45.07 | 81.59 | 71.11 | 67.47 | 78.77 |
| T14 | 55.82 | 94.21 | 87.82 | 81.76 | 91.51 | 89.24 | 87.97 | 62.02 |
| T15 | 61.46 | 95.29 | 88.03 | 95.36 | 93.66 | 95.64 | 83.90 | 62.15 |
| T16 | 84.77 | 97.59 | 76.68 | 73.66 | 93.81 | 93.98 | 91.55 | 89.14 |
| T17 | 42.29 | 80.43 | 68.93 | 89 | 86.89 | 85.73 | 85.57 | 40.98 |
| T18 | 70.39 | 94.79 | 80.55 | 79.77 | 90.74 | 90.51 | 89.80 | 73.32 |
| **Average** | **70.65** | **94.06** | **84.14** | **80.70** | **92.13** | **91.00** | **88.34** | **75.69** |

## 5.3 Agreement-based retraining technique

We use the entire set of features (TF) for assessing the accuracy of the agreement-based retraining approach. Table 4 shows the scores obtained separately for each split. The first two columns present the agreement level and the accuracy of classifiers on the agreed set. We observe that on average the two classifiers' predictions ($\mathtt{CL}_1$, $\mathtt{CL}_2$) agree in the majority of tweets (Agreement (%) column); in particular, they agree on 70.65% of the tweets on average. On this set of tweets, the average accuracy (Agreed accuracy column) is extremely high (94.06%). One may also note that the higher the agreement level, the higher is the achieved accuracy on the agreed set. The next two columns present the accuracy on the disagreed samples, when using the two variations of the retraining process (`CL(i)` and `CL(ii)` in Section 3.3). The next two columns show the results while combining the accuracy of the agreed and disagreed samples. On average, the first retraining variation, i.e. using only the samples of the new event for training, slightly outperforms the second. For comparison purposes, the last two columns of the table present the scores of the $\mathtt{CL}_1$ and $\mathtt{CL}_2$ classifiers trained with the bagging technique and applied independently. Those correspond to the standard supervised learning paradigm [3, 8]. Comparing the average scores of the classifiers in the two last columns (88.34% and 75.69% respectively) with those of the agreement-based retraining technique (92.13% and 91%), one can see a clear improvement in terms of classification accuracy (approximately 4% when compared to the best $\mathtt{CL}_1$ configuration).

## 5.4 Performance on different languages

We also assessed the classification accuracy of the framework for tweets written in different languages, i.e. the extent to which the framework is language-dependent.

We keep the five most used languages in the corpus (by number of tweets). Note that in many cases no language is detected, either because the text contains no text but just hashtags/URLs or the length of the text is too small to be detected by the language detector. For this reason, we also consider this category of tweets (denoted as `NO-LANG`), and thus compare between the following cases: English (`EN`), Spanish (`ES`), no language (`NO-LANG`), Dutch (`NL`) and French (`FR`). Table 5 shows the languages tested and the corresponding number of samples.

By using the total amount of features `TF`, we calculate the accuracy on each split ($T1$-$T18$) separately on the samples of each language. Figure 5 shows the results for each split when using the agreement-based retraining technique, according to the first and second variation of the method respectively. In most cases, it appears that fake detection accuracy remains relatively stable independent of language. The highest accuracy scores are achieved for `NO-LANG` followed by English and Spanish. Accuracy is somewhat lower for French and Dutch. This is an encouraging finding since it indicates that the framework is reliable even for languages, for which the language-specific features are not defined.

**Table 5: Accuracy for most frequent languages on VC.**

| **Language** | **EN** | **ES** | **NO LANG** | **NL** | **FR** |
|---|---|---|---|---|---|
| **Samples** | 12,301 | 1,107 | 729 | 279 | 271 |
| **Accuracy (%)** | 90.18 | 87.40 | 99.21 | 88.82 | 91.06 |

## 5.5 Comparison with state of the art

We also compare our method with the ones submitted to the MediaEval 2015 Verifying Multimedia Use task. These include the systems by `UoS-ITI` [15], `MCG-ICT` [9], and `CERTH-UNITN` [2]. For each of those, we only compare against their best *run*[12]. The comparison is done using the F1-score, which is the official metric of the task. According to the results (Table 6), the proposed method achieves the second best performance ($F = 0.934$), reaching almost

[12] In MediaEval, each system/team can submit up to five runs.

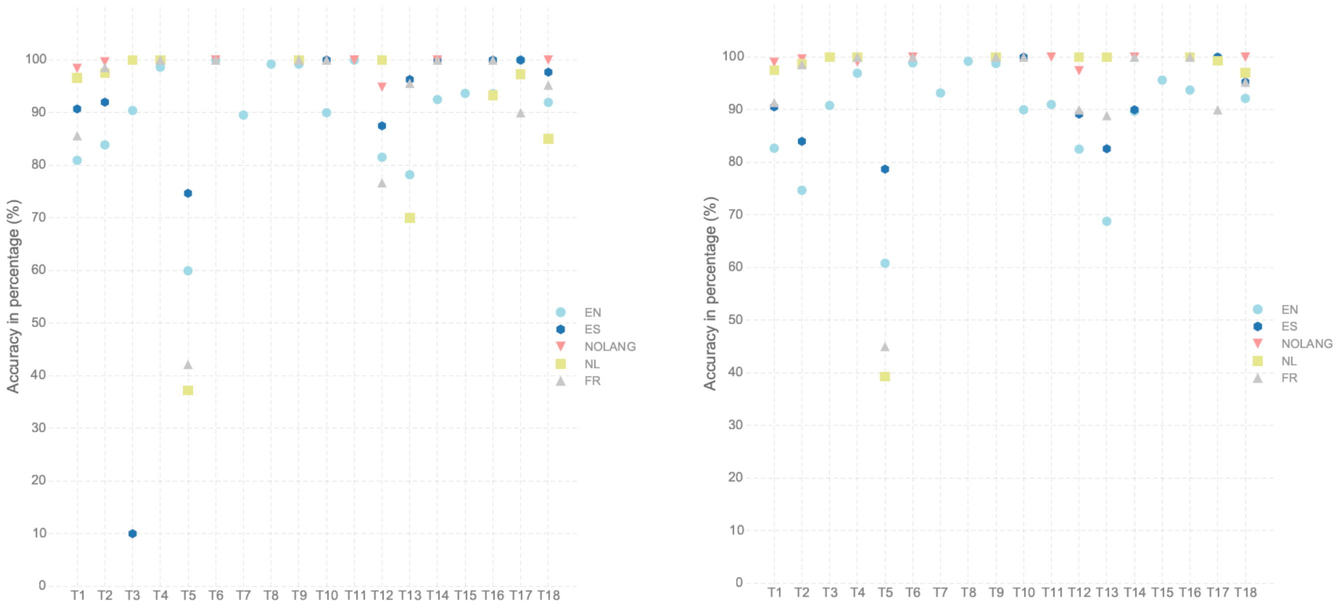


**Figure 5: Language-based accuracy based on the TF features and the agreement-based retraining technique: left) using just the agreed samples for training (CL(i)), and right) using the agreed and the initial training samples for training (CL(ii)).**

**Table 6: Comparison among MediaEval 2015 submissions and the performance of our T18 run.**

| Method | F1-Score |
|---|---|
| UoS-ITI [15] | 0.830 |
| MCG-ICT [9] | 0.942 |
| CERTH-UNITN [2] | 0.911 |
| Proposed | 0.934 |

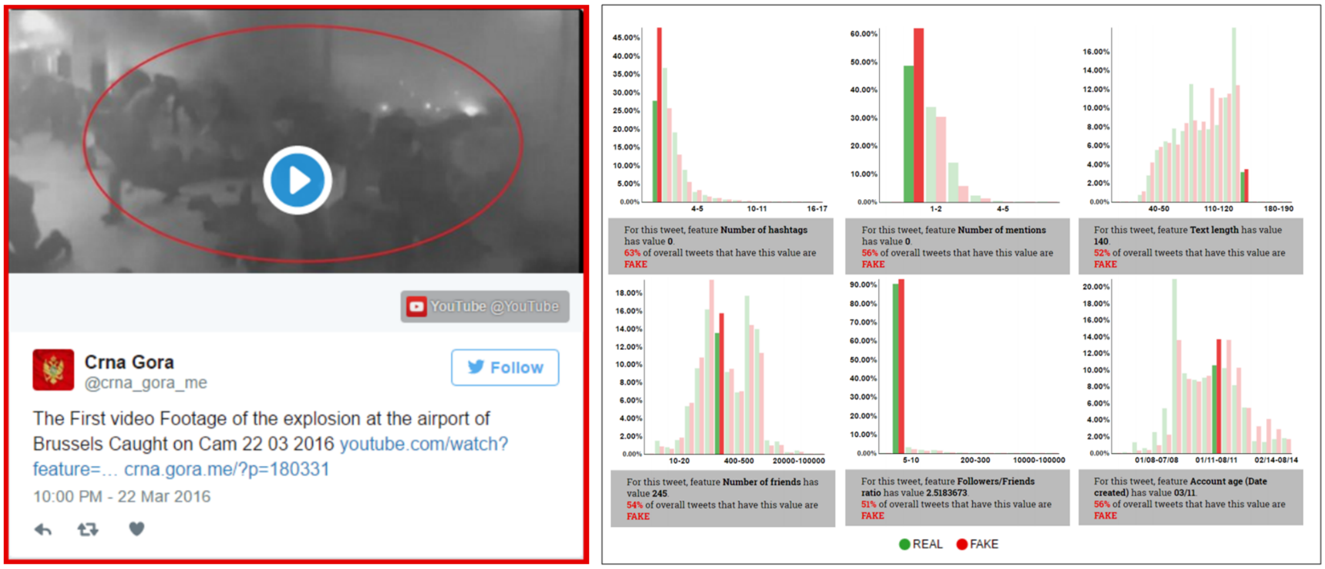
equal performance to the best run by MCG-ICT [9] ($F = 0.942$). The latter, however, uses an approach that is tailored to the specifics of the dataset. In particular, MGC-ICT relies on a model that first clusters tweets into topics according to the multimedia resource that they contain. Then, it extracts topic-level features for building the fake detection classifier. It is important to note that the dataset of the task makes available a list of tweets, their associated multimedia item and label (fake/real). The way the dataset is structured makes the MGC-ICT possible to apply. However, in a realistic setting, unseen tweets do not appear in clusters (except in the case of highly popular media items that are shared concurrently by numerous different posts), which makes the application of such an approach much more complex and its results questionable. In contrast, our method leads to comparable performance without suffering from such limitations.

### 5.6 Verification visualization

To demonstrate the utility of the web-based verification application, we present an example case study where the proposed visualization approach is used on a tweet that shared fake multimedia content in the context of the March 2016 terrorist attacks in Brussels. The tweet (Figure 6) claimed that the shared video depicted one of the explosions in Zaventem airport, but the video is actually from another explosion in a different airport a few years ago. Indeed, the proposed classification framework flags the tweet as fake and presents the features' distributions in order to get useful insights about the reasons for this decision. Three sample tweet- and user-based feature distributions are illustrated in the upper and lower part of Figure 6 respectively. For example, in the first plot, the number of hashtags for this tweet is shown to be zero and at the same time the respective bar is highlighted. The plot informs that 63% of the overall training tweets that have this value are fake, a fact that partially justifies the classification result. In the two following plots that display the number of mentions and the text length, similar conclusions can be made about the veracity of the tweet. In the user-based feature value distributions, the date of creation, the number of friends and the followers/friends ratio seem to give some additional strong signals regarding the credibility of the account, and as a result the veracity of the posted tweet.

## 6 CONCLUSIONS AND FUTURE WORK

We presented a robust and effective framework for fake multimedia detection on Twitter. Using a specially collected verification corpus, we provided evidence of the high accuracy of the proposed framework over a number of events of different size and nature, as well as considerable improvements in accuracy as a result of the newly proposed features, the use of bagging, and the application of an agreement-based retraining method that outperforms standard supervised learning. We also demonstrated the utility of a novel visualization approach for explaining the verification result.



YouTube @YouTube

Crna Gora
@crna_gora_me

Follow

The First video Footage of the explosion at the airport of Brussels Caught on Cam 22 03 2016 youtube.com/watch?feature=… crna.gora.me/?p=180331

10:00 PM - 22 Mar 2016

For this tweet, feature **Number of hashtags** has value **0**.
63% of overall tweets that have this value are FAKE

For this tweet, feature **Number of mentions** has value **0**.
56% of overall tweets that have this value are FAKE

For this tweet, feature **Text length** has value **140**.
52% of overall tweets that have this value are FAKE

For this tweet, feature **Number of friends** has value **245**.
54% of overall tweets that have this value are FAKE

For this tweet, feature **Followers/Friends ratio** has value **2.5183673**.
51% of overall tweets that have this value are FAKE

For this tweet, feature **Account age (Date created)** has value **03/11**.
56% of overall tweets that have this value are FAKE

**Figure 6: Tweet sharing fake video content and feature analysis produced by the Tweet Verification Assistant.**

To use the proposed approach in real-time settings, one should be cautious of the following caveat. The agreement-based retraining method requires a number of samples from the new event in order to be applied. Hence, for the first set of arriving items, it is not possible to rely on this improved step. Yet, the rate at which new items arrive in the context of breaking news events could quickly provide the algorithm with a sizeable set of tweets.

In the future, we are interested in looking further into the real-time aspects of fake content detection, and conduct experiments that better simulate the fake content detection problem as an event evolves. We also plan to conduct user studies to test whether the proposed visualization is understandable and usable by news editors and journalists. Finally, we also plan to extend the framework to be applicable to content posted on platforms other than Twitter.

## 7 ACKNOWLEDGMENTS

This work has been supported by the REVEAL and InVID projects, under contract numbers 610928 and 687786 respectively, funded by the European Commission.

## REFERENCES

[1] Christina Boididou, Katerina Andreadou, Symeon Papadopoulos, Duc-Tien Dang-Nguyen, Giulia Boato, Michael Riegler, and Yiannis Kompatsiaris. 2015. Verifying Multimedia Use at MediaEval 2015. In *MediaEval 2015 Workshop, Sept. 14-15, 2015, Wurzen, Germany.*

[2] Christina Boididou, Symeon Papadopoulos, Duc-Tien Dang-Nguyen, Giulia Boato, and Yiannis Kompatsiaris. 2015. The CERTH-UNITN Participation @ Verifying Multimedia Use 2015. (2015).

[3] Christina Boididou, Symeon Papadopoulos, Yiannis Kompatsiaris, Steve Schifferes, and Nic Newman. 2014. Challenges of Computational Verification in Social Multimedia. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion.* 743–748.

[4] Christina Boididou, Symeon Papadopoulos, Stuart E. Middleton, Duc-Tien Dang-Nguyen, Michael Riegler, Andreas Petlund, and Yiannis Kompatsiaris. 2016. The VMU Participation @ Verifying Multimedia Use 2016. In *Working Notes Proceedings of the MediaEval 2016 Workshop, The Netherlands, Oct 20-21, 2016.*

[5] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web.* ACM, 675–684.

[6] Daniel Gayo-Avello, Panagiotis Metaxas, Eni Mustafaraj, Markus Strohmaier, Harald Schoen, Daniel Peter Gloor, Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2013. Predicting information credibility in time-sensitive social media. *Internet Research* 23, 5 (2013), 560–588.

[7] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. Tweetcred: A real-time Web-based system for assessing credibility of content on Twitter. In *Proc. 6th International Conference on Social Informatics (SocInfo).*

[8] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. 2013. Faking Sandy: characterizing and identifying fake images on Twitter during Hurricane Sandy. In *Proceedings of the 22nd international conference on World Wide Web companion.* 729–736.

[9] Zhiwei Jin, Juan Cao, Yazi Zhang, and Yongdong Zhang. 2015. MCG-ICT at MediaEval 2015: Verifying Multimedia Use with a Two-Level Classification Model.. In *MediaEval 2015 Workshop, Sept. 14-15, 2015, Wurzen, Germany.*

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

[11] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL '03).* Association for Computational Linguistics, Stroudsburg, PA, USA, 423–430.

[12] Nora Martin and BA Comm. 2014. Information Verification in the Age of Digital Journalism. In *Special Libraries Association Annual Conference, Vancouver.*

[13] Juan Martinez-Romo and Lourdes Araujo. 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications* 40, 8 (2013), 2992–3000.

[14] Panagiotis Metaxas, Samantha Finn, and Eni Mustafaraj. 2015. Using Twitter-Trails.com to Investigate Rumor Propagation. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing.* ACM, 69–72.

[15] Stuart Middleton. 2015. Extracting attributed verification and debunking reports from social media: Mediaeval-2015 trust and credibility analysis of image and video. (2015).

[16] John O'Donovan, Byungkyu Kang, Greg Meyer, Tobias Hollerer, and Sibel Adalii. 2012. Credibility in context: An analysis of feature distributions in twitter. In *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Conference on Social Computing (SocialCom).* IEEE, 293–301.

[17] Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. 2011. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th international conference companion on World wide web.* ACM, 249–252.

[18] Judith A. Redi, Wiem Taktak, and Jean-Luc Dugelay. 2011. Digital Image Forensics: A Booklet for Beginners. *Multimedia Tools Appl.* 51, 1 (Jan. 2011), 133–162.

[19] Jaime Redondo, Isabel Fraga, Isabel Padrón, and Montserrat Comesaña. 2007. The Spanish adaptation of ANEW (affective norms for English words). *Behavior research methods* 39, 3 (2007), 600–605.

[20] Paul Resnick, Samuel Carton, Souneil Park, Yuncheng Shen, and Nicole Zeffer. 2014. Rumorlens: A system for analyzing the impact of rumors and corrections in social media. In *Proc. Computational Journalism Conference*.

[21] Anderson Rocha, Walter Scheirer, Terrance Boult, and Siome Goldenstein. 2011. Vision of the unseen: Current trends and challenges in digital image and video forensics. *ACM Computing Surveys (CSUR)* 43, 4 (2011), 26.

[22] Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. 2016. Hoaxy: A Platform for Tracking Online Misinformation. In *Proceedings of the 25th Intern. Conf. Companion on World Wide Web*. 745–750.

[23] Craig Silverman. 2013. Verification handbook. (2013).

[24] Eleftherios Spyromitros-Xioufis, Symeon Papadopoulos, Ioannis Kompatsiaris, Grigorios Tsoumakas, and Ioannis Vlahavas. 2014. A Comprehensive Study Over VLAD and Product Quantization in Large-Scale Image Retrieval. *IEEE Transactions on Multimedia* 16, 6 (2014), 1713–1728.

[25] Adam Tsakalidis, Symeon Papadopoulos, and Ioannis Kompatsiaris. 2014. An Ensemble Model for Cross-Domain Polarity Classification on Twitter. In *Web Information Systems Engineering–WISE 2014*. Springer, 168–177.

[26] Lexing Xie, Apostol Natsev, Xuming He, John R Kender, Mark Hill, and John R Smith. 2013. Tracking large-scale video remix in real-world events. *Multimedia, IEEE Transactions on* 15, 6 (2013), 1244–1254.

[27] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2015. Detecting image splicing in the wild (WEB). In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 1–6.

[28] Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2017. A Large-Scale Evaluation of Splicing Localization Algorithms for Web Images. *Multimedia Tools and Applications* 76, 4 (February 2017), 4801–4834.