Detecting relevant tweets in very large tweet collections: the London Riots case study

Joao P. Carvalho
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
Lisboa, Portugal
Email: joao.carvalho@inesc-id.pt

Hugo Rosa
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
Lisboa, Portugal
Email: hugo.rosa@inesc-id.pt

Fernando Batista
INESC-ID
Instituto Universitário de Lisboa
(ISCTE-IUL)
Lisboa, Portugal
Email: fernando.batista@inesc-id.pt

***Abstract*—In this paper we propose to approach the subject of detecting relevant tweets when in the presence of very large tweet collections containing a large number of different trending topics. We use a large database of tweets collected during the 2011 London Riots as a case study to demonstrate the application of the proposed techniques. In order to extract relevant content, we extend, formalize and apply a recent technique, called Twitter Topic Fuzzy Fingerprints, which, in the scope of social media, outperforms other well known text based classification methods, while being less computationally demanding, an essential feature when processing large volumes of streaming data. Using this technique we were able to detect 45% additional relevant tweets within the database.**

I. Introduction

Twitter was originally created in 2006 as a public social networking service enabling users to send and read short 140-character messages. After the "Arab Spring" [1] and other protests and riots occurring between 2010 and 2011, it became clear that important events are often commented on Twitter before they become "public news". This has led to a change in how the public perceives the importance of social networks, and even news agencies and networks had to adapt and start using Twitter as a potential (and some times preferential) source of information. However, using Twitter as a source of information involves many technical obstacles. As of mid 2015, more than 500 millions tweets covering thousands of different topics are published daily. Of these 500 million tweets, it is very unlikely that more than a few thousand, let us say in the range of 0.001%-0.01%, are relevant to a given discussion topic (even major topics). Therefore, filtering which content is relevant for a given discussion topic is far from trivial. Twitter contributes to solve this problem by providing a list of top trends [2] and the hashtag # mechanism: when referring to a certain topic, users are encouraged to indicate it through the use of a hashtag. E.g., "#refugeeswelcome in Europe!" indicates the topic of the tweet is the current refugees crisis in Europe. Websites such as Hashtags.org (https://www.hashtags.org/) make good use of this information to present Twitter trends, e.g. analytics available at www.hashtags.org/analytics/refugeeswelcome/. Other tools such as Twittermonitor [3] can also be used to obtain Twitter trends.

However, according to Mazzia et al. [4] only roughly 16% of all tweets are hashtagged. These numbers have been confirmed by our experiments, and can be partially explained by the fact that 140 characters is often not enough to communicate a thought, and including an #hashtag further aggravates the lack of available space. It is therefore clear that, in order to properly analyze a given discussion topic, it is essential to retrieve as much of the remaining 84% untagged information as possible. Since no other tagging mechanisms exist in Twitter, the process of retrieving tweets that are related to a given topic must use some kind of text classification process. The main goal of this paper is to present a Fuzzy Fingerprints based method that can be used to retrieve relevant tweets without the need for extensive parametrization or the need of expensive annotation of training sets. We use a large database of tweets collected during the 2011 London Riots as a case study to show the application of the proposed technique.

This paper is partially based on previous published work on Fuzzy Fingerprints, originally developed by Carvalho and Homem in [5], and first applied to text by the same authors in [6]. Fuzzy Fingerprint techniques were extended for the task of Twitter Topic detection by the present authors, compared to other classification techniques, and optimized using several private datasets in previous works [7], [8], [9]. Here we present a formalization for the method that includes some final developments concerning the fingerprint creation and the tweet-to-topic similarity function, and test it on a real world problem.

II. Related Work

The first goal of this work is essentially to automatically classify tweets into a set of trending topics. Tweet Topic Detection involves deciding if a given tweet is related to a given topic or a given #hashtag. Basically this can be categorized as a classification problem, albeit one with some particular characteristics that need to be addressed specifically: (1) it is a text classification problem where the texts to be classified are very short texts (up to 140 characters); (2) the number of possible categories is unknown and very large, (3) it fits the Big Data paradigm due to the huge amounts of streaming data.

It is important to address the distinction between the tasks of Topic Classification and Topic Detection. Topic Classification is well-known in Natural Language Processing (NLP) as the task of Text Categorization. It is classically defined by Feldman et al. [10] as finding the correct topic (or topics) for each document, given a restricted set of generic categories (subjects, topics) such as politics, sports, music, etc., and a collection of text documents. In the particular case of Twitter Topic classification, the tweets will often belong to at least one of those categories and it is very rare that a tweet does not fit into any topic. On the other hand, we have what we call Topic Detection, where an attempt is made to determine if a given document (in our case a tweet) is related to a given topic, where the number of possible topics is so large and topics are so unique among themselves, that there is a high probability that a tweet without a hashtag may very well not belong to any of the topics under consideration. Since Topic Detection and Topic Classification end up being very different problems, techniques used for one are not necessarily the best for the other. There are countless works and competitions on Twitter Topic classification (try to fit Tweets into topics such as New, Sports, Music, etc.), but to our knowledge none specifically dedicated to Twitter topic detection. The most similar works to Topic Detection within Twitter are those related with emerging topics, events or trends. In these works the authors use a wide variety of techniques regarding text analysis to find the most common related words and hence detect topics [11], [3], [12], [13], [14]. Note that in our work we assume the existence of trending topics and we aim at efficiently detecting tweets that are related to them despite not being explicitly marked as so. Anoher important difference is that in trending topic detection it is far from critical missing some topic related tweets tweets, while in Twitter topic detection, every single tweet is relevant. In what concerns the used techniques applied to text classification problems, a wide variety of methods has been applied previously, ranging from hand-coded rules to supervised and unsupervised machine learning. Some of the most well-known and commonly applied methods include: K-Nearest Neighbors (kNN), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), term frequency-inverse document frequency (tf-idf). Recently, most of the existing literature on topic detection/classification/trending on twitter, such as those presented above, rely on topic models, being LDA and its variations the most commonly reported technique and lastly the most successful [15]. Other alternatives for short texts classification involve short sentence similarity. This approach is not viable for this task since it is possible to have two tweets without a single common word that are both related to the same topic despite having zero similarity. In this case each new tweet to be classified would need an existing similarly phrased tweet in the training set, which is an unreasonable assumption. In previously published works presenting Fuzzy Fingerprints, we have successfully compared them against the most traditional methods (kNN, SVM, MNB, tf-idf). Other methods have been attempted, namely LDA, but the results when applied to the specific problem of tweet topic detection, were weak unless very extensive parametrization and testing was done *a priori*.

III. Material and Methods

A. Data

Between the 6th and 11th August 2011 thousands of people rioted in several boroughs of London with the resulting chaos generated looting, arson, and mass deployment of police. In the end five people died in what became known as the 2011 London Riots.

A large dataset known as TW-Master was created by The Guardian newspaper via the REST API during the riots [16], and then expanded using the users' timeline. For each user, tweets created after August 1st 2011 were retrieved up to the 3200 tweet limit from REST API statuses/user-timeline limitation. A total of 9,913,397 Tweets were collected from 8819 Twitter users.

Following the event, The Guardian publicly released Twitter data which included a list of 200 influential twitter users based on re-tweets during the riot period. The released dataset contained a total of 1,132,938 tweets that were posted during all August. According to The Guardian, the dataset contains 17795 tweets related with the London Riots. This data set was used for the case study presented in this article.

B. Twitter Topic Fuzzy Fingerprints

Fingerprint identification is a well-known and widely documented technique in forensic sciences. In computer sciences a fingerprint is a procedure that maps an arbitrarily large data item (such as a computer file, or author set of texts) to a compact information block, its fingerprint, that uniquely identifies the original data for all practical purposes, just as human fingerprints uniquely identify people. In order to serve for classification purposes, a fingerprint must be able to capture the identity of a given class. In other words, the probability of a collision, i.e., two classes yielding the same fingerprint, must be small.

Fuzzy Fingerprints were originally proposed for text classification by Homem et al. [6], where they were successfully used to detect authorship of newspaper articles (out of 73 different authors).

For text classification purposes, a set of texts associated with a given class is used to build the class fuzzy fingerprint. As in several other NLP bag-of-words methods, each word in each text represents a distinctive event in the process of building the class fingerprint, and distinct word frequencies are used as a proxy for the class associated with a specific text. In the particular case of Fuzzy Fingerprints, it was found out empirically that the order of the frequency of the words, is far more important than the frequency itself [6], and the weighting of the importance of the rank of each word is calculated using a Fuzzification function that assigns the rank to a fuzzy interval [0,1].

The set of the fuzzy fingerprints of all classes is known as the fingerprint library. Given a fingerprint library, the procedure originally proposed to classify a given text, consisted in

obtaining the text fuzzy fingerprint, and then using a fuzzy inspired similarity function to obtain the class with the most similar fingerprint. In this procedure the text (to be classified) fingerprint was obtained using exactly the same procedure as the class fingerprints, i.e.: (1) obtain the top-*k* most frequent words within the text; (2) order them; (3) apply the fuzzyfying function to each word.

It is possible to see by the description, that conceptually Fuzzy Fingerprints are *k*-sized 2 column arrays, where one column contains one of the top-*k* most frequent words, and the other its correspondent fuzzified value.

In order to adapt the Fuzzy Fingerprints method for tweet topic detection, several procedural changes were proposed in our previous works [7], [8], [9]. The main reason for such adaptation was the limited to 140 characters size of each tweet: it is impossible to create a distinctive fuzzy fingerprint for a single tweet since few, if any words, are repeated within the tweet. Here we propose some minor changes and formalize the process of creation of Twitter Fuzzy Fingerprints and Fingerprint Libraries based on a dataset of #hashtagged tweets, and the respective process of tweet topic detection.

1) Twitter Fuzzy Fingerprint Creation and Fingerprint Libraries: In order to create a fuzzy fingerprint for a given topic, it is necessary to obtain a set of properly classified tweets. In the case of Twitter, such set is usually easily obtained using a set of #hashtags that are used within the given topic. Even though far from all tweets are usually hashtagged, if a topic is worth of attention, the Twitter usage convention is to start hashtagging the topic in order for it to gain relevance (remember that the problem we are addressing is retrieving the usually sizable percentage of tweets that belong to the topic but are not hashtagged). In cases where only a few tweets are available, the fingerprint can be obtained recurring, for example, to newspaper articles or online blogs. It is obviously assumed that all tweets containing the relevant #hashtag are related to the topic

After obtaining the full set of properly classified tweets, i.e., tweets that are #hashtagged, the first step consists in pre-processing the tweets text as indicated in [7]: eliminate words with less than 3 characters. Stopwords are kept and stemming is not performed, since this was deemed to produce best results in all previously tested sets.

The next step consists in computing the top-k word list for each of the #hashtags: all words in the tweets containing #hashtag j are processed to obtain a list of k tuples $\{v_i, c_i\}$ where v_i is the i-th most frequent word and c_i the corresponding count. i.e., we obtain an ordered k-sized list containing the most frequent distinct words for each topic.

Due to the small size of a single tweet, its features should be as unique as possible in order to make the fingerprints distinguishable amongst the various topics. Therefore we propose to also account for the Inverse Class Frequency (icf) of each word existing in all the computed k tuples $\{v_i, c_i\}$. The icf is an adaptation of the well-known Inverse Document Frequency (idf), where topics are used instead of documents to distinguish the occurrence of common words:

TABLE I
FINGERPRINT HASH TABLE BEFORE AND AFTER ICF

Key	Feature	Counter	Feature	ICF
#michaeljackson	dead	4	dead	1.90
	rip	2	rip	0.95
	sing	1	sing	0.48
#haiti	earthquake	10	earthquake	4.77
	rip	5	rip	1.43
	help	1	help	0.17
#derek	show	8	show	3.81
	help	3	australia	0.95
	australia	2	help	0.52

$$icf_v = \log \frac{J}{J_v} \tag{1}$$

In (1), J is the size of the fingerprint library (i.e., the total number of different topics), and J_v is the number of topics where word v is present.

The product of the frequency of word v with its inverse class frequency, $tficf_v = c_v \times icf$, is used to re-order the k-sized word list of each topic.

Table I shows an example of a possible top-k output produced by the algorithm after going through a small training set, when considering a fingerprint size $k = 3$. By multiplying the occurrences of each word per topic with its icf, we obtain the third column of Table I. As expected, the term "help" drops one position in the ranking of words for the topic "#derek", since it was the only word occurring in more than one fingerprint.

$$\mu_{ji} = \begin{cases} 1 - \frac{(1-\frac{a}{k})\times i}{a} & i < a \\ \frac{a(1-\frac{i-a}{k-a})}{k} & i \geq a \end{cases} \tag{2}$$

The next step consists in fuzzifying each top-k list in order to obtain the topic fingerprint. The choice of the fuzzifying function is relevant and can obviously affect the obtained results. In previous works several alternative membership functions were tested, and given previous results [7], we propose the use of a fuzzifying function inspired in the Pareto rule (2) (Figure 1), where roughly 80% of the membership value is assigned to first 20% elements in the ranking, and the remaining 80% of the elements are assigned less than 20% of the membership value. In (2), μ_{ji} is the membership value of the i-th ranked word of class j, k is the fingerprint size, $i = [0, ..., k]$ and $a = 0.2 * k$. For example, for $k = 10$, $a = 2$.

Given the previous processing, we can obtain the fingerprint of class *j*, Φ_j, which is based on the top-k list, and consists of a size-k fuzzy vector where each position contains an element v_{ji} (in this approach v_{ji} is a word of class j, even though it could be another feature), and a membership value μ_{ji} representing the fuzzified value of the rank of v_{ji} (the membership of the rank), obtained by the application of (2). Formally, topic j will be represented by its size k fingerprint $\Phi_j = \{(v_{j1}, \mu_{j1}), (v_{j2}, \mu_{j2}), ..., (v_{jk}, \mu_{jk})\}$. The set of all #hashtag fingerprints will constitute the fingerprint library.

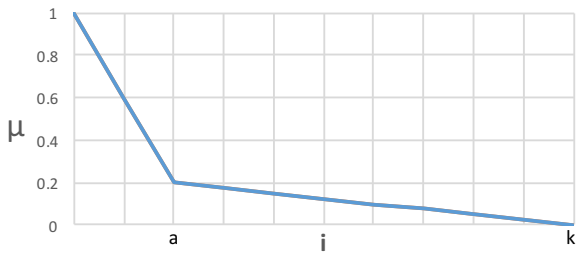

Fig. 1. Fuzzyfing function μ_{ji} that determines the membership value of the i-th ranked word of class j. k is the fingerprint size, and a is 0.2*K

2) Tweet Topic Detection using Twitter Fuzzy Fingerprints: Tweet to Topic Similarity Score: The original text fuzzy fingerprint detection method [6] consisted in creating a fingerprint for each text to be classified, and to compare its fingerprint with all fingerprints contained in the fingerprint library. That method is not applicable to very small texts, such as for example, tweets, since the word frequencies in a single tweet are not distinctive enough to create a fingerprint (within 140 characters very few relevant words, if any, are repeated). In order to address this issue we use a Tweet to Topic Similarity Score (T2S2) that tests how much a tweet fits to a given topic. The T2S2 score (3), does not take into account the size of the text to be classified, but considers its number of distinct words).

$$T2S2(T, \Phi_j) = \frac{\sum_{v_{ji}} \mu_{ji} : v_{ji} \in (T \cap S_{\Phi_j})}{\sum_{i=0}^{\min(\#T,k)} \mu_{ji}} \quad (3)$$

In (3), Φ_j is the fingerprint of topic j, T is the set of distinct words of the preprocessed tweet text, $S_{\Phi_j} = \{v_{j1}, v_{j2}, ..., v_{jk}\}$ is the set of words of fingerprint Φ_j, and μ_{ji} is the membership degree of word v_{ji} in the fingerprint Φ_j. Essentially, T2S2 consists of adding the membership values of every word v that is common between the tweet and the fingerprint Φ_j, and a normalization that consists of dividing it with the sum of the top x membership values of fingerprint Φ_j, where x is the minimum between k (the size of the fingerprint) and the cardinality of T.

T2S2 tends to 1 when most to all features of the tweet belong to the top words of the fingerprint, and approaches 0 when there are no common words between the tweet and the fingerprint, or the few common words are in the bottom of the fingerprint.

Tweets that have a T2S2 score to a given topic above a given threshold, are considered as being relevant to the topic and are retrieved from the database.

3) Parameter Optimization and Previous Results: In our previous work [7], the Twitter Topic Fuzzy Fingerprints performed very well on a set of 2 millions English, Spanish and Portuguese tweets collected over a single day, beating other widely used text classification techniques. In that occasion, the training set consisted of 11000 tweets containing the 22 of the top daily trends. 350 unhashtagged test tweets were properly classified with an f-measure score of 0.844 (precision=0.804, recall=0.889).

As part of a master thesis by Rosa [17], we did further work with a training set of 21000 tweets, from "21 impartially chosen topics of interest out of the top trends of the 18th of May, 2013". The test set was made of "585 tweets that do not contain any of the top trending hashtags" and "each tweet was impartially annotated to belong to one of the 21 chosen top trends". After extensive parameter optimization using a development set, the fuzzy fingerprint method scored an f-measure of 0.833 on the test set, when using k=20 fingerprints, words with less than 3 characters removed, no stopwords were removed and no stemming was performed. Any tweet with a T2S2 score above 0.10 was chosen for retrieval. This setup, proved to be not only more accurate than other well known classifying techniques (kNN, SVM, MNB. tf-idf), but also much faster (177 times faster than kNN, 419 times faster than SVM and at least twice as fast as MNB). An additional advantage consisted in the fact that whenever there is a new tweet to be classified, it is not necessary to build a new classification model (as in MNB).

The described setup (fingerprint size, T2S2 threshold, and text preprocessing parameters) was chosen *a priori* for the current London Riots case study, and should in principle provide good results for other Twitter datasets, since the characteristics of this particular set are very different from any of the previously tested sets. Using all previous parameters is a deliberate option taken to test (and show) if the method is generalizable.

IV. Calculation, Results and Discussion

In this section, we present the results we obtained by applying the proposed methods to the available London Riots dataset.

Using the Twitter Topic Fuzzy Fingerprints method, we created a "London Riots fingerprint" that allowed us to retrieve from the London Riots database, tweets that are relevant but not contained in the 17795 tweets list made public by The Guardian (section III-A). By obtaining a richer set of relevant tweets, it is possible to perform more detailed studies and analysis on the events that occurred in 2011. As an application example, we created a graph representation of the users in the extended set, and determined which users were most important in broadcasting the topic using the PageRank algorithm.

A. London Riots Fuzzy Fingerprint

As it was mentioned in section III-A, the available data set consists of 1,132,938 tweets. The dataset contains thousands of distinct hashtags, but only 4 of those hashtags have enough occurrences and were considered relevant for the purpose of creating the London Riots Fuzzy Fingerprint, namely: #londonriots; #ukriots; #riots; #riotcleanup. In order to make the most out of the London riots topic, the hashtags #londonriots, #ukriots, #riots and #riotscleanup were aggregated into a single #londonriots class.

TABLE II
TRAINING SET TREND DISTRIBUTION

Top Trend	Count
#londonriots	11490
#ukriots	2733
#riots	2332
#riotcleanup	1832
#lfc	1193
#london2012	93
#motogp	0
#eurovision	12
#libya	1517
#f1	898
#mariobrosep	20
#mcfc	628
#theparadigmshift	0
#projectallout	0
#seo	268
#ionlyhaveloveforgod	0
#architecture	0

The REST API's "GET trends/weekly", now deprecated, returns the top trending topics for each day in a given week and was used to select 13 additional #hashtags. They are used to perform the Inverse Class Frequency step introduced in Section III-B1 that allows for a better discrimination. Table II shows the list of topics, some of which, despite being top trends for the days of the London Riots, do not have any tweet occurrences in our database. The low frequency (sometimes zero) of tweets containing the top trending topics can be explained by two main facts: (1) Twitter's own view on what constitutes a trending topic, since the trends list captures the hottest emerging topics, not just what is most popular; (2) A possible bias in the data extraction performed by The Guardian, since the database is mostly based on tweets posted by users considered influential by the Guardian, who might naturally not be fans of, for example, the Eurovision or MotoGP.

The tweets in our dataset that contain at least one of the hashtags in Table II, totaling 23016 tweets, are used for creating fingerprints. This training data contains 18387 tweets related with London Riots and 4629 tweets not related with London Riots, thus being rather unbalanced, i.e., different classes classes/hashtags have different amount of tweets.

The parameter setup used to execute the Twitter Topic Fuzzy Fingerprint method, was the same that studies [7], [17] have shown to be optimal for both performance and speed:

- threshold value for T2S2 = 0.10
- Size of the fingerprint, $k = 20$
- removing words with less than 3 characters from corpus
- not removing stopwords from the corpus
- not performing stemming operations

Table III shows the obtained London Riots fuzzy fingerprint. After obtaining the fingerprint, we tested the similarity of each of the remaining 1,109,922 tweets against it, and marked all tweets having a T2S2 score above 0.1 as being related to the London Riots. The method returned 25757 tweets as being related to the London Riots topic. Based on prior results in several datasets (smaller and unrelated datasets, but diverse),

TABLE III
THE LONDON RIOTS FINGERPRINT

rank	Feature	μ	rank	Feature	μ
1	police	1.00	11	riots	0.13
2	riot	0.80	12	shops	0.11
3	rioters	0.60	13	hackney	0.10
4	cover	0.40	14	#hackney	0.09
5	http://t.co/0hg1bhi	0.20	15	#birminghamriots	0.08
6	croydon	0.19	16	boris	0.06
7	clapham	0.18	17	birmingham	0.05
8	@riotcleanup	0.16	18	army	0.04
9	causes	0.15	19	#manchesterriots	0.03
10	cameron	0.14	20	rioting	0.01

we were confident that the results should be quite good, and an overview of the results indicated it. Since it would be unbearably expensive to check each tweet, we hired an external annotator to create a blind reference annotation in order to evaluate the results and show the applicability of the method. The method proceeded as follows:

We relaxed the T2S2 parameters and applied the method again, knowing that we would probably increase the percentage of True Positives, but certainly obtain a much larger amount of False Positives (we had no idea of how large these increases would be). We obtained 29,053 tweets, which were given to the annotator without any further indications than that he should mark if each tweet was related or not with the London Riots topic. In case of doubt, he should mark them with 'Y?', 'N?' or simply '?'. He was unaware of the dataset balance and what would be a favorable annotation. Since only one annotator was used, many doubts arised and more than 11K tweets were marked with 'Y?', 'N?', or '?'. The 3 authors of this paper then proceeded to individually try to disambiguate such cases. At the end of this process, 7858 tweets were confirmed with 'Y', 1320 annotations 'Y?' 'N?' were reversed, and 1930 tweets left marked with '?'. As a result, the reference dataset consisted of 25,795 'Y' (true positives), 1328 'N' (false positives) and 1930 '?' (doubt). Ideally all those tweets should be 'Y', since that's what we were looking for, but since we used a non-optimal version of the method, we were expecting several False Positives.

The performance depends on how we deal with the unknowns, but even in the most unfavourable case, i.e., if we consider all "?" as False Positives, we are facing a Precision (True Positive Rate) of almost 0.89, and a best case result (where all the "?" are True Positives, of over 0.95 which is not disparate to previous results (remember that this was not the set obtained using the optimal results). The next step consisted in giving the annotator the remaining more than 1M tweets, and ask him to find Positives in that set (they would naturally be considered the experiment False Negatives). Since checking more than 1M tweets using a single annotator is unfeasible, a semi-automatic procedure was used based on the previous annotation, the T2S2 score of each tweet, and several hand-made regular expressions.

From the existing information, a table was produced containing relevant tweet meta-data. The annotation process was

conducted using the following strategy: (1) check the text of individual tweets and validate or correct the initial annotation until finding a possible pattern, either related or not related with London Riots; (2) apply regular expressions to get a list of similar tweets, related with the pattern, and that can be easily checked altogether; (3) Check and mark the list of returned tweets and go back to step 1. The annotator used 3 different tags: "Y" (related with London Riots), "N" (not related) and "?" (not sure). This strategy is very efficient during the initial iterations, where a simple pattern returns big lists of similar tweets that can be check and marked altogether. However, as one proceeded with the annotation, patterns that return similar tweets that have not been previously checked are much more difficult to discover. Simple heuristics, such as looking at the list of words triggered by the fingerprint, or sorting the list of the tweets by their T2S2 score, helped finding and validating more problematic tweets. At the end of this process no additional "Y" tweets were found, but there is obviously no guarantee at all that there are no other False Negatives (even if the number is for sure relatively small).

Based on the reference annotation, we validated the 25757 tweets obtained while applying the optimal pre-defined parameters and all tweets were valid, indicating a Precision (True Positive Rate) of 1 (i.e. no False Positives were retrieved). There were at least 38 False Negatives (detected according the first step of the annotation process), and likely several others in the remaining almost 1.1M tweets. Nevertheless such number will never be in the order of thousands (or they would have been detected during the second annotation phase and during several random smapling procedures). This hints to a very interesting Recall value (TP/(TP+FN)). Even if there are an additional 5000 FN (which we think is very unlikely), the Recall would be close to 0.85. The most optimistic case (which we obviously do not support), would give a Recall of 25757/25795=0.9985, and corresponds to the Recall in the evaluated set. Even though these numbers are very high compared to the overall results presented in [7], it should be noted that similar results were obtained for some of the topics in that dataset, and also to the overall result in the dataset presented in [9].

Despite the uncertainty on the obtained Recall results, the overall result, and the most relevant to the problem in question, is that we were able to retrieve with 100% Precision a group of 25757 tweets related to the London Riots topic (from a set of 1,109,922 tweets), which represents an increase of 45% relevant tweets from the 17795 mentioned by the Guardian.

V. Conclusion

This work uses Twitter Topic Fuzzy Fingerprints to process The Guardian's London Riots Twitter database. This method allowed us to expand the number of tweets considered relevant for the events of the 2011 London Riots by 45% with a precision of virtually 1, confirming the high effectiveness of the method when applied to text social network mining in the specific task of detecting if a tweet is relevant to a given discussion topic. As a result of the process, we obtained an extended dataset, composed of 25757 tweets (compared to the original 17795) that can be used in future studies.

Acknowledgment

Work supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under reference UID/CEC/50021/2013.

References

[1] C. Huang, "Facebook and twitter key to arab spring uprisings: report," http://www.thenational.ae/news/uae-news/facebook-and-twitter-key-to-arab-spring-uprisings-report, June 2011, accessed: 2014-05-02.

[2] Twitter, "To trend or not to trend," https://blog.twitter.com/2010/trend-or-not-trend, 2010, accessed: 2014-03-28.

[3] M. Mathioudakis and N. Koudas, "Twittermonitor: Trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 1155–1158.

[4] A. Mazzia and J. Juett, "Suggesting hashtags on twitter," Master's thesis, University of Michigan, 2010.

[5] N. Homem and J. P. Carvalho, "Mobile phone user identification with fuzzy fingerprints," in *EUSFLAT-LFA 2011*. Atlantis Press, Jul. 2011, pp. 860–867.

[6] ——, "Authorship identification and author fuzzy fingerprints," in *30th Annual Conference of the North American Fuzzy Information Processing Society*, ser. NAFIPS2011, 2011.

[7] H. Rosa, F. Batista, and J. P. Carvalho, "Twitter topic fuzzy fingerprints," in *WCCI2014, FUZZ-IEEE, 2014 IEEE World Congress on Computational Intelligence, International Conference on Fuzzy Systems*, ser. IEEE Xplorer, Beijing, China, July 2014, pp. 776–783.

[8] H. Rosa, J. P. Carvalho, and F. Batista, "Twitter topic fuzzy fingerprints," *IEEE International Conference on Fuzzy Systems'2014*, 2014.

[9] ——, "Detecting a Tweet's Topic within a Large Number of Portuguese Twitter Trends," in *3rd Symposium on Languages, Applications and Technologies*, ser. OpenAccess Series in Informatics (OASIcs), M. J. V. Pereira, J. P. Leal, and A. Simões, Eds., vol. 38. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014, pp. 185–199.

[10] R. Feldman and J. Sanger, *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York, NY, USA: Cambridge University Press, 2006.

[11] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015. [Online]. Available: http://dx.doi.org/10.1111/coin.12017

[12] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, ser. MDMKDD '10. New York, NY, USA: ACM, 2010, pp. 4:1–4:10.

[13] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani, "Emerging topic detection using dictionary learning," in *20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: ACM, 2011, pp. 745–754.

[14] A. Saha and V. Sindhwani, "Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization," in *5th ACM Intern. Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012, pp. 693–702.

[15] M. D. Hoffman, D. M. Blei, and F. R. Bach, "Online learning for latent dirichlet allocation." in *NIPS*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 856–864.

[16] K. Crockett and R. Styles, "Twitter riot dataset," 2011.

[17] H. Rosa, "Topic detection within social networks," Master's thesis, Instituto Superior Tecnico, 2014.