Data Science and Engineering (2018) 3:86–100
https://doi.org/10.1007/s41019-018-0066-4

A Frequent Named Entities-Based Approach for Interpreting Reputation in Twitter

Nacéra Bennacer Seghouani[1] · Francesca Bugiotti[1] · Moditha Hewasinghage[2] · Suela Isaj[3] · Gianluca Quercini[1]

Received: 4 April 2018 / Revised: 29 May 2018 / Accepted: 30 May 2018 / Published online: 13 June 2018

Abstract

Twitter is a social network that provides a powerful source of data. The analysis of those data offers many challenges among those stands out the opportunity to find reputation of a product, a person or any other entity of interest. Several approaches for sentiment analysis have been proposed in the literature to assess the general opinion expressed in tweets on an entity. Nevertheless, these methods aggregate sentiment scores retrieved from tweets, which is a static view to evaluate the overall reputation of an entity. The reputation of an entity is not static; entities collaborate with each other, and they get involved in different events over time. A simple aggregation of sentiment scores is then not sufficient to represent this dynamism. In this paper, we present a new approach to determine the reputation of an entity on the basis of the set of events in which it is involved. To achieve this, we propose a new sampling method driven by a tweet weighting measure to give a better quality and summary of the target entity. We introduce the concept of Frequent Named Entities to determine the events involving the target entity. Our evaluation achieved for different entities shows that 90% of the reputation of an entity originates from the events it is involved in and the breakdown into events allows interpreting the reputation in a transparent and self-explanatory way.

Keywords Reputation · Named Entities · Frequent itemsets · Sampling · Twitter · Opinion mining

1 Introduction

Twitter has become one of the most popular social media platforms at the moment. It includes 1 billion user accounts and millions of active users who post information about their daily life or about relevant events. A Twitter user can follow any number of other users. Tweets from a user with a high number of followers have a larger audience and subsequently a higher visibility. As a result, these tweets are more probable to be seen, liked or retweeted. Information is spread through Twitter by means of retweets and favorites. Therefore, the more retweets and the more favorites a tweet gets, the more it spreads, as it gets more audience. There has been several incidents where Twitter has been faster in spreading information than the mainstream media.

Given that any kind of information can be posted and shared, it is possible to filter out tweets related to a person, a product, an organization or any other entity of interest. Data extraction through crawling or querying depends on

✉ Francesca Bugiotti
francesca.bugiotti@lri.fr

Nacéra Bennacer Seghouani
nacera.bennacer@lri.fr

Moditha Hewasinghage
moditha@essi.upc.edu

Suela Isaj
suela@cs.aau.dk

Gianluca Quercini
gianluca.quercini@lri.fr

[1] LRI, CentraleSupélec, Paris-Saclay University, 91190 Gif-sur-Yvette, France

[2] Universitat Politecnica de Catalunya, 08034 Barcelona, Spain

[3] Aalborg University, 9220 Ålborg, Denmark

the APIs provided by Twitter, and retrieving relevant data is a challenge due to the noise such as spam and false information. The opinion about an entity, held by the public, is widely known as *reputation*. Natural language processing techniques adapted for short texts, abbreviations and emoticons are widely used for sentiment analysis expressed in tweets [4, 8, 13, 20]. Most of these extend the overall reputation of a given entity. The reputation is evaluated by aggregating the sentiment scores of individual tweets in which the target entity is involved or by breaking down the tweets into topics.

Nonetheless, entities collaborate with each other and get involved in different events over time. Therefore, *the reputation of an entity is not static but rather quite dynamic*; especially, public figures are the typical example of reputation influenced by events. For instance, the involvement of a public figure in a charity event arises positive feelings, while a scandal creates negative sentiments. The existing sentiment analysis techniques would aggregate both events (the charity and the scandal) to a neutral sentiment. Hence, the simple aggregation discards the dynamism of the entity by cutting the links with the original events which contributed to the reputation.

Even when we consider entities such as products, movies or organizations, their reputation is highly correlated with the events and the entities they are part of. Thus, the overall reputation or the public opinion of an entity is dependent on the other entities it is involved with and the events occurring at that time.

In this paper, we present a new approach to determine the reputation of an entity on the basis of the set of events in which it is involved. This work extends the paper we presented in [3]: (1) including more related work; (2) giving more details about the algorithms, specifically about the sampling algorithm and its parameters; (3) studying the correlation between parameters of interest and their relation to the entities; (4) and finally, extending the experiments by analyzing another kind of entity of interest, products.

Our main contributions are the following:

- We propose a new sampling method driven by a tweet weighting measure to give a better quality and summary of the target entity. This measure is based on the influence the tweets have on the audience by taking into account the retweets, the number of followers and the favorites. The more a tweet is liked, the more it is retweeted and the more followers its owner has, the higher the weight will be.
- Besides, we introduce the concept of Frequent Named Entities (FNEs) to determine the events involving the target entity. Using frequent entities, we interpret the reputation of a given entity in a self-explanatory way, through the events it is involved in.
- Our evaluation achieved for different kind of target entities: persons, products and movies. The results show that 90% of the reputation of an entity originates from the events it is involved in. The breakdown into events allows interpreting the reputation in a transparent and self-explanatory way. Moreover, the sampling method improves the interpretation of the reputation since the weighted sample technique yields richer information by being able to discover more events.

The remainder of the paper is organized as follows. In Sect. 2, we discuss the related work, and a detailed description of our approach is provided in Sect. 3. We present our experimental results in Sect. 4, and we conclude in Sect. 5.

2 Related Work

Twitter has been used broadly for gathering information about an entity of interest. Characterizing an attitude as positive, negative or neutral toward a topic is known as *sentiment analysis*. Most of the contribution in the field focuses on finding sentiments in the tweet level [1, 5, 13, 25], some of them suggest aggregating the sentiments as a simple sum [4, 7, 17, 22, 24], while the problem of the reputation of an entity has not been specifically addressed.

Natural language processing is a well-established research area in computer science. There is a lot of research carried out in understanding sentiment and emotions using natural language techniques. Sentiwordnet [8] uses the synonym set or synset in WORDNET to give three numerical scores to describe how positive, negative or objective are the terms that are contained in it. The analysis of sentiments has been proven to be useful in reputation management and marketing [14]. Furthermore, the trends in micro-blogging sites like Twitter are actually correlated with the real-world scenarios [5]. There are Internet slang and acronyms that are vastly used in tweets which carry valuable information in understanding the underlying sentiment. Moreover, the use of emoticons contributes to the sentiment [15]. In [13], they have used Internet-specific acronyms, emoticons and domain-specific text processing to successfully detect the sentiments of tweets and classify them into 3 categories with the help of naive Bayes classifiers. Unigram model has been compared to tree kernel and senti-feature-based model, proving that both the latest outperforms the first [1].

Machine learning techniques prove to be effective with sentiment analysis: a semi-supervised approach that uses an interpolation between a universal labeled training set as a base, processed with SVM, and then a topic-related unlabeled training set for enrichment, processed with LDA in [25] and naive Bayes that uses topic-related clusters in [22]. In order to augment the accuracy of the classifier, semantic sentiment analysis is used in [19]. Emoticons, repeated letters or acronyms have been used in [4] to aggregate the sentiments of the tweets, related to a product. Domain-dependent sentiment analysis has been studied in [29] and the effect of hashtags in assigning sentiment scores to tweets in [24]. Sentiment strength or sentistrength has been developed to extract the sentiment of tweets [20], and it was also used in My Space, another social network [21] proving to be quite powerful in both. The tool takes into consideration emoticons, repeated letters, phrasal verbs and everyday expressions, exclamation marks and repeated punctuation. It has incorporated a misspelling correction algorithm and trained by machine learning techniques. Sentistrength shows a higher accuracy compared to several other learning methods.

In this context, we decided to exploit sentistrength in our work as a base tool for sentiment extraction from tweets.

Identifying products and persons is explored in [7] using pattern discovery and mining of comparative sentences inside blogs, forums and product reviews. In [17], different entities are further classified into topics (using hashtags) and the overall opinion is summarized based on the different topics. In contrast to both approaches, we are interested in finding the reputation of an entity in Twitter, which is based on news, events and activities. Therefore, exploring events is needed besides the traditional methods of sentiment analysis or text mining. Thus, instead of mining hashtags or words, we mine entities. Named Entity Recognition techniques are used since they find the entities, locations, companies, etc, involved in the event. Moreover, our contribution focuses on presenting the opinion about an entity, exploring itemset mining techniques with the Named Entities that co-occur together. We show that the Named Entities prove to be quite powerful in opinion summarization.

3 Approach

Figure 1 depicts the overall approach. To extract reputation of people and products from Twitter, the first step consists in querying Twitter to retrieve data related to a target entity E and in extracting the set of Named Entities from the collected tweets. The overall dataset is then described using a Twitter representation model we defined to represent the tweets, the users and the entities. This model is detailed in Sect. 3.1. In the second step, we enrich the information in the tweets to retrieve the tweets that have influence on the audience using a sampling algorithm detailed in Sect. 3.3. In the third step, on top of the sampled data, we apply frequent itemset mining algorithm to extract the Frequent Named Entities (FNEs) related to the entity of interest E. This step is presented in Sect. 3.4.

3.1 Twitter Data Representation

Twitter data can be seen as a network characterized by a high interconnectivity between users and tweets. Each user or tweet of this network is rich of attributes. More formally, Twitter data are represented as a graph as follows:

$$\mathcal{T} = \langle \mathcal{V}, \mathcal{U} \rangle \quad (1)$$

where $\mathcal{V}$ is the set of nodes and $\mathcal{U}$ is the set of directed edges between nodes. Different types of nodes are defined in $\mathcal{V}$:

- t is a *tweet*, accompanied by attribute values, which include the text of the tweet, the id of the tweet, the number of favorites and the number of retweets.
- u is a *user* with attributes as username and number of followers.
- h is a *hashtag* extracted from the tweet.
- e is an *entity* discovered in tweets.
- url is an *url* found in a tweet.

Different types of directed edges are defined in $\mathcal{U}$:

- $<u, t>$ is an edge from u to t with the label "has tweeted".
- $<t, h>$ is an edge from t to h with the label "has hashtag".
- $<t, e>$ is the edge from t to e with the label "has entity".
- $<t, url>$ is the edge from t to url with the label "has url".

3.2 Problem Formulation

Twitter graph provides rich information about events that involve an entity of interest and its relation to other entities. It is rather naive to interpret the reputation of an entity of interest E by the means of a simple aggregation. A simple

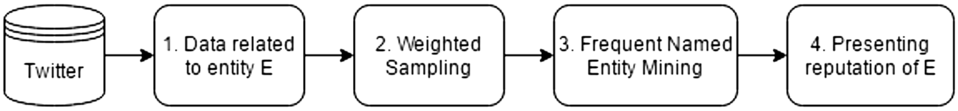

Twitter
1. Data related to entity E
2. Weighted Sampling
3. Frequent Named Entity Mining
4. Presenting reputation of E

Fig. 1 Overall view of the approach

aggregation of the sentiment of tweets provides an overall sentiment of the entity but loses the links to the events in which E is involved.

The *reputation interpretation* problem aims to describe the reputation of an entity E from the events where E participates. More formally, the reputation interpretation has the following input and output:

- *Input* The Twitter graph filtered by the entity of interest E $\mathcal{T}_E = <\mathcal{V}, \mathcal{U}>$
- *Output* The reputation of an entity E defined as follows in Eq. 2:

$$R_E = \{(i_k, r_k) | 1 \leq k \leq n\} \quad (2)$$

where i_k is a frequent set related to the target entity E and r_k is the associated reputation.

For instance, let us consider an entity E and the Twitter graph $\mathcal{T}$. After applying weighted sampling (step 2 in Fig. 1), we get 5 tweets where the entities A, B, C and D appear as represented by the following edges:

- $<t_1, A>$, $<t_1, B>$
- $<t_2, B>$, $<t_2, C>$, $<t_2, D>$
- $<t_3, A>$, $<t_3, B>$, $<t_3, C>$, $<t_3, D>$
- $<t_4, A>$, $<t_4, B>$, $<t_4, D>$
- $<t_5, A>$, $<t_5, B>$, $<t_5, C>$, $<t_5, D>$

Let us suppose that the sentiment analysis gave the following results:

- $t_1 : [+40; -60]$
- $t_2 : [+50; -50]$
- $t_3 : [+25; -75]$
- $t_4 : [+30; -70]$
- $t_5 : [+40; -60]$

The numbers show the positive sentiment versus the negative one. For example, we can interpret it as tweet t_1 is 40% positive and 60% negative.

Let us assume that the frequent set of entities are i_1, i_2, i_3, i_4 and i_5, we can calculate the reputation r_k of each frequent set i_k as the normalized sum of sentiments of the tweets. For instance, i_1 is contained in t_2 and t_3. Hence, r_2 is based on sentiments of t_2 and t_3. Finally, we can show the reputation of E as:

$$R_E = \begin{cases} i_1 = \{A, B, C\}, r_1 = [+32.5; -67.5], \\ i_2 = \{A, B, D\}, r_2 = [+31.7; -68.3], \\ i_3 = \{A, C, D\}, r_3 = [+32.5; -67.5], \\ i_4 = \{B, C, D\}, r_4 = [+32.5; -67.5], \\ i_5 = \{A, B, C, D\}, r_5 = [+32.5; -67.5] \end{cases}$$

Note that the reputation could be seen as a collection of events, accompanied by the sentiment. For example, the sentiment of the event that involved the frequent set $i_1 = \{A, B, C\}$ is $+32.5\%$ positive and -67.5% negative.

3.3 Weighted Sampling

Sampling data have widely been addressed to retrieve data from Twitter due to the immense number of data flowing through daily. Twitter has provided a REST API[1] which allows running queries against the data to retrieve a sample of the actual content on Twitter. But the REST API has a rate limit which allows users to query in 15-min windows. Twitter also has a streaming API to listen to a 1% sample of the live Twitter feed. But using the streaming API requires a lot of bandwidth and storage space and no historical data are available as it is realtime.

Sampling techniques are discussed in a vast number of papers. The most important question that we should pose is: "Do we went a statistically representative sample that aligns with the real, large Twitter dataset or do we want a filtered sample that focuses on the relevant tweets?". Several papers have contributed to finding a statistically representative sample [9, 11, 16, 18, 23, 26–28] using different approaches such as breadth-first search, random walk, unbiased sampling and expert sampling. Since we want to find the reputation of an entity and our goal is the richness and relevance of the sample, we find these methods not suitable. The idea of focused crawling related to a specific topic, based on weights, has been used in [12] and expert sampling in [10]. The study provided by [6] underlines the importance of the retweets and of the mentions in judging about the influence of the users. Inspired by this work, retrieving tweets related to a specific entity is done through querying Twitter with a keyword by considering the three main following parameters that influence the quality of the tweet:

- *The number of times the tweet is retweeted* People retweet information that they agree on and they want to spread in the crowd.
- *The favorite count of the tweet* Marking a tweet as *favorite* is an expression of approval.
- *The number of followers of the user that has tweeted* Number of followers indicate the penetration of the user in the crowd, in the meantime it represents the interest of the crowd on the user.

In order to sample weighting on the aforementioned parameters, we defined two algorithms; Algorithm 1 provides the procedure of weight calculation in tweet level, taking into account how many times the tweet is retweeted and is marked as favorite and how many followers the user has. The absolute values of these parameters need to be

[1] https://dev.twitter.com/rest/public.

well scaled to be integrated into a weighting function. Moreover, the intervals of the values vary depending on the case. Therefore, we do not study this impact. Rather, we use a simple ranking approach to weight our tweets, which ranks the tweets individually for each parameter and then averages the ranking to conduct a weight. Algorithm 2 generates a biased sampling toward the weight of the tweets.

In order to calculate the weight, Algorithm 1 uses the tweets gathered by step 1 in Fig. 1 and their respective parameters of interest. We select a parameter p of interest, for instance the retweets, to assign a tweet t_i a ranking value ρ_i^p according to p value (line 3). If t_1 has more retweets than t_2, a higher ranking will be assigned to t_1 compared to t_2. We iterate this procedure for all parameters of interest. An average of rankings of all parameters is proposed as a merged metric for all rankings (lines 7–9). Then the weight w_i of each tweet t_i is calculated as in line 13.

Our Algorithm 2 promotes a dynamic approach of selecting tweets in random, biasing on their weight. The input of this algorithm is $\{(t_i, w_i)\}$, produced by Algorithm 1. To define if t_i will be selected or not, it generates a random number ω (line 2) and then compares w_i to ω (line 3). It is obvious that high weighted tweets have more chance to be selected.

Algorithm 1 Weight Calculation Algorithm

```
Input: Set of tweets related to $E$ from graph $\mathcal{T}$ with their parameter values
Output: Set of tweet and weight pairs $\{(t_i, w_i)\}$
 1: for each p do
 2:    for each $t_i$ do
 3:       $\rho_i^p \leftarrow$ rank of $t_i$ according to $p$ value
 4:    end for
 5: end for
    max $\leftarrow 0$
 6: for each $t_i$ do
 7:    $\rho_i \leftarrow \sum_{p=1}^{n} \frac{\rho_i^p}{n}$
 8:    if $\rho_i >$ max then
 9:       max $\leftarrow r_i$
10:    end if
11: end for
12: for each $t_i$ do
13:    $w_i \leftarrow \rho_i/\max$
14: end for
15: return$\{(t_i, w_i)\}$
```

Algorithm 2 Weighted Sampling

```
Input: $\{(t_i, w_i)\}$ produced from Algorithm 1
Output: A sample of tweets $S$
 1: for each $(t_i, w_i)$ do
 2:    Let $\omega$ be a random number $[0, 1]$
 3:    if $w_i > \omega$ then
 4:       add $t_i$ to S
 5:    end if
 6: end for
 7: return $S$
```

3.4 Reputation of Frequent Named Entities

The sample S retrieved from Algorithm 2 will be used to find the reputation of the entity E, aided from the frequent entities in S [2]. Named Entities (NEs) carry valuable information as they represent people, location, time and monetary values. Considering a tweet t as a transaction containing a set of entities e as items including the entity of interest E, we define the following concepts:

- Itemset: set of Named Entities that appear together in a tweet
- Frequent itemset: set of Named Entities that frequently appear together in a tweet
- Support of a itemset: the percentage of tweets of S that contains the itemset

We introduce the notion of Frequent Named Entities as follows:

Definition 1 A Frequent Named Entity (FNE), denoted i_k, is a set of e that is maximal according to a predefined support in S.

A FNE i_k describes an event associated with a reputation r_k defined as follows:

Definition 2 Sentiment of the reputation of an event i_k, denoted r_k, will be the ratio between the sum of all positive sentiments pos_i and the negative sentiments neg_i of the tweets t_i that contain i_k.

$$r_k = \left[\frac{\sum_i pos_i}{\sum_i pos_i + \sum_i neg_i}, \frac{\sum_i neg_i}{\sum_i pos_i + \sum_i neg_i}\right]$$

The sum of all underlying positive sentiments of tweets, as well as the sum of the negative ones, can be transformed into normalized proportions that indicate the reputation of an entity E.

Definition 3 The reputation of an entity E, denoted R_E, is the set of events i_k where E is involved, accompanied by their sentiment of reputation r_k. Formally: $R_E = \{(i_k, r_k)|1 \leq k \leq n\}$.

We propose finding FNEs and interpret the reputation of the entity of interest E by its corresponding FNEs and their reputation. This approach is described in Algorithm 3. We intend to find the tweets that contain the FNEs and aggregate their sentiment (lines 5–10). Note that in line 10 we use the normalized definition of reputation, in order to avoid misleading that comes from absolute values of sentiments. Algorithm 3 outputs the reputation of E, R_E, as defined in Definition 3.

Algorithm 3 Calculating aggregated reputation of the FNEs

```
Input: Set of $\{t_i \in S\}$ with their sentiment and set of entities $\{< t_i, E_i, pos_i, neg_i >\}$
    where $E_i \leftarrow \{e|\exists < t_i, e > \in \mathcal{V}\}$. Set of FNEs $\{i_k\}$ explored in $S$
Output: $R_E = \{(i_k, r_k)|1 \leq k \leq n\}$
 1: $R_E \leftarrow \emptyset$
 2: for each $i_k$ do
 3:   $pos_k \leftarrow 0$
 4:   $neg_k \leftarrow 0$
 5:   for each $< t_i, E_i, pos_i, neg_i >$ do
 6:     if $i_k \subseteq E_i$ then
 7:       $pos_k \leftarrow pos_k + pos_i$
 8:       $neg_k \leftarrow neg_k + neg_i$
 9:     end if
10:     $r_k \leftarrow \left[\frac{pos_k}{pos_k+neg_k}, \frac{neg_k}{pos_k+neg_k}\right]$
11:     add $< i_k, r_k >$ to $R_E$
12:   end for
13: end for
14: return $R_E$
```

In this way, the reputation of an entity can be explored through the relations it has with other entities. The data manage to explain itself about the reputation extraction because the information is transparent to the user. Since Algorithm 3 provides normalized values of reputation, each $<i_k, r_k>$ can be accompanied by their support in S to express the coverage of this opinion in the dataset.

4 Experiments

In this section, we run experiments to evaluate our approach in different aspects such as the richness of the samples, effectiveness of frequent entity mining, and comparing the ranking of the sample to the population.

Retrieving data from Twitter can be overwhelming due to the immense number of data flowing through daily. Twitter has provided a REST API which allows running queries against the data. The REST API data are a sample

Table 1 Dataset characteristics

Dataset	Total tweets	Average retweets	Average favorites	Average followers
Obama	15418	873.841	255.787	51983.204
Trump	8918	662.575	243.960	43305.510
La La Land	18852	338.720	31.686	15328.642
The Voice	6864	1286.715	137.539	36492.422
Samsung	2069	418.317	32.42	55233

of the actual content on Twitter. Twitter also has a streaming API which allows users to listen to a 1% sample of the live twitter feed. We use the REST API to collect the data since we are interested in older tweets, as well as users. To collect data about a certain topic, we used a query having a string as parameter (such as `Obama`). In [3], the datasets of *Trump*, *Obama* are collected in January 2017, *La La Land* and *The Voice* in March 2017, while for the new entity, *Samsung*, the data are collected in March 2018. Neo4j[2] graph database for data storage, respecting the Twitter graphs definition of Sect. 3.

We improved the quality of the text by separating merged words inside the hashtag. For example, `#iamsohappy` and `#iam#sohappy` will be handled by our cleaning algorithm to produce `i am so happy`. We are using the corpus of words of sentistrength[3] for word identification and then different techniques for organizing the sentence and discarding not relevant words.

We have used Stanford NLP[4] to identify the Named Entities from the retrieved tweets, after identifying the entities e stored in the same database as separate nodes with the edge $<t, e>$ relating to the tweet t. Stanford NLP also has a sentiment analysis module. But there have been tools that are designed and optimized for small text sentiment analysis specially for Twitter. In this paper, we have used sentistrength as the sentiment analysis tool. Sentistrength scores a given text with a positive and negative value. The text will have both a negative and a positive score from -5 to -1 and 1–5. For instance, if a text is highly negative, it will have a score of -5 and 1 indicating that there is no positive score but a high negative value. Once the sentiment analysis is done, the scores are also stored in the database for each of the tweets as parameters. In the following, we firstly present statistics about the collected data and then the results provided by each step.

Table 2 Dataset characteristics related to NE

Dataset	Density of NE	Distinct NE	Coverage of NE
Obama	1.818	1865	0.916
Trump	1.888	1559	0.897
La La Land	1.382	1948	0.630
The Voice	1.061	1124	0.547
Samsung	1.81	454	0.78

4.1 Data Analytics

In this section, we analyze our datasets gathered through REST API from Twitter. In Sect. 4.1.1, we run general statistics about the size of the dataset, number of retweets, number of followers etc, as well as advanced statistics regarding the richness of the datasets. Later, we continue with the correlation of our parameters of interest in Sect. 4.1.2 and their relation to Named Entities in Sect. 4.1.3.

4.1.1 Statistics of Datasets

We collected four datasets of tweets: *Obama* dataset, *Trump* dataset, *La La Land* dataset, *The Voice* dataset and *Samsung* dataset by querying Twitter with respective strings.

In the context of the describing the dataset characteristics, we define two notions:

- *Density of NEs*—Density of Named Entities. It expresses the average number of Named Entities linked to a tweet
- *Coverage of NEs*—Coverage of Named Entities. It represents the percentage of the tweets in the dataset that contains at least one Named Entity.

The average parameter values for each of the datasets are presented in Table 1.

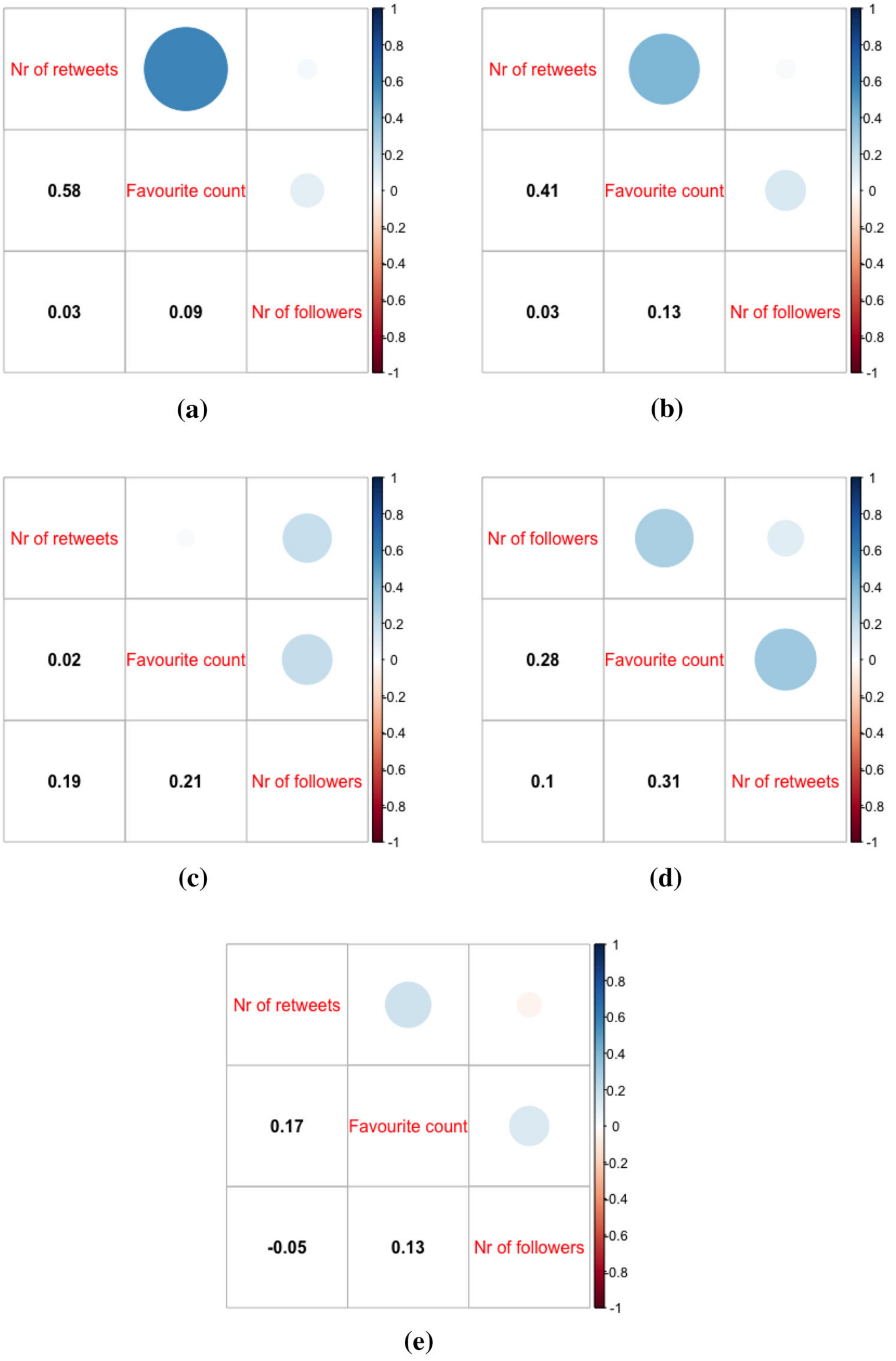
The datasets regarding public figures have a high density of Named Entities, as well as a high coverage (Table 2). Moreover, in terms of general characteristics, tweets regarding public figures come from popular users, and they are considerably retweeted and liked. Tweets regarding *La La Land* are less influential in terms of parameters of

[2] https://neo4j.com.

[3] http://sentistrength.wlv.ac.uk.

[4] https://nlp.stanford.edu.

interest. For instance, the average favorites of *Trump* and *Obama* are eight times more than the average favorites of *La La Land* and *Samsung*. The same argument goes for the retweets; *La La Land* and *Samsung* have almost half the retweets of *Trump* and almost one third of the retweets of *Obama*. Moreover, *La La Land* has a moderate density and low coverage of Named Entities. *Samsung*, on the contrary, has the highest number of followers compared to other datasets. This could be due to the fact that the community who follows technology is highly likely to be active on Twitter and excited about the brand and wants to follow the news. But the number of retweets is low compared to the followers.



(a) (b)

(c) (d)

(e)

Fig. 2 Correlation of parameters, **a** *Trump*, **b** *Obama*, **c** *La La Land*, **d** *The Voice*, **e** *Samsung*

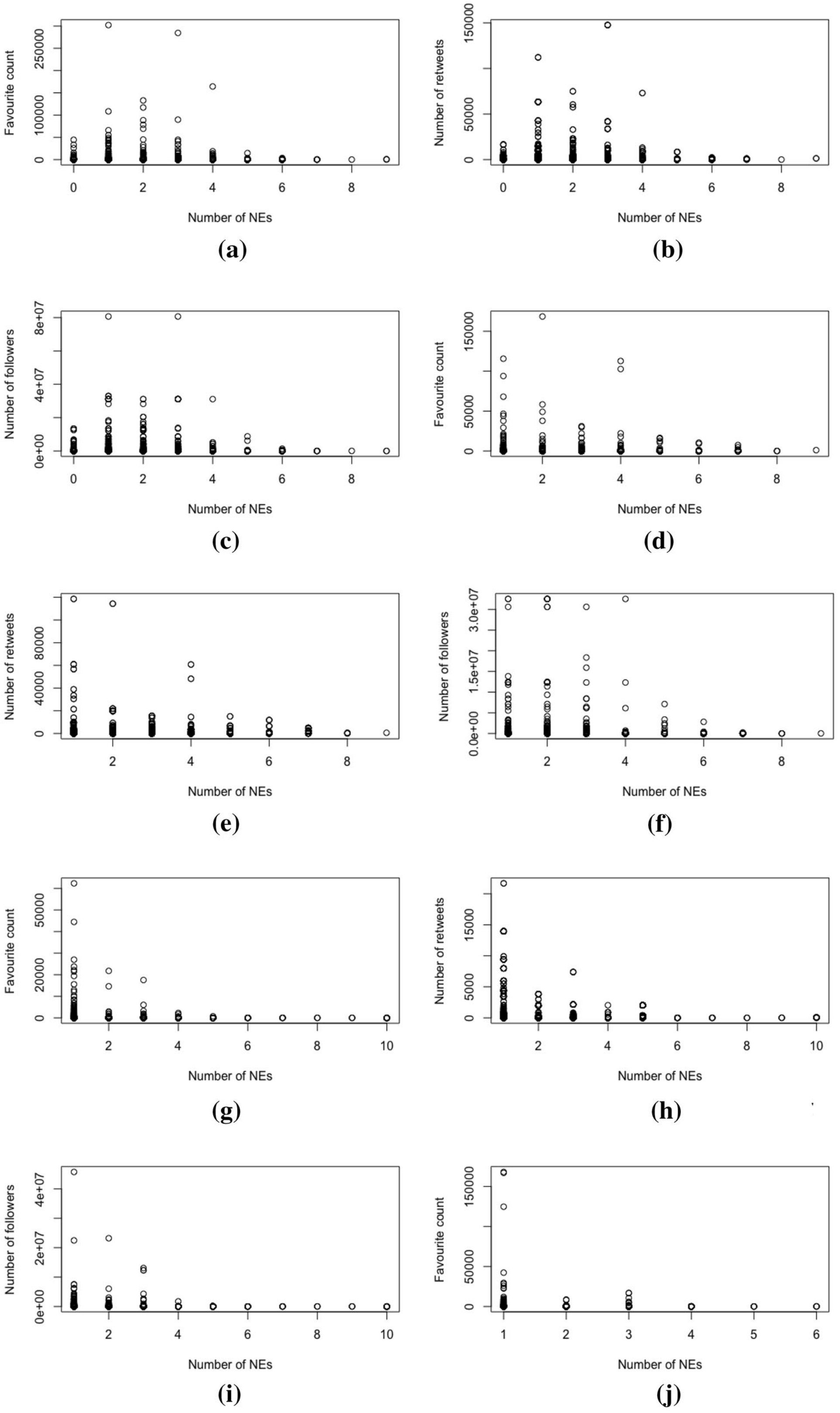


Fig. 3 Relation of parameters of interest and NEs in our datasets, **a** Favorite count versus NEs in *Obama*, **b** Retweets versus NEs in *Obama*, **c** Followers versus NEs in *Obama*, **d** Favorite count versus NEs in *Trump*, **e** Retweets versus NEs in *Trump*, **f** Followers versus NEs in *Trump*, **g** Favorite count versus NEs in *La La Land*, **h** Retweets versus NEs in *La La Land*, **i** Followers versus NEs in *La La Land*, **j** Favorite count versus NEs in *The Voice*, **k** Retweets versus NEs in *The Voice*, **l** Followers versus NEs in *La La Land*, **m** Favorite count versus NEs in *Samsung*, **n** Retweets versus NEs in *Samsung*, **o** Followers versus NEs in *Samsung*

When it comes to *The Voice*, the parameters of interest are satisfying; for instance, the average number of retweets is considerably higher than all other four datasets. However, it should be noted that sometimes retweets are a consequence of a marketing or advertising, not a real interest parameter; as if we compare it to the favorite count, it is a lower than *Obama* or *Trump* but still higher than *La La Land*. When it comes to the coverage and the density of Named Entities, *The Voice* is inferior to all other four datasets. For instance, compared to the public figures, it has half the density and half the coverage. *Samsung*, on the contrary, has a high density of NEs but with lower coverage. It is interesting for our evaluation to take into consideration datasets with different characteristics.

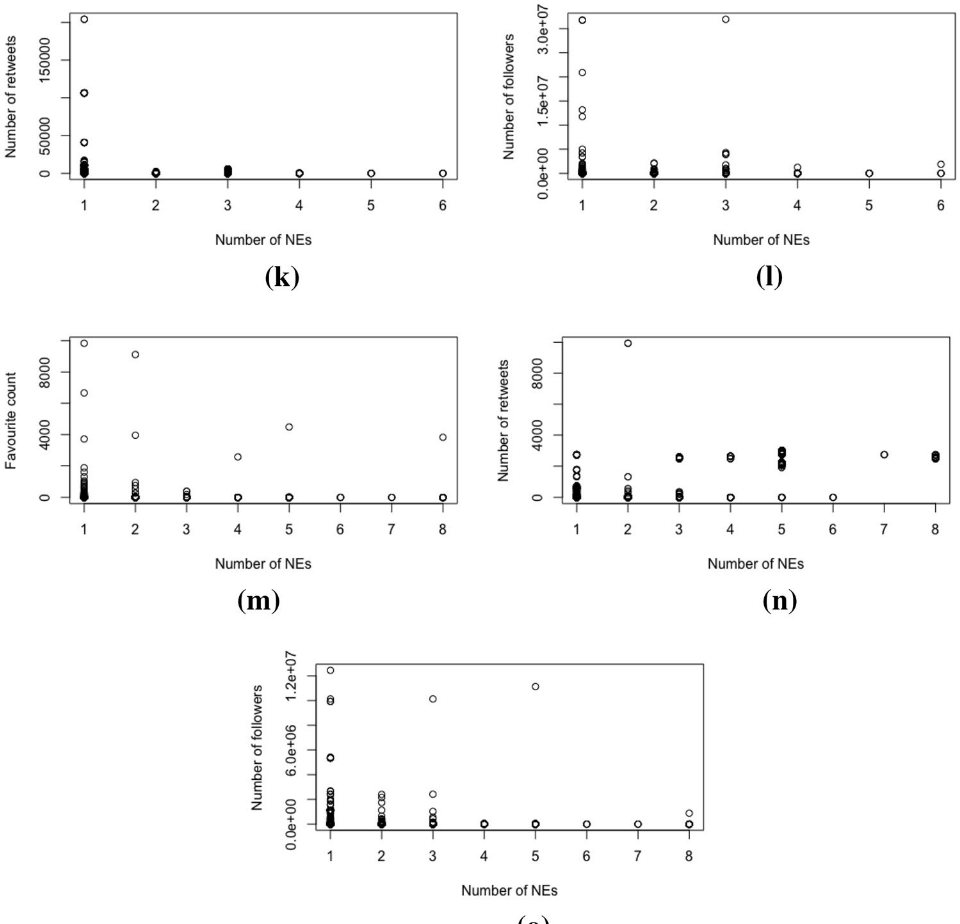
4.1.2 Correlation of Parameters of Interest

Since our proposed sampling algorithm is based on the number of retweets, the number of followers and the favorite count, we study the correlation of these parameters in our datasets. Intuitively, we would expect them to be correlated positively, under the assumption that people who have more followers are influential, so they would get more retweets and likes. In addition to this, tweets that have good content are retweeted and the owner of the tweet would get more followers.

The relation between the parameters of interest is shown in Fig. 2. As expected, the correlations are positive, expect a small negative correlation of − 0.05 between the number of followers and number of retweets in *Samsung*. The highest correlation of 0.58 is between number of retweets and the favorite count for *Trump* (Fig. 2a), followed by 0.41 for *Obama* (Fig. 2b). Nevertheless, we notice low correlations between the number of followers and the favorite count or number of retweets of the datasets of people. This is an interesting observation that shows that the number of followers of the owner is not correlated with the content of the tweets he posts.

La La Land (Fig. 2c), *The Voice* (Fig. 2d) and *Samsung* (Fig. 2e) are characterized generally by low correlations. In contrast to the datasets of people, the number of followers seems to be more correlated with the content of the tweets. Generally, the low values of correlation indicate that the parameters are not dependent on each other. Consequently, our sampling algorithm is not sensitive to the internal correlation between the parameters, resulting in a better sample.

Fig. 3 continued



(k)

(l)

(m)

(n)

(o)

4.1.3 Relation of Parameters of Interest with NEs

Previously, we studied the correlation of parameters of interest. In this section, we will focus on the relation between the parameters of interest and the number of Named Entities in tweets. We show the relation of number of NEs versus the favorite count (Fig. 3a), the number of retweets (Fig. 3b) and the number of followers (Fig. 3c).

The general behavior of this relation is that it resembles a normal distribution highly skewed on the right side. The tweets with 1–3 NEs in *Obama*have the most retweets and most likes, and their owners have more followers. Apparently having a high number of NEs does not imply high values of parameters of interest and vice versa. Nevertheless, we can explain this behavior from the fact that it is highly unlikely for a tweet to contain more than 6 entities; therefore, this relation is rarer to observe. Similarly, *Trump* dataset shows that tweets with more than 6 entities do not get high values of parameters of interest, simply because it is a rare event. However, in the case of *Trump*, there is a considerable amount of reaction even for tweets with only one entity, which would be *Trump*. This shows that event involving only *Trump* gets retweets and likes (Fig. 3d, e). In the case of the number of followers, the behavior is equally distributed between tweets of 1–4 entities (Fig. 3f).

La La Land (Fig. 3g, h) behaves similarly to *Trump*, most of the reaction as related to tweets with 1 entity. In the case of the followers (Fig. 3i), in contrast to *Trump*, *La La Land* continues to be concentrated in tweets with one NE. It is important to mention that *La La Land* is characterized by lower coverage and density of NEs compared to the people datasets, but also by low values of parameters of interest.

The Voice is the dataset that has the lowest numbers in terms of coverage and density of NEs. These statistics are obvious in the corresponding figures (Fig. 3j–l). The distribution is almost equal for tweets with more than one entity. *Samsung* brings new insights, where the tweets with more than 4 entities are able to get a reaction comparable to tweets with less than 4 entities (Fig. 3m–o). Apparently, events, where *Samsung* is involved, contain more NEs compared to other entities and they are interesting enough as to attract the audience.

4.2 The Richness of Weighted Sample

In our approach, we propose using weighted sampling for reputation discovery. Our hypotheses states that the weighted sampling provides richer information than the random sampling. Therefore, we extract a random sample and a weighted sample, following Algorithm 2 from all datasets. We compare the richness of the information in terms of these indicators:

- Number of Hashtags
- Number of URLs
- Number of Named Entities

These indicators are calculated for each of the samples. We iterated the procedure for 10 random samples and 10

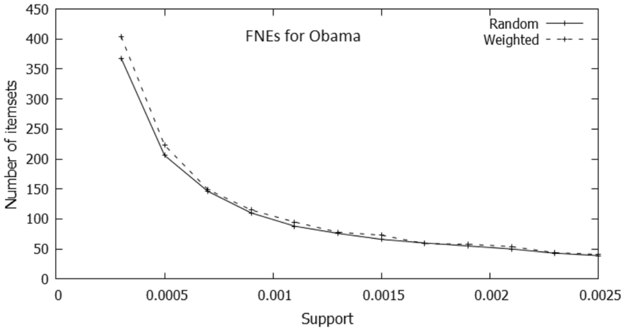


Fig. 4 Average number of itemsets for *Obama* dataset

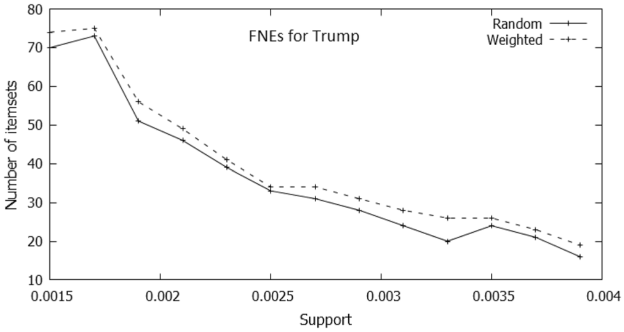


Fig. 5 Average number of itemsets for *Trump* dataset

Table 3 Average indicators of the samples

	Random			Weighted		
	Hashtags	Entities	URLs	Hashtags	Entities	URLs
Obama	14048.6	1828.5	5007.9	14256.1	1839.8	5230.2
Trump	8450.38	1609	2981.75	8655.25	1666.12	3094.5
La La Land	7986.9	1198.9	3102.9	9799.2	1081.6	3230.1
The Voice	1047.2	2856.7	1353	668.7	3368.2	1658
Samsung	1930	419.4	521.4	2069	458	656.2

weighted samples, for each of the datasets. The average of the indicators is presented in Table 3.

According to Table 3, the weighted sample is significantly richer in terms of the aforementioned indicators for *Trump*, *Obama*, and *Samsung* datasets. Tweets that contain more information are more useful to be analyzed. Nevertheless, in terms of entities in *La La Land* and in terms of hashtags in *The Voice*, weighted sample has not been able to perform better. Since one of our parameters of interest is retweet count, sometimes for the movies and TV shows promotional tweets are retrieved, which might not be richer in information.

4.3 Frequent Named Entity Mining in Weighted Sample

Frequent Named Entities are discovered through itemset mining techniques [2]. The tweets are considered as transactions and the Named Entities as itemsets. We used R to perform these experiments, arules package and eclat algorithm.

For all three datasets, we used 50 random samples and 50 weighted samples to extract FNEs and to get an average of the number of FNEs for each support value. For *Obama* dataset (Fig. 4) and *Trump* dataset (Fig. 5) the weighted sample performs better for each of the support values, providing more FNEs than the random sample.

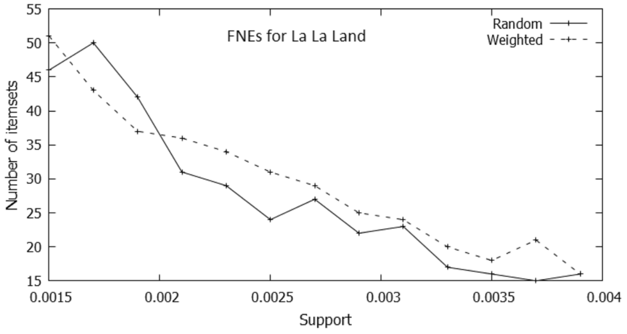


Fig. 6 Average number of itemsets for *La La Land* dataset

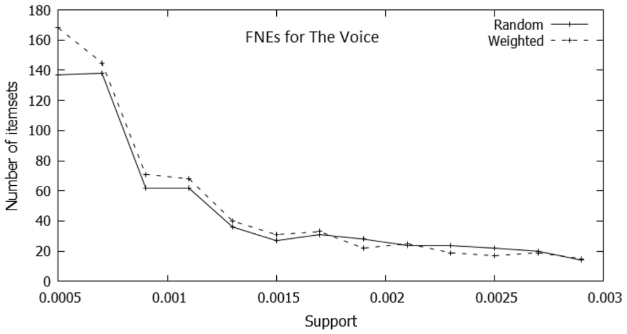


Fig. 7 Average number of itemsets for *The Voice* dataset

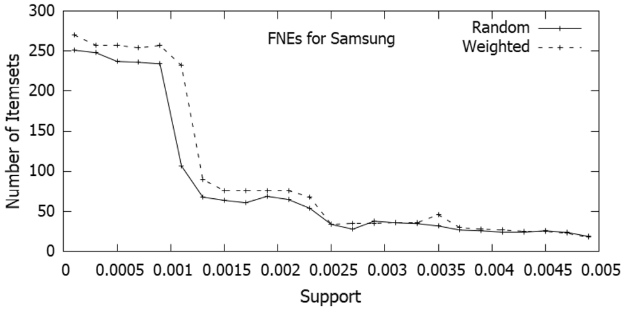


Fig. 8 Average number of itemsets for *Samsung* dataset

Obama dataset in Fig. 4 shows a similar behavior as *Trump* dataset. For the same support, the weighted sample performs better, sometimes significantly better; in the low support values, the weighted sample provides 20–40 more FNEs than the random sample.

The weighted sample in *La La Land* (Fig. 6), in general, extracts more FNEs than the random sample. However, there are fluctuations in this behavior. The reason behind this event might be related to the fact that the itemsets in the random sample are dependent only to the support, while for the weighted sample, the parameters of interest play an important role as well. Since *La La Land* was inferior in terms of parameters of interest and in density and coverage of Named Entities, compared to the public figures' datasets, the weighted sample is not able to make a sustainable difference.

In the case of *The Voice* dataset (Fig. 7), the weighted sample is superior to the random sample. In contrast to *La La Land*, even though *The Voice* has lower density and coverage, the weighted sample maintains a more stable behavior, since it is advantageous in terms of parameters of interest. Thus, we can highlight here the ability of the weighted sample to produce richer information, provided that the parameters of interest are satisfying, even though the dataset itself might be poor in terms of Named Entities.

Samsung (Fig. 8) proves to be robust for different values of support in terms of FNEs discovered through the weighted sample. The weighted sample is consistently better than the random sample, especially in low support values where it has an advantage of 20–30 FNEs more than the random sample.

4.4 Comparing the Ranking of the FNEs

Since we are exploring FNEs through samples, we want to guarantee that the FNEs discovered are similar to the FNEs of the population. We ran eclat algorithm on the whole datasets to discover the FNEs. As we need to compare lists of itemsets, Kendall rank correlation is helpful in identifying how similar the lists are. It takes into consideration

Table 4 Similarity coefficients

Coefficient	*Obama*	*Trump*	*La La Land*	*The Voice*	*Samsung*
Spearman	0.7979	0.764	0.653	0.790	0.771
Kendall	0.8984	0.597	0.789	0.618	0.678

the concordant pairs(C) and discordant pairs(D) to generate a value between -1 and 1. Concordant implies that if rank (x) > rank (y) in the list A, then rank (x) > rank (y) in list B as well. Otherwise, they are discordant pairs. The higher the Kendall value, the more similar the lists are. The Kendall coefficient is defined as:

$$r = \frac{C - D}{n(n-1)/2} \tag{3}$$

where n is the number of pairs that are compared.

Spearman's rank order is used as well to compute the similarities between different ranks. Even though Kendall coefficient is more direct, as it considers the agreeing and disagreeing ranks, the Spearman coefficient tends to find the relationship between ordinal variables as in the following formula:

$$\rho = 1 - \frac{6\sum_{d_i^2}}{n(n^2 - 1)} \tag{4}$$

where d is the distance between ranks and n is the number of pairs that are compared.

We matched and ranked the FNEs in the population and in the sample. Then, we calculated the Kendall coefficient and the Spearman rank order for both rankings. We repeated the experiment for 10 samples from *Obama*, *Trump*, *La La Land*, *The Voice*, and *Samsung* dataset. The average values of 10 samples of each dataset regarding Kendall and Spearman coefficient are presented in Table 4, showing a considerable similarity between the sample and the whole population in terms of ranking of itemsets.

4.5 Reputation Through Frequent Named Entities

Exploring the reputation of an entity through the Frequent Named Entities that the dataset contains is interesting to discover. In this experiment, we used Algorithm 3 to track back the tweets of the sample that represent the explored FNEs. For each FNE, we found the sentiment and calculated its reputation. In order to respect the frequency of the FNE in the sample, we weighted the reputation by the support of the FNE. In the end, we calculated an overall reputation as in:

$$\sum_{k=1}^{n} r_k * s_k \tag{5}$$

where r_k is the sentiment of the reputation of the itemset i_k and s_k is the support of the i_k in S.

Table 5 Reputation extraction through FNEs

Dataset	Whole population		Weighted sample	
	Positive	Negative	Positive	Negative
Obama	40.79	$-$ 59.20	40.91	$-$ 59.08
Trump	32.04	$-$ 67.96	38.22	$-$ 61.77
La La Land	74.42	$-$ 25.57	90.87	$-$ 09.12
The Voice	56.06	$-$ 43.93	55.28	$-$ 44.71
Samsung	79.31	20.69	83.6	17.4

We implemented this idea for *Obama* dataset and *Trump* dataset and repeated the experiment 10 times for each case (Table 5). Both datasets related to public figures showed a precise alignment of the reputation explored through FNEs after weighted sampling with the reputation of the whole population. The average accuracy of the interpretation through FNEs is 90%. Nevertheless, in the case of the movie *La La Land*, we can distinguish a difference between both results. This misalignment comes from the fact that movies are not as dynamic as public figures; therefore, the reputation of a movie is enriched by FNEs, but not defined by them. We can also explain this result with the lower coverage and density of NE in *La La Land* compared to *Obama* and *Trump* (Table 2). Moreover, since the parameters of interest are the lowest compared to the other datasets, the weighted sample cannot exploit a lot of behavior from the dataset.

The Voice is an interesting case, as since it is a TV show, it is expected to behave as *La La Land*. Even though it is inferior in density and coverage of Named Entities, it manages to round up the reputation of the entity from its Frequent Named Entities almost precisely. The advantage of *The Voice* lies in the fact that the weighted sample is more powerful, due to the fact that the parameters of interest are considerably better than *La La Land*. *Samsung* reveals new insights regarding Named Entities. *Samsung* resembles *La La Land* in terms of parameters of interest; it has low retweets and favorite count, but a high number of followers. Moreover, in terms of density of NEs and coverage of NEs, *Samsung* is rich, comparable to the people datasets of *Trump* and *Obama*. As a result, *Samsung* manages to have a good interpretation of reputation regarding the whole dataset.

In the case of *La La Land*, through Frequent Named Entities it is possible to discover dominating opinions that bias the dataset. For instance, in all of our 10 samples, the first FNE was related Emma Stone and JAEBUM and had a reputation of (+ 100, $-$ 0). Emma Stone has held a picture of JAEBUM as a gesture of appreciation, and this event has gone viral on Twitter. As a result, it dominated the dataset in a positive way and affects the whole reputation. Emma Stone is the main actress in *La La Land*, that is the reason why this event is part of *La La Land* dataset. However, this event is not related to the movie. With the help of the itemset mining, viral events that are not relevant can be distinguished and discarded from the aggregation.

To conclude, rich tweets in Named Entities are able to interpret better the reputation of target entities. When the tweets have a considerably low coverage and density of NEs, then the weighted sample provides a better sample to discover the reputation. This is the case of *La La Land* in our experiments, which is able to overcome the problem of low density and coverage of NEs through the weighted sample. The parameters of interest contribute in selecting rich tweets and improving the results.

It is important to note that our contribution does not focus on finding a reputation, but in enriching the interpretation of reputation by the means of Frequent Named Entities. We have found some interesting observation such as the reputation obtained for *Trump* was $(+30, -60)$, whereas *Donald Trump* had $(+50, 50)$ and the itemset $\{$Trump,Obama$\}$ had $(+52, -48)$. Trump who by himself has a negative score has a more positive score together with *Obama*; this could be because people may be comparing Trump to the former president who has a more positive attitude from people. This self-explanatory approach gives the user the possibility to interpret the information, and since it breaks down the reputation of an entity into the reputation of the groups of entities it belongs to, the user has the freedom to use the pieces of reputation in a meaningful way.

5 Conclusions

We addressed the problem of reputation discovery and aggregation of sentiments by exploring the underlying entities that coexist in the data. We stressed the importance of information interpretation in explaining the reputation of an entity. We introduced a weighted sampling technique to improve the richness of the dataset.

We evaluated our approach comparing random and weighted sample in terms of statistics of indicators, and we tested the power of Frequent Named Entity Mining on reputation discovery. Our proposed weighted sampling technique proved to have an advantage over the random sample. We showed that our approach proves to be generally robust to the type of the entity of interest. In the case of entities that have low values of coverage and density of FNEs like *La La Land* and *The Voice*, the weighted sampling helps in improving the reputation discovery. We pointed out that FNEs contribute in around 90% of the reputation of the entity, especially in cases of public figures, who are highly dynamic in their collaborations with other entities.

This idea yields promising in Twitter, due to the entity interconnections, so we suggest implementing it on other social networks. Social networks are affected by the linkage between nodes, and this property should be exploited in aggregating information.

In this paper, we used a ranking algorithm based on properties of interest to weight the tweets. Further studies on weighting techniques or choosing and transforming the properties of interest could improve the quality of the sample.

We encourage the research on the reputation extraction through Frequent Named Entities, as it is self-explanatory and transparent. Further work could be applied on merging and combining the reputation of the itemsets, in order to compute to the reputation of the entity. Aggregation techniques for reputation discovery could enrich this work and contribute to reputation integration.

References

1. Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R (2011) Sentiment analysis of twitter data. In: Proceedings of the workshop on languages in social media, Association for Computational Linguistics, pp 30–38
2. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. Acm SIGMOD Rec 22:207–216
3. Bennacer N, Bugiotti F, Hewasinghage M, Isaj S, Quercini G (2017) Interpreting reputation through frequent named entities in twitter. In: International conference on web information systems engineering. Springer, pp 49–56
4. Bizhanova A, Uchida O (2014) Product reputation trend extraction from twitter. Social Networking, Scientific Research Publishing, 2014
5. Bollen J, Mao H, Pepe A (2011) Modeling public mood and emotion: twitter sentiment and socio-economic phenomena. ICWSM 11:450–453
6. Cha M, Haddadi H, Benevenuto F, Gummadi PK (2010) Measuring user influence in twitter: the million follower fallacy. ICWSM 10(10–17):30
7. Ding X, Liu B, Zhang L (2009) Entity discovery and assignment for opinion mining applications. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1125–1134
8. Esuli A, Sebastiani F (2006) Sentiwordnet: a publicly available lexical resource for opinion mining. Proc LREC 6:417–422
9. Gabielkov M, Rao A, Legout A (2014) Sampling online social networks: an experimental study of twitter. ACM SIGCOMM Comput Commun Rev 44:127–128

10. Ghosh S, Zafar MB, Bhattacharya P, Sharma N, Ganguly N, Gummadi K (2013) On sampling the wisdom of crowds: random versus expert sampling of the twitter stream. In: Proceedings of the 22nd ACM international conference on information and knowledge management, pp 1739–1744
11. Gjoka M, Kurant M, Butts CT, Markopoulou A (2010) Walking in facebook: a case study of unbiased sampling of OSNs. In: Infocom, 2010 Proceedings IEEE, pp 1–9
12. Gouriten G, Maniu S, Senellart P (2014) Scalable, generic, and adaptive systems for focused crawling. In: Proceedings of the 25th ACM conference on hypertext and social media, pp 35–45
13. Hangya V, Berend G, Farkas R (2013) Szte-nlp: sentiment detection on twitter messages. In: Second joint conference on lexical and computational semantics, vol 2, pp 549–553
14. Heerschop B, Hogenboom A, Frasincar F (2011) Sentiment lexicon creation from lexical resources. In: International conference on business information systems. Springer, pp 185–196
15. Hogenboom A, Bal D, Frasincar F, Bal M, de Jong F, Kaymak U (2013) Exploiting emoticons in sentiment analysis. In: Proceedings of the 28th annual ACM symposium on applied computing, pp 703–710
16. Leskovec J, Faloutsos C (2006) Sampling from large graphs. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 631–636
17. Meng X, Wei F, Liu X, Zhou M, Li S, Wang H (2012) Entity-centric topic-oriented opinion summarization in twitter. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, pp 379–387
18. Nazi A, Zhou Z, Thirumuruganathan S, Zhang N, Das G (2015) Walk, not wait: faster sampling over online social networks. Proc VLDB Endow 8(6):678–689
19. Saif H, He Y, Alani H (2012) Semantic sentiment analysis of twitter. In: International semantic web conference. ACM, pp 508–524
20. Thelwall M, Buckley K, Paltoglou G (2011) Sentiment in twitter events. J Am Soc Inf Sci Technol 62(2):406–418
21. Thelwall M, Buckley K, Paltoglou G (2012) Sentiment strength detection for the social web. J Am Soc Inf Sci Technol 63(1):163–173
22. Van Canneyt S, Claeys N, Dhoedt B (2015) Topic-dependent sentiment classification on twitter. In: European conference on information retrieval. Springer, pp 441–446
23. Wang T, Chen Y, Zhang Z, Xu T, Jin L, Hui P, Deng B, Li X (2011) Understanding graph sampling algorithms for social network analysis. In: 2011 31st international conference on distributed computing systems workshops, pp 123–128
24. Wang X, Wei F, Liu X, Zhou M, Zhang M (2011) Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM international conference on Information and knowledge management, pp 1031–1040
25. Xiang B, Zhou L, Reuters T (2014) Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. ACL 2:434–439
26. Zheng B, Wang H, Zheng K, Su H, Liu K, Shang S (2018) Sharkdb: an in-memory column-oriented storage for trajectory analysis. World Wide Web 21(2):455–485
27. Zheng K, Su H, Zheng B, Shang S, Xu J, Liu J, Zhou X (2015) Interactive top-k spatial keyword queries. In: 31st IEEE international conference on data engineering, ICDE 2015, Seoul, South Korea, 13–17 April 2015, pp 423–434
28. Zheng K, Zheng B, Xu J, Liu G, Liu A, Li Z (2017) Popularity-aware spatial keyword search on activity trajectories. World Wide Web 20(4):749–773
29. Zhou Z, Zhang X, Sanderson M (2014) Sentiment analysis on twitter through topic-based lexicon expansion. In: Australasian database conference. Springer, pp 98–109