Expert Systems with Applications 40 (2013) 2992–3000

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Detecting malicious tweets in trending topics using a statistical analysis of language

Juan Martinez-Romo *, Lourdes Araujo

NLP & IR Group, Dpto. Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia (UNED), Madrid 28040, Spain

ARTICLE INFO

Keywords:
Spam detection
Social network
Statistical natural language processing
Machine learning

ABSTRACT

Twitter spam detection is a recent area of research in which most previous works had focused on the identification of malicious user accounts and honeypot-based approaches. However, in this paper we present a methodology based on two new aspects: the detection of spam tweets in isolation and without previous information of the user; and the application of a statistical analysis of language to detect spam in trending topics. Trending topics capture the emerging Internet trends and topics of discussion that are in everybody's lips. This growing microblogging phenomenon therefore allows spammers to disseminate malicious tweets quickly and massively. In this paper we present the first work that tries to detect spam tweets in real time using language as the primary tool. We first collected and labeled a large dataset with 34 K trending topics and 20 million tweets. Then, we have proposed a reduced set of features hardly manipulated by spammers. In addition, we have developed a machine learning system with some orthogonal features that can be combined with other sets of features with the aim of analyzing emergent characteristics of spam in social networks. We have also conducted an extensive evaluation process that has allowed us to show how our system is able to obtain an F-measure at the same level as the best state-of-the-art systems based on the detection of spam accounts. Thus, our system can be applied to Twitter spam detection in trending topics in real time due mainly to the analysis of tweets instead of user accounts.

1. Introduction

Twitter is one of the most popular online social networking and microblogging services that enables its users to send and read text-based posts of up to 140 characters, known as "tweets". Nowadays, millions of users use Twitter to keep in touch with friends, meet new people and discuss about everything. In a social network, people exchange a great deal of information and it is common to observe that certain individuals have especially strong influences on others. Choi and Han (2012) call these highly influential people opinion leaders. For these reasons and because of its fast growing, cyber criminals have used it as the new platform to achieve their malicious goals. The growing popularity of microblogging sites like Twitter has sparked a corresponding rise in social networking scams (Yang, Harkreader, & Gu, 2011). This Twitter scam is in full swing, using tempting messages like "Just saw this photo of you" followed by a link that, when you click it, takes you to a site that uploads malware (Villeneuve, 2010) onto your computer. Sometimes, by exploiting the phishing techniques (Nishanth, Ravi, Ankaiah, & Bose, 2012; Ramanathan & Wechsler, 2012), the message may seem to come from one of your regular followers, perhaps even a friend or relative. In fact, their Twitter account has been hijacked. Hackers also use Twitter to send coded update messages to computers they have previously infected with rogue code to control bot-nets (Nazario, 2009), which are assemblages of infected PCs that can be directed to spy on their users, send spam, or attack web sites with fake traffic.

One of the most popular tools in twitter is the list of trending topics (Lee, 2012) that capture the hottest emerging trends and topics of discussion. Using this feature of twitter, people can quickly gather news about a particular topic or learn at a glance which are the topics on which most people speak. Unfortunately, this growing microblogging phenomenon allows spammers to disseminate malicious tweets. Twitter provides several methods for users to report spam and these reports are investigated by Twitter and the accounts being reported are suspended in case of spam. However, reporting spam abuses using these methods is not very useful for trending topics because the suspension process is slow while the trending topics are ephemeral in most cases and they last for a few hours or a day at most.

While most existing approaches (Benevenuto, Magno, Rodrigues, & Almeida, 2010; Lee, Caverlee, & Webb, 2010; Shekar, Wakade, Liszka, & Chan, 2010; Stringhini, Kruegel, & Vigna, 2010; Thomas, Grier, Ma, Paxson, & Song, 2011a, Thomas, Grier, Song, & Paxson, 2011b; Wang, 2010; Yardi, Romero, Schoenebeck, & Boyd, 2010) focus on detecting Twitter criminal accounts individually,

* Corresponding author.
E-mail addresses: juaner@lsi.uned.es (J. Martinez-Romo), lurdes@lsi.uned.es (L. Araujo).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.015

our approach for the spam problem focuses on the detection of tweets containing spam instead of detecting spam accounts. The detection of spam tweets itself can be useful for filtering spam on real time search (Benevenuto et al., 2010), whereas the detection of spammers is related with the detection of existent spam accounts. In fact, a way to detect spammers would be filtering users who have written many spam tweets. In addition, when a spam account is detected, Twitter suspends it or even blocks his IP address temporally, so spammers only need to create a different account to continue sending spam messages or wait a while for his IP address is unlocked.

In this paper we propose several new features based on language models to improve spam detection on Twitter trending topics. Language models (Ponte & Croft, 1998) are probabilistic methods that have been previously used successfully in areas of speech recognition, machine translation, part-of-speech tagging, parsing and information retrieval and even in some previous works for the detection of Splogs (Mishne, Carmel, & Lempel, 2005), nepotistic links (Benczúr, Bíró, Csalogány, & Uher, 2006) and Web spam (Araujo & Martinez-Romo, 2010; Martinez-Romo & Araujo, 2009). While probabilistic models have been proposed and studied for information retrieval since as early as 1960's, they had not really shown clear advantages over the traditional vector space model until around 1998, when Ponte and Croft (Ponte & Croft, 1998) published a pioneering work which uses a different probabilistic model for retrieval, i.e., the query likelihood scoring method. Statistical language models have been developed to capture linguistic features hidden in texts, such as the probability of words or word sequences in a language. A statistical language model (SLM) is a probability distribution $P(s)$ over strings S that attempts to reflect how frequently a string S occurs as a sentence (Clarkson, 1997).

The underlying idea is as follows: we examine the use of language in the topic, a tweet, and the page linked from the tweet. In the case of spam tweets, these language models are likely to be substantially different: the spammer is usually trying to divert traffic to sites that have no semantic relation. We exploit this divergence between the language models to effectively classify tweets as spam or non-spam.

Previous works have proved that language model disagreement techniques are very efficient in tasks such as blocking blog spam and detecting nepotistic links and Web spam. For this reason, we want to apply these techniques to improve classification in a spam Twitter labeled dataset of around 34 K trending topics, 21 million tweets and 6 million URLs. We use an extension of the basic language modeling approach (Zhai & Hirst, 2008) to analyze the divergence between the language models of a trending topic and each suspicious message tagged with that topic. We apply Kullback–Leibler Divergence (KLD) (Nederhof & Satta, 2008) between their respective language models. KLD is an asymmetric divergence measure originating in information theory, which measures how bad the probability distribution M_q is at modeling M_d.

The remaining of the paper proceeds as follows: Section 2 presents the previous works in the Twitter spam research area; Section 3 shows the architecture of the system; Section 4 is devoted to the dataset and labeling process; Section 5 describes the analysis of the language model employed in spam detection and the proposed features; Section 6 studies the classification process; Section 7 shows the evaluation methodology; Section 8 discusses the limitations and future work. Finally, Section 9 draws the main conclusions.

2. Previous work

There are several studies that have addressed the detection of spam on Twitter although most of them focus on detecting Twitter criminal accounts individually, instead of focusing on the detection of tweets containing spam. Yardi et al. (2010) studied the behavior of a small group of spammers, finding that they exhibit very different behavior from non-spammers in terms of posting tweets, replying tweets, followers, and friends. Shekar et al. (2010) presented a strategy to filter the pharmaceutical spam on Twitter by specific keywords. The Monarch project at Berkeley Thomas et al., (2011a) used a real-time system to identify link spam in Twitter messages. They implemented a logistic regression classifier and crawl URLs as they are submitted to web services to determine whether the URLs direct to spam. Much preliminary work (Benevenuto et al., 2010; Lee et al., 2010; Stringhini et al., 2010; Wang, 2010; Wang, 2012) relies on account features including the number of followers and friends, text similarities between tweets, URL ratios, and account creation dates, although most of these features can be easily manipulated by spammers. That is why works have emerged recently that try to extract features that are most difficult to simulate. Yang et al. (2011) and Yang, Harkreader, Zhang, Shin, and Gu (2012) focused on relations between spam nodes and their neighboring nodes measuring three graph-based features: local clustering coefficient, betweenness centrality, and bi-directional links ratio. They also introduced other features based on timing and automation. Song, Lee, and Kim (2011) considered the relations between spam senders and receivers such as the shortest paths and minimum cut, because spam nodes usually cannot establish robust relationships with their victim nodes. However, the extraction of these features involve a large consumption of time and resources. Ghosh et al. (2012) have studied a phenomenon in Twitter until now reserved for the web: Link farming. The authors explain how spammers try to acquire large numbers of follower links in the social network in order to modify the ranking of their tweets by search engines. Chu, Gianvecchio, Wang, and Jajodia (2012) propose a classification system based on a set of measurements with a collection of over 500 K accounts with the main goal of observing the difference among human, bot, and cyborg in terms of tweeting behavior, tweet content, and account properties.

In relation to the use of language models for detection of spam on different types of resources on the Web, there have been works such as Mishne et al. (2005), that applied language models to Blog spam detection. Here, the authors estimate language models from the original post and each comment in a Blog and then, they compare these models using a variation on the Interpolated Aggregate Smoothing. In particular, this measure calculates the smoothed Kullback–Leibler divergence between the language model of a short fragment of text (original post) and a combined language model of knowledge preceding this text (previous comments). Benczúr et al. (2006) proposed to detect nepotistic links to Web spam filtering by using language models tested on a 31 M page crawl of the *.de* domain with a manually classified 1000-page random sample. In this method, a link is down-weighted if the language models from its source and target page have a great disagreement. Specifically, they used Kullback–Leibler divergence between the unigram language model of the target and source pages. Then, they feed suspicious edges into a weighted *PageRank* calculation to obtain *NRank*, the "nepotism rank" of the pointed page, which is subtracted from the original *PageRank* value. Martinez-Romo and Araujo (2009) and Araujo and Martinez-Romo (2010) applied a language model approach to different sources of information extracted from a Web page, in order to provide high quality indicators in the detection of Web spam. Their hypothesis was that two pages linked by a hyperlink should be topically related, even though this were a weak contextual relation. For this reason they analyzed different sources of information from a Web page that belongs to the context of a link and applied Kullback–Leibler divergence on them for characterizing the relationship between two linked pages.

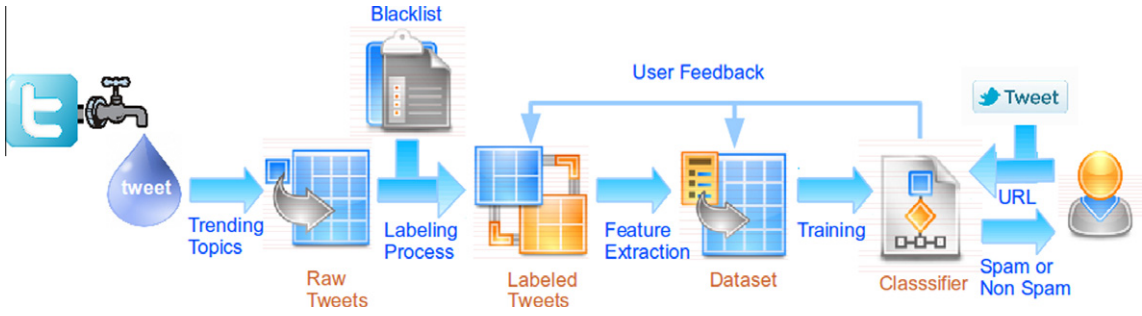

Fig. 1. Architecture of the proposed system for filtering spam tweets in trending topics.

Our system, as we will see, has adapted the analysis of language to a new problem as spam in social networking. Despite the possible similarities with previous work, this scenario introduces new problems such as the small size of the messages, the vocabulary used, the limited context of information, and of course the challenge of real-time filtering messages.

3. Architecture

In this section we present the architecture of the system for filtering spam URLs as they are posted to Tweets. Fig. 1 presents a scheme of the proposed model.

The system consists of five processes:

- **Trending Topics Collection**. First, the system obtains a set of tweets associated with a given trending topic. The trending topics are retrieved periodically to get a heterogeneous set of tweets.
- **Spam Labeling**. The second process is the spam labeling of the trending topics, where the system uses several blacklists[1] to detect spam URLs in tweets and label the collection in this way. The labeled set obtained in this step will be used to train the system and to detect new spam tweets.
- **Feature Extraction**. Then, a features extraction task is performed to represent each labeled tweet using natural language processing and content analysis techniques.
- **Classifier Training**. The final dataset, consisting of the labeled set of tweets and each tweet represented by a set of features, is used by the classifier to train its model and acquire necessary knowledge in the detection of spam.
- **Spam Detection**. The classification algorithm obtains a tweet from a user as input and notifies to the user whether it is spam or not. The user, in turn, can inform the system of a possible error in the classification process. In the case of a supervisor considers the system was wrong may decide to modify the dataset in. Thus the classifier performs a continuous learning process.

4. Trending topics collection and spam labeling

In order to evaluate our approach to detect spam tweets, we need a labeled collection of tweets, pre-classified into spam and non-spam. To the best of our knowledge, no such collection is publicly available. We then built a crawler that using Twitter API[2] methods allowed us to collect trending topics and their associated tweets from 30 April 2012 to 8 May 2012. We used five computers to perform parallel requests, where each computer requested the trending topics and associated tweets from different English-speaking geolocations (Global, US, England, Canada, and Australia). Trending topics are cached for 5 min, so requests for fresh information were made with a delay between them of more than five minutes. The resulting dataset contains around 34 K trending topics, 20 million tweets (restricted to English language) and 6 million URLs.

Then, it is necessary to identify spam tweets from the crawled dataset. Our dataset for training and testing the classifier consists of several sources: Google Safe Browsing,[3] Capture-HPC,[4] SURBL,[5] Project Honey Pot,[6] and flagged URLs provided by Twitter's Link Service.[7] In case a URL posted to Twitter, any of its redirects, frame URLs, or any of its source URLs become blacklisted (Shih, Chiang, & Lin, 2008), we treat the tweet as spam. We clearly acknowledge the limitations of our analyzed dataset and we take into account that it may still contain some bias and the number of spam tweets is a lower bound of the real number. However, even for such a set of spam tweets, we can still use them to test the performance of our system on detecting these spam tweets.

In total, 168 K tweets were labeled as spam, which represents 8% of the analyzed tweets. Strictly we can not say that the rest of tweets are non-spam, however in order to prove our model they will be used as reliable tweets. Since the number of non-spam tweets is much higher than the number of spam, a set of 340 K tweets have been selected randomly to be included in the dataset, what represents twice the number of spam tweets. Thus, the labeled dataset used in our experiments is composed of 508 K tweets (168 K spam and 340 K non-spam).

5. Features extraction

In this section we propose several new features based on language models to improve spam detection on Twitter trending topics. These features are one of the main contributions of our work since the statistical analysis of language has not been used in the detection of spam on twitter up to now, and also provides a method for detection of illegitimate tweets hardly manipulable. The result of the extraction of features will be the final dataset used by the classifier for detecting spam tweets. This dataset will have two sets of tweets tagged as spam or non spam and a set of features to represent each tweet. In the case that a user determines that a tweet has been misclassified, all tweets affected by the same URL may be re-labeled and updated in the dataset. We outline our

[1] A spam blacklist is a list or register of web sites which, for one reason or another, have been classified as spam.

[2] Twitter API. Accessed July 2012, https://dev.twitter.com

[3] Google Safe Browsing API. Accessed July 2012, https://developers.google.com/safe-browsing.

[4] Capture-HPC Client Honeypot/ Honeyclient API. Accessed July 2012, https://projects.honeynet.org/capture-hpc.

[5] SURBL, URI Reputation Data. Accessed July 2012, http://www.surbl.org.

[6] Project Honey Pot. Accessed July 2012, http://www.projecthoneypot.org

[7] Flagged URLs as Malware or Spam by Twitter's Link Service (http://t.co). Accessed July 2012, http://support.twitter.com/articles/109623-faqs-about-twitter-s-link-service-http-t-co.

Fig. 2. Sample of three types of language models used to detect spam tweets: the suspicious tweet, a set of tweets related to a trending topic called thread, and the target page linked by the suspicious tweet.

language model based approach to identify spam tweets below.

5.1. Language models

A language model is a statistical model for text analysis, which is based on a probability distribution over pieces of text, indicating the likelihood of observing these pieces in a language. Usually, the real model of a language is unknown, and is estimated using a sample of text representative of that language. Different texts can then be compared by estimating models for each of them, and analyzing the models using well-known methods for comparing probability distributions.

In this paper, we have used three types of language models corresponding to the three units of text that we believe are involved in the model of spam on Twitter. Specifically these language models are (shown in Fig. 2): a set of tweets related to a trending topic, the suspicious tweet, and the page linked by the suspicious tweet.

One of the most successful methods based on term distribution analysis uses the concept of Kullback–Leibler Divergence (Cover & Thomas, 1991) (KLD) to compute the divergence between the probability distributions of terms of two particular documents considered. In order to compare these models, we have used a variation on the Interpolated Aggregate Smoothing used by Allan, Wade, and Bolivar (2003). This method by estimating maximum likelihood of the unigram occurrence probabilities, calculates the smoothed divergence between the language model of a short fragment of text and a combined language model of knowledge preceding this text. Specifically we look at the differences in the term distribution between two text units computing the KLD:

$$KLD(T_1||T_2) = \sum_{t \in T_1} P_{T_1}(t) log \frac{P_{T_1}(t)}{P_{T_2}(t)} \quad (1)$$

where $P_{T_1}(t)$ is the probability of the term t in the first text unit, and $P_{T_2}(t)$ is the probability of the term t in the second text unit.

Probabilities are smoothed with Jelinek–Mercer method (Chen & Goodman, 1996) which is a linear interpolation of the maximum likelihood model with the collection model, using a coefficient λ to control the influence of each model. The two language models we are comparing are any pair of the three types of language models proposed in Fig. 2: the suspicious tweet, a thread of tweets, and the target page linked by the suspicious tweet. Here we can see a sample of two models used in our work: the model of a thread of tweets (P_{thread}) and the model of the suspicious tweet (P_{tweet})

$$P_{thread}(t) = \lambda_1 P_{ML(thread)}(t) + (1 - \lambda_1) P_{ML(collection)}(t) \quad (2)$$

$$P_{tweet}(t) = \lambda_2 P_{ML(tweet)}(t) + (1 - \lambda_2) P_{ML(collection)}(t) \quad (3)$$

where $ML(\beta)$ is the probability of the term t in β using maximum likelihood. Smoothing is applied by using a general probability model of words on a collection ($P_{ML(collection)}$), obtained from all threads by taking each thread as a unique document.

5.2. Language model based features

One of the main contributions of this work is the extraction of features that measure the degree of divergence between the language models compared. We could have determined a larger number of relations of divergence. However, considering the issue of computational complexity, we have chosen several features that are easy to compute and that are useful in the Twitter spam detection. Moreover, we have used Lucene (Gospodnetic & Hatcher, 2004) to carry out the calculus, which is a source information retrieval library.

We explain the methodology used for the extraction of these features below. The first step is to get every trending topic and its associated tweets. The tweets are divided into reliable and suspects. The suspicious tweets are those that link to a web page and the rest are categorized as reliable. The reliable tweets become part of the thread of tweets language model. The suspicious tweets will be used to the task of classification and to evaluate our system. Then all the URLs in suspicious tweets are analyzed and tweets are classified as spam or non-spam according to the method described in Section 4.

The first feature that we extract compares the divergence between the models of the languages from the thread of tweets and from the text of the suspicious tweet. There are sophisticated techniques used by spammers to insert in tweets the most common keywords to carry out their queries. However, in most cases the topic or the terms used in the spam tweets have no relationship to the adja-

cent posts. This is reasonable, as spammers use robots to insert malicious messages in a undiscriminating way. A variant of this feature we propose is to analyze the ten tweets posted just before and after to capture a possible divergence bubble in the thread. Thus, these 20 tweets, that we call bubble, are used to build the language model instead of the full thread. The computational cost of the extraction of these features is very low, although their effectiveness are not as high as the next feature.

As we said above, the topic or the terms used in the spam tweets usually have no relationship to the adjacent comments. However, there are spammers who use more sophisticated techniques and therefore its detection is more difficult. For this reason, we propose to calculate the divergence between the models of the languages from the thread of tweets and from the text on the page pointed by the suspicious tweet. Thus, if the analyzed thread is about "Justin Bieber" and the linked page is dedicated to the sale of pharmaceutical products, it is reasonable to expect a high divergence between their language models. In this case we also compare the page pointed with the tweets bubble. Thus, we can obtain a more suitable value of divergence with a set of messages whose relationship with the full thread is difficult to discover. The effectiveness of these features is very high, however the computational cost is higher because of having to download each page linked by a suspicious tweet. For this reason, we decided to analyze also the title of the page instead of the whole page. This comparison can be useful from two points of view: to capture some information in case that the page has no content or is impossible to extract the text (flash, etc.), and also to reduce time and cost by making an HTTP request lighter and not having to retrieve the full page.

Finally, we have also decided to analyze the history of the user who posted a tweet in the current thread. Thus, suspicious behaviors can be analyzed with greater perspective. The feature that we propose based on the user profile is the divergence with the previous tweet a user has written in the same thread and also with the subsequent tweet, and then getting the average, if any. Moreover, since each tweet is analyzed independently, we have included the average of all values of divergence previously studied for a user in the same thread.

5.3. Content attributes

In theory, the greater the number of relevant attributes of a classifier, the higher predictive power. For this reason and to complete the features proposed in the previous section, we decided to complement our work with a set of attributes based on the content. Some of these attributes have been previously proposed in the work by Benevenuto et al. (2010). Thus, it is possible to obtain characteristic patterns present in the automatic construction of spam tweets which are not detected by features based on the divergence of content.

These features are based on the following metrics: number of URLs per number of words, number of hash-tags per words, number of words, number of characters, number of URLs, number of hash-tags, number of numeric characters, number of users mentioned, number of words from a list of spam words, number of times the tweet has been replied (RT @username), number of tweets posted in the thread by the same user, and time since the last tweet posted by the same user. In total, we have 12 more features based on the characteristics of the content of tweets.

6. Classification

For the classification tasks, we have used the Weka (Witten & Frank, 2005) software because it contains a whole collection of machine learning algorithms for data mining tasks. In particular we have chosen the following classification algorithms: Decision Tree (C4.5), a decision support tool that uses a tree-like graph or model of decisions and their possible consequences; Naive Bayes, a statistical classifier based on the Bayes' theorem; Logistic Regression, a generalized linear model to apply regression to categorical variables; Support Vector Machines (SVMs) which aims at searching for a hyperplane that separates two classes of data with the largest margin; Decorate, a meta-learner for building diverse ensembles of classifiers by using specially constructed artificial training examples; and finally, Random Forest, an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees. Additional information about the classifiers is available in most standard machine learning texts (Witten & Frank, 2005; Mitchell, 1997).

We have adopted a set of well-known (Castillo, Donato, Gionis, Murdock, & Silvestri, 2007) performance measures in spam research: true positive (*TP* or recall), false positive rate (*FP*), accuracy (*A*), and *F-measure*. *F-measure* combines precision *P* and recall *R* by $F = 2\frac{PR}{P+R}$. For evaluating the classification algorithms, we focus on the *F-measure* as it is a standard measure of summarizing both precision *P* and recall *R*. The evaluation of the learning schemes used in all the predictions of this paper was performed by a fivefold cross-validation. For each evaluation, the dataset is split into 5 equal partitions and trained 5 times. Every time the classifier trains with 4 out of the 5 partitions and uses the fifth partition as test data.

We performed an extensive evaluation with the classifiers presented above that are implemented in the Weka toolkit and the performance of each classifier is shown in Fig. 3. For these experiments we have used only the features based on language models presented in Section 5.2. After several experiments and according to what can be seen in Fig. 3 we decided to use a SVM classifier considering that obtained the best results, and is also a state of the art method in spam filtering tasks (Guzella & Caminhas, 2009). We used the default options of these algorithms, since algorithmic details of these classifiers are beyond the scope of this paper.

6.1. Spam Misclassification Trade-off

In the dataset that we use, the non-spam instances outnumber the spam ones to such an extent that the classifier accuracy improves by misclassifying a disproportionate number of spam instances. Moreover, we think that errors for misclassifying *non-spam* tweets as *spam* do not have the same impact that misclassifying a *spam* tweet as *non-spam*. Thus, we have used a cost-sensitive evaluation for SVM (Witten & Frank, 2005) algorithm implemented in *Weka* for classification, which allows establishing different costs of misclassifying. We have imposed a zero cost to right predictions, and we have set to *spam* tweets misclassified as *non-spam* a cost *C* times higher than *non-spam* tweets misclassified as *spam*. Furthermore, as the aim of this work is to maximize the *F-measure*, we have looked for the value of *C* which maximize these measures. According to several experiments shown in Fig. 4 we have set $C = 5$. For these experiments we have used only the features based on language models presented in Section 5.2. However, we believe that the best value of *C* depends on the system's objectives and the cost mechanism could be applied in other way.

7. Evaluation

In this section, we have evaluated the performance of our machine learning system. Although we wished to compare our results with other works in the area of the detection of spam on Twitter, a meaningful comparison is not possible at this moment primarily because of two reasons: Most studies focus on the detection of spammers instead of spam tweets, and from our point of view those

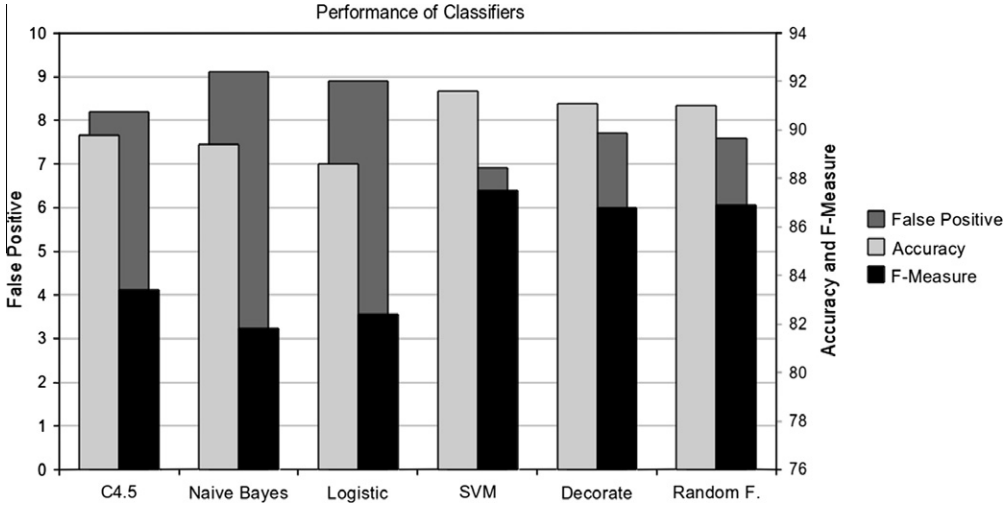

Fig. 3. False Positive Rate (FPR), Accuracy, and F-measure, for different classification algorithms.

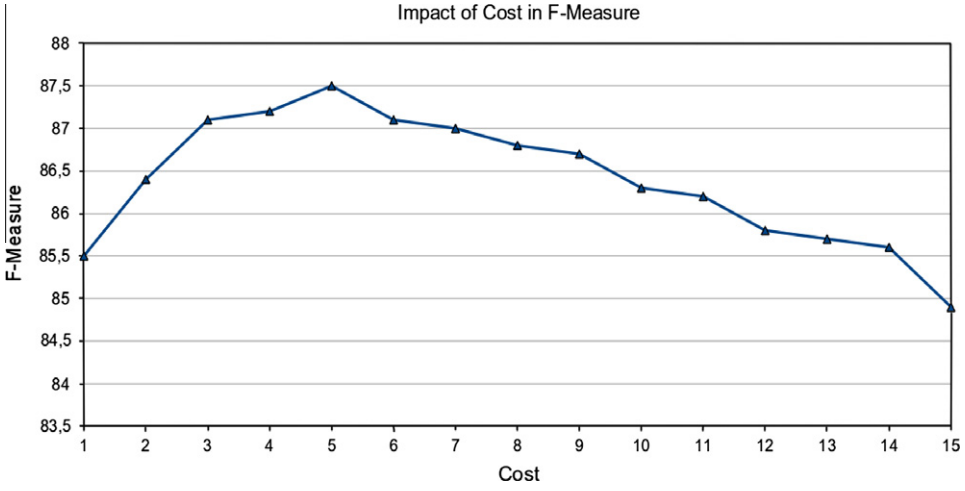

Fig. 4. Evolution of *F-measure* obtained by applying different values to the misclassification C in the confusion matrix.

problems are too different enough to be compared. Furthermore, due to the privacy policies of Twitter, tweets can not be redistributed. This means that there is no public collection with which to make tests and therefore each author uses its own dataset. Thus, each dataset has different size, date and so characteristics.

However Table 1 shows a comparison of the most similar systems to ours although as stated above, this comparison is not fair because the data and the objectives are different in each case. The first observation is that the most similar approach to the one described in this paper is the work of Benevenuto et al. (2010) because it is the only one that performs the detection of spam in tweets instead of user accounts. Nevertheless, besides that the collection used in their work is different, as explained above, and the size of their labeled collection is slightly lower, the recovery of tweets has been carried out taking into account the users that belonged to the same topic. The rest of works add the common differences to the system of Benevenuto besides the differences in the type of detected spam. These works get all tweets from a user and they can make a profile for each user, obtaining more information on the context of a tweet. For the detection of spam tweets, the information available is smaller since each tweet is analyzed in isolation in the context of a topic. These differences are reflected

Table 1
Comparison of different spam detection systems on Twitter. Columns from left to right indicate: the reference to the system (*System*), type of spam detection approach used depending on whether the objective was the detection of spam accounts or spam tweets (*Detection*), F-Measure (*F-Measure*), classification algorithm used (*Classifier*), False Positive rate (*FP*), and sizes (stated in number of profiles or tweets) of the original collection and labeled collection that has been used in the experiments (*Collection Size*).

Spam detection systems on twitter					
System	Detection	F-measure	Classifier	FP	Collection size
Our system	Tweet	0.883	SVM	0.063	20 M/500 K Tweets
Benevenuto et al. (2010)	Tweet	0.872	SVM	0.075	55 M/1 K Users
Yang et al. (2011)	Account	0.884	Random F.	0.060	500 K/5.5 K Users
Wang (2010)	Account	0.917	Naive Bayes	n/a	25 K/500 Users
Lee et al. (2010)	Account	0.888	Decorate	0.057	210 K/1 K Users

Table 2
Confusion Matrix obtained from our machine learning system for predicted spam and non-spam on the dataset with true spam and non-spam.

Twitter spam dataset – confusion matrix		
True	Predicted	
	Spam	*Non-Spam*
Spam	89.3%	10.7%
Non-Spam	6.3%	93.7%

Table 3
Features, True Positive rate (TP), False Positive rate (FP), Accuracy (A), and F-measure (F-Measure) for Twitter Spam classifiers using different feature sets. For each column in the table the best values (highest TP, A and F-measure, and lowest FP) are shown in bold.

Twitter spam dataset – feature set					
Feature set	# Features	TP	FP	A	F-measure
Content (C)	12	79.7	6.6	88.9	82.6
Lang. Models (LM)	8	88.7	6.9	91.6	87.5
C ∪ LM	20	**89.3**	**6.3**	**92.2**	**88.3**

in the Table 1 also showing slightly better results for the works focused on detecting spam accounts. However, our work based on the detection of spam tweets besides obtaining a similar false positive rate to these other studies, has the main advantage that it can be used in a real-time system because it only needs the information of the tweets of that topic and therefore its computational cost is lower. It is also remarkable a curious effect observed in Table 1. Looking at the labeled collection size used in the experiments, it can be seen that the greater the collection, the smaller the F-measure and the False Positive rate. This suggests that the size of the collection can influence the results.

Table 2 illustrates the resulting confusion matrix obtained with our machine learning system when we use the features described above. The total number of features is 20, which implies a reduced set of attributes. However, these features obtain an efficient performance getting results in the state of the art.

From the confusion matrix we would like to emphasize the non-spam and spam correctly classified, that reach values of 89.3% and 93.7%. In addition, only 6.3% of the non-spam tweets were misclassified as spam, reflecting the effect of the cost added to the classification algorithm. However, this value could be reduced even more because in most cases corresponds to non-spam tweets that link to pages not directly related to the whole thread but only to a few tweets sent previously. Continuing with the example of "Justin Bieber", a user asked whether he will sing in LA. Another user linked the events scheduled at the Staples Center and another links from events in LA in response to previous user. This tweet was classified as spam by our system. In this case, although the pointed page was mostly sports events and the tweet could be considered as suspect because of including several links in the same tweet, it should not be classified as spam.

Table 3 shows the results obtained using different sets of features. We can notice how the language-models-based features had a better performance than content-based features, even having a smaller size. However, despite having better values for True Positive rate, Accuracy and F-Measure, the False Positive rate is slightly higher. On the other hand, we see as the union of the two sets obtained better results, taking advantage of the two sets and overcoming the results of each set separately. Thus, we have an F-measure as the best systems in the state-of-the-art, and also reducing spam tweets misclassified as non-spam, which we believe are more harmful. It is also important to note that these features could be applied in a real-time system.

Table 4
Result of the classification of the spam detection system over a set of unlabeled tweets and its manual evaluation.

Non labeled twitter spam dataset			
Collection size	Detected spam	Manual evaluation	
10,116 Tweets	496 Tweets	431 Spam tweets	86.9%
		38 Borderline tweets	7.6%
		27 Non-spam tweets	5.4%

7.1. Manual evaluation over a second dataset

To complete the evaluation process we decided to conduct a manual evaluation on a new set of tweets. The main goal of this new evaluation is to decrease the possible effect of sampling bias and so, we followed an evaluation process similar to that used in Yang et al. (2011), which should also serve to validate the results obtained above.

The new dataset is composed of 10116 tweets retrieved as described in Section 4, although they were not labeled using blacklists to test the performance of the system on a new sample. Thus, we used the dataset described in Section 4 as training set and the new dataset as a test set.

Results of the spam detection system on the new dataset are shown in Table 4. Given that the main problem of spam classifiers is the false positive rate, a group of assessors analyzed the tweets classified as spam. These spam tweets were tagged by a group of volunteers labeling tweets as "non-spam", "spam" or "borderline". In our experiments, we restricted the datasets using only tweets labeled at least by two persons independently, and for which all assessors agreed.

Table 4 illustrates the 496 spam tweets that the system detected on a set of 10,116 unlabeled tweets. In the later evaluation process conducted by assessors, it was proved that the system had detected 431 tweets correctly, 27 incorrectly, and 38 were marked as borderline. Analyzing in depth the borderline tweets, all showed evidence of spam although assessors wanted to reflect doubts on the results, since these tweets did not appear to have malicious intentions. However it was checked that 23 of the borderline spam tweets contained URLs present in the blacklists described in Section 4.

Summing up, the spam detection system is able to detect a 94.5% of spam tweets (spam and borderline tweets), and obtain a false positive rate of 5.4%. These results show a higher performance than those obtained in the first dataset. In our opinion, this is due to the evaluation of the first dataset that is based only on the presence of URLs in blacklists. The blacklists are an important source of information but publishing spam URLs is not immediate and that delay can be a determining factor when analyzing tweets in trending topics. This further strengthens our commitment to have a spam detection system that in addition to having updated blacklists, allows the user to feed back the system with a misclassification.

8. Discussion and future work

We clearly acknowledge the limitations of our analyzed dataset and we take into account that it may still contain some bias and the number of spam tweets is a lower bound of the real number. However, there is no public collection with which to make tests and so we believe that to create a labeled collection requires a great effort and we have tried to use a methodology to assess objectively our system. Despite this, we also believe that the scientific community should work towards finding a common evaluation framework in which to compare the spam detection systems.

In our future work, we will work to select the most appropriate features for use in a detection system in real time or reduce the cost even more. For that, it would be interesting not to analyze each suspicious URL of each tweet, or not treat each tweet with a link as a suspect but only those whose language is divergent. We would also like to analyze the characteristics of each type of spam and see how the proposed features affect the detection of each of these types of spam.

9. Conclusion

In this paper, we present a new methodology to detect spam tweets in Twitter trending topics, which differs from previous works that had focused on the detection of spam accounts. Our study is based on the analysis of the language used in each tweet, to identify those messages whose purpose is to divert traffic from legitimate users to spam websites. Two tools that are available to spammers are the 140 characters in a tweet and the linked pages. In addition, because of growing microblogging phenomenon and trending topics, spammers can disseminate malicious tweets quickly and massively. Thus, we use an extension of the basic language modeling approach to analyze the divergence between the language models of a trending topic and each suspicious message tagged with that topic. We also use other sources of information involved in the performance of Twitter such as the messages around the suspicious tweet or the content and title of the pointed page. The result is a reduced set of features, which used in conjunction with our machine learning system, allows us to obtain an F-measure in the state-of-the-art. Our system reaches values of 89.3% and 93.7% in non-spam and spam correctly classified, and only 6.3% of the non-spam tweets were misclassified as spam. We have also performed a second evaluation test with a new set of unlabeled tweets and a group of assessors in order to a further evaluation of the system. Assessors concluded that the spam detection system was able to detect a 94.5% of spam tweets and obtain a false positive rate of 5.4%.

Our main contributions are the novel use of language analysis to extract some features hardly manipulated by spammers, and a set of new features which are an orthogonal representation of each Tweet and that can be combined with other sets of features to improve spam detection in social networks. Our system is also designed to get the most benefit from the extracted features, adapting the analysis of language to the main characteristics of the tweets and particularly the trending topics. Thus, our system can be applied to spam detection in trending topics in real time mainly because our system focuses on the detection of spam tweets instead of spam accounts.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the project Holopedia (TIN2010–21128-C02–01) and the Regional Government of Madrid under the Research Network MA2VICMR (S2009/TIC-1542).

References

Allan, J., Wade, C., & Bolivar, A. (2003). Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. SIGIR '03.* (pp. 314–321). New York, NY, USA: ACM. http://dx.doi.org/10.1145/860435.860493.

Araujo, L., & Martinez-Romo, J. (2010). Web spam detection: new classification features based on qualified link analysis and language models. *IEEE Transactions on Information Forensics and Security, 5*(3), 581–590.

Benczúr, A. A., Bíró, I., Csalogány, K., & Uher, M. (2006). Detecting nepotistic links by language model disagreement. In *WWW '06: Proceedings of the 15th international conference on World Wide Web* (pp. 939–940). New York, NY, USA: ACM.

Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. (2010). Collaboration, Electronic messaging. In *Anti-Abuse and Spam Conference – CEAS 2010* (pp. 1–9).

Castillo, C., Donato, D., Gionis, A., Murdock, V., & Silvestri, F. (2007). Know your neighbors: web spam detection using the web topology. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 423–430). New York, NY, USA: ACM.

Chen, S. F., & Goodman, J., 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics. ACL '96. Association for Computational Linguistics*, Stroudsburg, PA, USA (pp. 310–318). http://dx.doi.org/10.3115/981863.981904.

Choi, S.-M., & Han, Y.-S. (2012). Representative reviewers for internet social media. *Expert Systems with Applications., 40*(4), 1274–1282.

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2012). Detecting automation of twitter accounts: are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing, 9*(6), 811–824.

Clarkson, P., 1997. Statistical language modeling using the cmu-cambridge toolkit. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.5987>.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York, NY, USA: Wiley-Interscience.

Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., et al. (2012). Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st international conference on World Wide Web. WWW '12* (pp. 61–70). New York, NY, USA: ACM. http://dx.doi.org/10.1145/2187836.2187846.

Gospodnetic, O., & Hatcher, E. (2004). Lucene in Action. *Manning*.

Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. *Expert Systems with Applications, 36*(7), 10206–10222.

Lee, C.-H. (2012). Unsupervised and supervised learning to evaluate event relatedness based on content mining from social-media streams. *Expert Systems with Applications, 39*(18), 13338–13356. <http://www.sciencedirect.com/science/article/pii/S0957417412007841>.

Lee, K., Caverlee, J., & Webb, S. (2010). Uncovering social spammers: social honeypots+machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. SIGIR '10* (pp. 435–442). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1835449.1835522.

Martinez-Romo, J., & Araujo, L. (2009). Web spam identification through language model analysis. In *Proceedings of the fifth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*. ACM.

Mishne, G., Carmel, D., & Lempel, R. (2005). Blocking blog spam with language model disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*.

Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill.

Nazario, J. (2009). Twitter-based botnet command channel. Arbor Networks. <http://asert.arbornetworks.com/2009/08/twitter-based-botnet-command-channel> (accessed April 1, 2010).

Nederhof, M.-J., & Satta, G. (2008). Computation of distances for regular and context-free probabilistic languages. *Theoretical Computer Science, 395*(23), 235–254. <http://www.sciencedirect.com/science/article/pii/S0304397508000388>.

Nishanth, K. J., Ravi, V., Ankaiah, N., & Bose, I. (2012). Soft computing based imputation and hybrid data and text mining: The case of predicting the severity of phishing alerts. *Expert Systems with Applications, 39*(12), 10583–10589.

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275–281). New York, NY, USA: ACM.

Ramanathan, V., & Wechsler, H. (2012). Phishgillnet–phishing detection using probabilistic latent semantic analysis. *EURASIP Journal on Information Security, 2012*(1), 1.

Shekar, C., Wakade, S., Liszka, K., & Chan, C. -C. 29–1 Dec (2010). Mining pharmaceutical spam from twitter. In *10th International conference on intelligent systems design and applications (ISDA), 2010* (pp. 813–817).

Shih, D.-H., Chiang, H.-S., & Lin, B. (2008). Collaborative spam filtering with heterogeneous agents. *Expert Systems with Applications, 35*(4), 1555–1566. <http://www.sciencedirect.com/science/article/pii/S0957417407003715>.

Song, J., Lee, S., & Kim, J. (2011). Spam filtering in twitter using sender–receiver relationship. In R. Sommer, D. Balzarotti, & G. Maier (Eds.), *Recent advances in intrusion detection. Lecture notes in computer science* (Vol. 6961, pp. 301–317). Berlin/Heidelberg: Springer.

Stringhini, G., Kruegel, C., & Vigna, G. (2010). Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference. ACSAC '10* (pp. 1–9). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1920261.1920263.

Thomas, K., Grier, C., Ma, J., Paxson, V., & Song, D. May (2011a). Design and evaluation of a real-time url spam filtering service. In *IEEE Symposium on Security and Privacy (SP)* (pp. 447–462).

Thomas, K., Grier, C., Song, D., & Paxson, V. (2011b). Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. IMC '11* (pp. 243–258). New York, NY, USA: ACM. http://dx.doi.org/10.1145/2068816.2068840.

Villeneuve, N. (2010). Koobface: Inside a crimeware network. Munk School of Global Affairs. Infowar Monitor (JR04-2010).

Wang, A. H. July (2010). Don't follow me: Spam detection in twitter. In *Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT)* (pp. 1–10).

Wang, A. H. (2012). Machine learning for the detection of spam in twitter networks. In M. S. Obaidat, G. A. Tsihrintzis, & J. Filipe (Eds.), *e-Business and telecommunications. Communications in computer and information science* (Vol. 222, pp. 319–333). Berlin Heidelberg: Springer.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (second ed.). Morgan Kaufmann.

Yang, C., Harkreader, R., & Gu, G. (2011). Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In R. Sommer, D. Balzarotti, & G. Maier (Eds.), *Recent advances in intrusion detection. Lecture notes in computer science* (Vol. 6961, pp. 318–337). Berlin/Heidelberg: Springer.

Yang, C., Harkreader, R., Zhang, J., Shin, S., & Gu, G. (2012). 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. WWW '12* (pp. 71–80). New York, NY, USA: ACM. http://dx.doi.org/10.1145/2187836.2187847.

Yardi, S., Romero, D., Schoenebeck, G., & Boyd, D. (2010). Detecting spam in a twitter network. *First Monday, 15*(1), 1–13.

Zhai, C. X., & Hirst, G. (2008). *Statistical Language Models for Information Retrieval*. Morgan and Claypool Publishers.