Date of publication January 2020.

Learning Political Polarization on Social Media using Neural Networks

LORIS BELCASTRO[1], RICCARDO CANTINI[1], FABRIZIO MAROZZO[1,2], DOMENICO TALIA[1,2] and PAOLO TRUNFIO[1,2]

[1]DIMES. University of Calabria, Italy
[2]DtoK Lab Srl, Rende, Italy

Corresponding author: F. Marozzo (e-mail: fmarozzo@dimes.unical.it).

ABSTRACT Social media analysis is a fast growing research area aimed at extracting useful information from social media platforms. This paper presents a methodology, called *IOM-NN* (*Iterative Opinion Mining using Neural Networks*), for discovering the polarization of social media users during election campaigns characterized by the competition of political factions. The methodology uses an automatic incremental procedure based on feed-forward neural networks for analyzing the posts published by social media users. Starting from a limited set of classification rules, created from a small subset of hashtags that are notoriously in favor of specific factions, the methodology iteratively generates new classification rules. Such rules are then used to determine the polarization of people towards a faction. The methodology has been assessed on two case studies that analyze the polarization of a large number of Twitter users during the 2018 Italian general election and 2016 US presidential election. The achieved results are very close to the real ones and more accurate than the average of the opinion polls, revealing the high accuracy and effectiveness of the proposed approach. Moreover, our approach has been compared to the most relevant techniques used in the literature (sentiment analysis with NLP, adaptive sentiment analysis, emoji- and hashtag- based polarization) by achieving the best accuracy in estimating the polarization of social media users.

INDEX TERMS Social media analysis, opinion mining, user polarization, neural networks, sentiment analysis, political events.

I. INTRODUCTION

Every day millions of people use social media and produce huge amount of digital data that can be effectively exploited to extract valuable information concerning human dynamics and behaviors. Such data, commonly referred as Big Data, contains valuable information about user activities, interests, and behaviors, which makes it intrinsically suited to a very large set of applications [1]. Big Social Data analysis is a sub-field of Big Data analysis aimed at studying the interactions of users on social media for extracting useful information, such as moods or opinions on topics or events of interest [2].

This paper presents a new methodology, namely *IOM-NN* (*Iterative Opinion Mining using Neural Networks*), for estimating the polarization of public opinion on political events characterized by the competition of factions or parties. It can considered as an alternative technique to traditional opinion polls, since it is able to capture the opinion of a larger number of people more quickly and at a lower cost. In particular, IOM-NN uses an automatic incremental procedure based on feed-forward neural networks for analyzing the posts published by social media users. Starting from a limited set of classification rules, created from a small subset of hashtags that are notoriously in favor of specific factions, our methodology iteratively generates new classification rules. A classification rule allows to determine if a post is in favor of a faction based on the words/hashtags it contains. Then, such rules are used to determine the polarization of social media users - who wrote posts about the political event - towards a faction.

The proposed methodology has been applied to two case studies for analyzing the polarization of a large number of Twitter users during the 2018 Italian general election and the 2016 US presidential election. The results obtained by IOM-NN have been compared to opinion polls collected before voting and the most relevant techniques used in the literature (i.e., *sentiment analysis with NLP* [3], *adaptive sentiment analysis* [4], *emoji-based polarization* [5], *hashtag-based polarization* [6]). The results achieved by IOM-NN are very close to the real ones and more accurate than opinion polls and other relevant techniques, revealing the high accuracy and effectiveness of the proposed approach. For example, considering the 2018 Italian general election and the four parties that received the highest number of votes (M5S, PD, LEGA, FI), IOM-NN achieved a mean average error (MAE) of 1.13 percentage points and a log accuracy ratio (LogAcc) very close to 1. Opinion polls achieved a MAE of 3.74 percentage points and a LogAcc of 0.81. Compared with the other existing techniques, IOM-NN turned out to be the most accurate in forecasting the winning candidate. For example, considering the 2016 US presidential election, IOM-NN has been able to correctly identify the winning candidate in 8 out of 10 states, while the other techniques identified the winner in up to 6 out of 10 states.

Compared to existing techniques, our methodology includes the following innovative aspects: $i)$ it allows to classify a much high number of users, which in our case studies results to be ten times larger than that involved in traditional opinion polls; $ii)$ it measures the political consensus of a faction considering the number of users supporting it, without being influenced by users who published a large volume of posts; $iii)$ it is able to process posts written in any language without the need to use dictionaries or translation systems; $iv)$ at the best of our knowledge, it is the first methodology in this research field that exploits an iterative learning approach to increase the amount of classified data; $v)$ collected data has been statistically validated for assessing the representativeness of users involved in the analysis.

This manuscript significantly extends a previous work [7] in the following main aspects: $i)$ it provides an in-depth definition of the steps of the methodology and a formal description through pseudo-code of the algorithms used for classifying the posts and predicting the users' polarization (see Sections III-B and III-C); $ii)$ it presents and discusses a new case study on which the methodology has been applied and tested (see Section IV); $iii)$ it includes the evaluation of the statistical significance of the collected data for the two case studies; and $iv)$ it reports more extensive and detailed tests and comparisons with relevant techniques used in the literature (see Section IV).

The remainder of the paper is organized as follows. Section II discusses related work. Section III describes the proposed methodology. Section IV presents the case studies and Section V concludes the paper.

II. RELATED WORK

In recent years, social media analysis is arousing great interest in various scientific fields, such as sociology, political science, linguistics, and computer science [8]. In this section, we focus on the main techniques and algorithms proposed for measuring public opinion and predicting election results through social media. As suggested in [9] and [10], the existing techniques can be divided into three main categories: *volume-based*, *sentiment-based* and *network-based*. For each category, the main proposed solutions and their differences with respect to our technique are discussed.

Volume-based techniques counts the number of mentions (e.g., posts, likes, retweets) related to a candidate/party for predicting the election results. In many cases, such techniques analyze social media data for predicting the outcome of an election. For example, Gaurav et al. [11] proposed a technique based on moving average aggregate probability, which infers the results of an election by counting how many times a candidate's name is mentioned in tweets. Tumasjan et al. [12] used micro-blogging data for understanding how people express their political orientation, showing a high closeness between the volume of tweets mentioning a party and the election results. Burnap et al. [13] analyzed the volume of mentions for calculating an overall score for each party.

Unlike volume-based techniques, which consider the number of posts in favor of a faction, our technique takes into account the number of users supporting a faction. In this way, IOM-NN obtains more accurate results since it is not influenced by users who published a large number of posts.

Sentiment- or opinion-based techniques exploit natural language processing (NLP) or text mining algorithms for understanding the opinion of users towards political candidates or parties. Such techniques result to be more advanced than the volume-based ones as they analyze the textual content of posts to calculate a score.

The techniques based on *natural language processing* consider the hierarchical structure of a text to understand its meaning and sentiment. For example, Oikonomou et al. [3] exploited Textblob[1], a Python library for natural language processing, to predict the outcome of USA presidential elections in three states of interest (i.e., Florida, Ohio and North Carolina). Wong et al. [14] combined convex optimization techniques with SentiStrength[2], a lexicon-based sentiment analysis tool, for modeling the political behaviors of users by analyzing tweets and retweets. Alashri et al. [15] analyzed Facebook posts about the 2016 US presidential election with CoreNLP[3] [16], one of the most popular tool for natural language processing, to calculate a score for each political candidate.

Techniques based on *text mining* discover the sentiment of a text by considering only the words it contains without analyzing its structure. For example, El Alaoui et al. [4] proposed an adaptive sentiment analysis approach that generates dictionaries from tweets classified as positive/negative for the different factions. Such dictionaries are then used to calculate a score for each faction. Similarly, Marozzo and Bessi [6] calculated a polarization score for each faction by considering only the hashtags in tweets labeled as positive. Chin et al. [5] exploited the emojis contained in a post to determine its sentiment (e.g., positive or negative). Other studies exploited machine learning techniques for discovering the political orientation of users, such as classification

[1] https://textblob.readthedocs.io/
[2] http://sentistrength.wlv.ac.uk/
[3] https://stanfordnlp.github.io/CoreNLP/

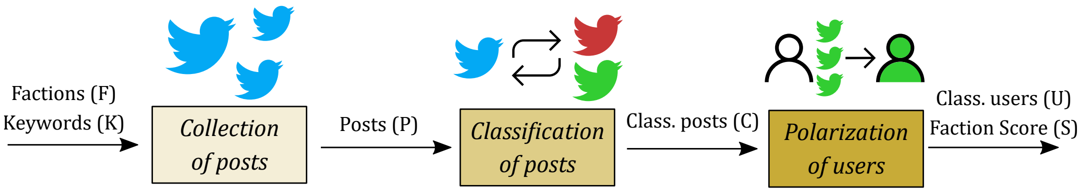

FIGURE 1: Execution flow of IOM-NN.

models based on the Naïve Bayes algorithm [17] or logistic regression [18].

IOM-NN is a text mining technique that uses bag-of-words and neural networks to classify posts, and consequently users who wrote such posts about a political event. Compared to existing text mining techniques, its iterative approach allows to greatly increase the number of classified posts, while the use of neural networks permit to automatically discover classification rules with a high level of accuracy. With regard to NLP techniques, it is worth noting that their usability and accuracy depend on the specific tool used and the supported languages. In fact, the most popular tools for natural language processing (e.g., CoreNLP) support sentiment analysis only for English texts. Instead IOM-NN classifies users by analyzing the words/hashtags contained in the posts that can be written in any language, without using dictionaries or translation systems.

Network-based techniques analyze the network structure of social media users, which support or discuss about certain candidates or parties, for understanding the dynamics of public opinion. Such analysis can provide useful insights for estimating the standing of political events or identifying the opinion leaders on a social media platform [19]. In fact, some studies have demonstrated a relation between the centrality of political candidates on social networks and their electoral consensus [20] [21]. However, it should be noticed that such techniques require the use of specific data that represents the social network structure, which is often visualized through graphs or sociograms. In our study, we collected and analyzed tweets containing specific keywords or hashtags on the political event under analysis, without capturing the structure of the related social network. For this reason, in this paper we cannot make a comparison with network-based techniques.

Some studies highlighted the issues related to the use of social media data for predicting the outcome of political events, which are language barrier, misclassification, data imbalance and reliability [10]. During the design of IOM-NN we faced such issues by proposing the following solutions:

- *Language barrier*. Our technique classifies users by analyzing the words/hashtags contained in their posts, regardless of the language used to write them.
- *Misclassification*. Starting from a limited set of classification rules, created from a small set of hashtags that are notoriously in favor of specific factions, the methodology iteratively generates new classification rules.
- *Data imbalance*. To avoid the learning process being biased towards majority classes, a random under-sampling approach is used to balance the dataset at each training phase (see Algorithm 1).
- *Data reliability*. The statistical significance of the collected data has been evaluated for assessing the representativeness of users, i.e., understanding whether they can be considered voters in the political event under analysis.

III. PROPOSED METHODOLOGY (IOM-NN)

As mentioned in Section I, IOM-NN is a methodology for estimating the polarization of public opinion during a political event, which is characterized by the rivalry of different factions. As shown in Figure 1, the proposed methodology consists of three main steps:

1) *Collection of posts*: posts are collected by using a set of keywords related to the selected political event (see Section III-A).
2) *Classification of posts*: the collected posts are then classified by using an incremental procedure implemented through neural networks (see Section III-B).
3) *Polarization of users*: the classified posts are analyzed for determining the polarization of users towards a faction (see Section III-C).

For each step, a formal description and practical examples are provided in the following sections. For the sake of clarity, Table 1 reports the meaning of the main symbols used to describe the different steps.

Symbol	Meaning
$\mathcal{E}$	Political event
$F = \{f_1, f_2, ..., f_n\}$	Factions
$K = K_{context} \cup K_F^{\oplus}$	Context keywords and positive faction keywords
$K_F^{\oplus} = K_{f1}^{\oplus} \cup ... \cup K_{fn}^{\oplus}$	Positive keywords grouped by factions
P	All the posts in input
C^i	Classified posts at the i-th iteration
N^i	Not classified posts at the i-th iteration
M^i	Classification model generated at the i-th iteration
C	Classified posts
U	Polarized users
S	Faction score

TABLE 1: Meaning of the most important symbols used in the proposed methodology.

A. COLLECTION OF POSTS

A political event $\mathcal{E}$ is characterized by the rivalry of different factions $F = \{f_1, f_2, ..., f_n\}$. Examples of political events and relative factions are: $i)$ municipal election, in which a faction supports a mayor candidate; $ii)$ parliament election, in which a faction supports a party; $iii)$ presidential election, in which a faction supports a presidential candidate. The posts are collected by using the keywords that people commonly use to refer the political event $\mathcal{E}$ on social media. Such keywords K can be divided in two groups:

- $K_{context}$, which contains generic keywords that can be associated to $\mathcal{E}$ without referring to any specific faction in F.
- $K_F^{\oplus} = K_{f1}^{\oplus} \cup ... \cup K_{fn}^{\oplus}$, where $K_{fi}^{\oplus}$ contains the keywords used for supporting $f_i \in F$ (positive faction keywords).

The keywords in K are given as input to public APIs provided by social media platforms, which permit to *collect* posts containing one or more keywords. Posts are not collected in real time, but downloaded at a given time after their publication (e.g., 24 hours). In this way, we are able to get some statistics related to the popularity of a post (e.g., number of shares, number of likes). Since data collection is usually a continuous process, new keywords can be discovered and integrated in K during the collection procedure. It is important to highlight that obtaining a representative collection of posts depends on two factors: $i)$ the quality and the number of keywords used; $ii)$ the amount of data that can be downloaded from social media. Regarding the latter factor, it is increasingly difficult to obtain complete data from social media platforms due to the restrictions introduced for protecting the privacy of users.

The collected posts are *pre-processed* before the analysis. In particular, they are modified and filtered as follows:

- The text of posts is normalized by transforming it to lowercase and replacing accented characters with regular ones (e.g., IOVOTOSI or iovotosí $\rightarrow$ iovotosi).
- Words are stemmed for allowing matches with declined forms (e.g., vote or votes or voted $\rightarrow$ vot).
- Stop words are removed from text by using preset lists.
- All the posts written in a language different from the one(s) spoken in the nation(s) hosting the considered political event are filtered out.

The output of this step is a collection of posts P. Figure 2 shows an example of how posts are collected using keywords about the 2016 US presidential election. Some of these keywords are generic (e.g., election2016), and others are used to support a specific candidate (e.g., #imwithher for Clinton and #votetrump for Trump).

Before the analysis, the *statistical significance* of the collected data has to be evaluated. We studied the age, gender and geographical distribution of social media users who generated such data. The aim is to assess the users' representativeness by understanding whether they can be considered

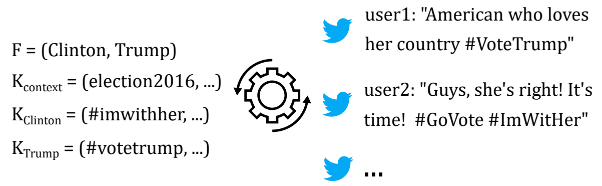

FIGURE 2: Example of how the *collection of posts* step works.

voters of the political event under analysis (more details in Sections IV-A1 and IV-B).

B. CLASSIFICATION OF POSTS

Algorithm 1 shows the pseudo-code used for classifying the posts. The input is composed of: the posts P generated in the previous step, the set of positive faction keywords $K_F^{\oplus}$, the maximum number of iterations max_{iters}, the minimum increment of the classified posts eps at each iteration, and a threshold th. The output is a collection of posts C that have been classified in favor of a faction.

The algorithm is divided in two parts. The fist part (lines 1-9) performs the preliminary iteration (iteration 0). At this iteration, IOM-NN exploits the set of positive faction keywords ($K_F^{\oplus}$) for classifying a part of the posts. Specifically, it classifies a post in favor of a faction if it contains only positive keywords for such faction. In general, at the end of this iteration, a small amount of posts are classified, since not all users use keywords in $K_F^{\oplus}$ for declaring their support to factions. The second part (lines 10-21) iteratively generates new classification rules for classifying other posts. At each iteration, such rules are inferred by exploiting the posts that have been classified at the previous iterations. In the following of this section, we discuss in detail the code of Algorithm 1.

The algorithm initializes an empty set C^0 for storing the classified posts and builds a classification model M^0 based on the positive faction keywords $K_F^{\oplus}$ (lines 1-2). The model defines a set of rules for calculating a binary vector v_b, where $v_b[i]$ is 1 if p contains at least one keyword from $K_{fi}^{\oplus}$, 0 otherwise. The algorithm iterates (lines 3-7) on each post p in P performing the following operations:

- classifies p using M^0, which produces a vector v_b (line 4);
- if p is in favor of a single faction f (lines 5-6), the classified post (i.e., a pair $\langle p, f\rangle$) is added to C^0 (line 7).

At the end of iteration 0, the set of classified posts C^0 is stored in C (line 8). The set of unclassified posts (N^0) is obtained as the difference between P and C^0 (lines 9). It is important to note that the number of keywords has to be balanced among the different factions for avoiding a classification process biased towards a faction.

The second part of the algorithm (lines 10-21) performs at most max_{iters} iterations. Specifically, at the i-th iteration, the following operations are performed:

- It initializes an empty set C^i for storing the classified posts at i-th iteration (line 11).
- It builds a classification model M^i by training a neural network using the classified posts at previous iterations $C^0 \cup \ldots \cup C^{i-1}$ (lines 12). The training set is balanced by using a random under-sampling approach to avoid a learning process biased towards majority classes.
- For each unclassified post at the previous iteration N^{i-1} (line 13), the algorithm classifies p using M^i, which produces a vector of probabilities v_p (line 14), where $v_p[i]$ is the probability that p supports f_i. If the maximum value of v_p is greater than the given threshold th, the post is assigned to the most likely faction f (lines 15-16) and added to C^i (line 17).
- The set of classified posts C^i is added to C (line 18), and the unclassified posts N^i are obtained as difference between N^{i-1} and C^i (line 19).
- If the ratio between the size of C^i and the size of N^{i-1} is lower than eps or greater that $1 - eps$, then it breaks the loop (lines 20-21).

Finally, the algorithm returns the dictionary C containing all the posts classified at the various iterations (line 22). Since the parameters of the neural network are randomly initialized, IOM-NN repeats the post classification phase with a new random seed for n_{seeds} times, in order to reduce the risk of getting stuck in local minima or saddle points.

Figure 3 shows how the post classification algorithm (Algorithm 1) works starting from a set of posts P. At the iteration 0, the classification model M^0 is created using the faction keywords K_F. This model is used to classify P, which generates two subsets for classified (C^0) and unclassified posts (N^0) respectively. At iteration 1, a new model M^1 is trained using C^0 and is used to classify the unclassified posts generated at the previous iteration (N^0). The classification process splits N^0 in two new subsets: C^1 for classified posts and N^1 for unclassified ones. Then, at the $i{-}th$ iteration, the model M^i is trained using $C^0 \cup \ldots \cup C^{i-1}$. For example, at iteration 2 the model M^2 is trained using $C^0 \cup C^1$. The process is repeated in subsequent iterations until the ratio between the size of C^i and the size of N^{i-1} is lower than eps or greater that $1 - eps$. At the end, the whole set of classified posts is obtained as the union of the C^i produced at each iteration, while the remaining posts (N^2) are classified as neutrals.

ALGORITHM 1: Classification of the posts.

Input : Set of posts P, set of positive faction keyword $K_F^{\oplus}$, threshold th, minimum increment eps, maximum number of iterations max_{iters}
Output: Classified posts C

1 $C^0 \leftarrow \emptyset$;
2 $M^0 \leftarrow textualModel.build(K_F^{\oplus})$;
3 **for** $p \in P$ **do**
4   $v_b \leftarrow classify(M^0, p)$;
5   **if** $sum(v_b) = 1$ **then**
6     $f \leftarrow argmax(v)$;
7     $C^0 \leftarrow C^0 \cup \langle p, f \rangle$;
8 $C \leftarrow C^0$;
9 $N^0 \leftarrow P \setminus C^0$;
10 **for** $i = 1; i <= max_{iters}; i{+}{+}$ **do**
11   $C^i \leftarrow \emptyset$;
12   $M^i \leftarrow neuralNetwork.train(C^0 \cup \ldots \cup C^{i-1})$;
13   **for** $p \in N^{i-1}$ **do**
14     $v_p \leftarrow classify(M^i, p)$;
15   **if** $max(v_p) > th$ **then**
16       $f \leftarrow argmax(v)$;
17       $C^i \leftarrow C^i \cup \langle p, f \rangle$;
18   $C \leftarrow C \cup C^i$;
19   $N^i \leftarrow N^{i-1} \setminus C^i$;
20   **if** $\frac{|C^i|}{|N^{i-1}|} < eps \vee \frac{|C^i|}{|N^{i-1}|} > 1 - eps$ **then**
21   **break**
22 **return** C

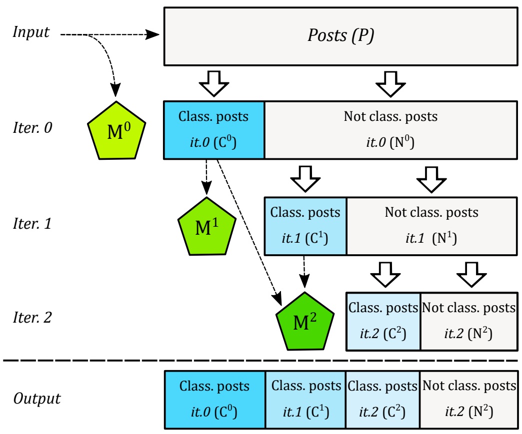

FIGURE 3: Example of the *classification of posts* algorithm terminating in three iterations.

Table 2 shows an example of post classification on ten tweets about the 2016 US presidential election. The input of the algorithm is composed of a set of tweets regarding the political event and a set of faction keywords $K_F^{\oplus}$:

- $K_{Clinton}^{\oplus}$={#voteHillary, #imwithher, #strongertogether, #hillary2016}
- $K_{Trump}^{\oplus}$={#voteTrump, #maga, #americafirst, #wakeupamerica}

At iteration 0, $K_F^{\oplus}$ is used to generate M^0, which allows to classify 5 tweets. At iteration 1, classified tweets at iteration 0 are used to train M^1. This model generates new classification rules, such as:

- since Donald Trump has been accused of sexual assault by some women, tweets with keywords *#sex* and *#woman* are classified in favor of Clinton;
- similarly, since Hillary Clinton contravenes the federal laws by using personal email account for government business, tweets with keywords *email* and *#hillary* are classified in favor of Trump.

At iteration 2, the algorithm learns other classification rules about *immigration*, a topic on which the two candidates

Iteration	TweetId	Tweet	UserId	Class
It. 0	t1	American who loves her country **#VoteTrump**	u1	Pro-Trump
	t2	Guys, she's right! It's time! #GoVote **#voteHillary**	u2	Pro-Clinton
	t3	Women detail sexual allegations against Trump #sex #woman **#ImWitHer**	u2	Pro-Clinton
	t4	List of Trump's accusers and their allegations #misconduct #sex #woman	u3	*Unclassified*
	t5	Hillary Clinton used personal email for government business **#VoteTrump**	u4	Pro-Trump
	t6	How Hillary Clinton used her personal email #scandal #hillary	u4	*Unclassified*
	t7	Hillary supports immigration reform with a pathway to citizenship #hillary	u2	*Unclassified*
	t8	A wall between the U.S. and Mexico #trump #mexico #immigration	u3	*Unclassified*
	t9	Decide the future of the US. Go vote!	u5	*Unclassified*
	t10	Let's make America great again **#MAGA**	u1	Pro-Trump
	...	...	...	...
It. 1	t4	List of Trump's accusers and their allegations #misconduct **#sex #woman**	u3	Pro-Clinton
	t6	How Hillary Clinton used her personal **email** #scandal **#hillary**	u4	Pro-Trump
	t7	Hillary supports immigration reform with a pathway to citizenship #hillary	u2	*Unclassified*
	t8	A wall between the U.S. and Mexico #trump #mexico #immigration	u3	*Unclassified*
	t9	Decide the future of the US. Go vote!	u5	*Unclassified*
	...	...	...	...
It. 2	t7	Hillary supports **immigration** reform with a pathway to citizenship **#hillary**	u2	Pro-Clinton
	t8	A wall between the U.S. and Mexico **#trump** #mexico **#immigration**	u3	Pro-Trump
	t9	Decide the future of the US. Go vote!	u5	*Unclassified*
	...	...	...	...

TABLE 2: Example of how the *classification of posts* algorithm works.

had an opposite opinion. The iterative learning process ends when the algorithm is no longer able to generate new classification rules and therefore to classify new tweets.

C. POLARIZATION OF USERS

Algorithm 2 shows the pseudo-code of the algorithm used for determining the polarization of users. The input is a collection of classified posts C (i.e., output of Algorithm 1), a filtering function $filter$ and its parameters par_f, and a polarization function $polarize$ and its parameters par_p. The output is composed of a collection of classified users U and a faction score (S) containing the polarization percentages of each faction.

As first step, the classified posts are aggregated by user to produce a dictionary (C_U), which contains the list of classified posts P_u for each user u (line 1). Two empty variables are initialized for storing the output (lines 2-3).

On each pair $\langle u, P_u \rangle$ of C_U, the algorithm performs the following operations (lines 4-7):

- It filters out all the pairs that do not match the criteria defined by the $filter$ function (line 5). For example, users who published a number of posts below a given threshold are skipped.
- Using the classified posts P_u, it computes v_s^u a vector containing the score of user u for each faction (line 6). The score vector is calculated by using the function $polarize$.
- It adds the pair $\langle u, v_s \rangle$ to U (line 7).

Then, the algorithm calculates the overall faction score S as the normalized sum of the user vector scores $\langle u, v_s^u \rangle$ (lines 8-10). Finally, the two output are returned (line 11).

The *filter* and *polarize* functions, used for analyzing the data collected for our case studies (see Section IV), have been configured as follows. Specifically, a user u is considered only if he/she fulfills the following criteria: $i)$ u posted at least *minPosts* on the political event of interest; $ii)$ it exists a faction f for which u has published more than 2/3 of his/her posts. For each user u, the $polarize$ function returns a vector score as follows: the percentage of posts written by u in favor of preferred faction f, 0 for the other factions.

ALGORITHM 2: Prediction of user polarization.

Input : Classified posts C, filtering function $filter$, filtering function parameters par_f, polarization function $polarize$, polarization function parameters par_p.
Output: Classified users U, faction score S

```
1  C_U ← aggregateByUser(C);
2  U ← ∅;
3  S ← ∅;
4  for ⟨u, P_u⟩ ∈ C_U do
5      if filter(⟨u, P_u⟩, par_f) then
6          v_s^u ← polarize(P_u, par_p);
7          U ← U ∪ ⟨u, v_s^u⟩;
8  for ⟨u, v_s^u⟩ ∈ U do
9      S ← S + v_s^u;
10 S ← S/sum(S);
11 return U, S
```

Figure 4 shows how the user polarization algorithm (Algorithm 2) works on the classified posts shown in Table 2. For each user, the posts if favor of Clinton and Trump are counted. Users who fulfill the criteria of filter function are considered and added to the set of classified users U. Then U is combined and normalized to obtain the vector S containing the overall polarization percentages.

IV. CASE STUDIES

In this section we describe and analyze two case studies: the 2018 Italian general election and 2016 US presidential

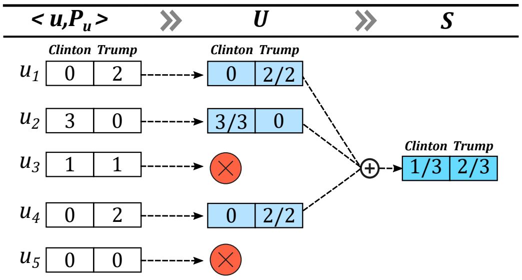

FIGURE 4: Example of how the *user polarization* algorithm works.

election. In both case studies, for each faction f_i we defined three set of keywords $K^{\oplus}_{fi}$, $K^{\ominus}_{fi}$ and $K^{\bigcirc}_{fi}$ that are respectively positive, negative and neutral keywords for faction f_i. For example, for the *Movimento 5 Stelle* ($M5S$) faction in the 2018 Italian general election, $K^{\oplus}_{M5S}$ contains keywords used to clearly support $M5S$ party (e.g., *#iovotoM5S*), $K^{\ominus}_{M5S}$ contains keywords to speak negatively about $M5S$ (e.g., *#maiM5S*), $K^{\bigcirc}_{M5S}$ contains neutral keywords for $M5S$ (e.g., *m5s* or *movimento5stelle*).

As described in Section III, *IOM-NN* exploits only positive faction keywords ($K^{\oplus}_{fi}$) for classifying posts and then for determining the polarization of users. For evaluating the accuracy of IOM-NN, we carried out an extensive comparison with the most relevant techniques used in literature:

1) *Sentiment analysis with NLP* [3] [15]. For each post, we used CoreNLP [16] for calculating a sentiment score that ranges from 0 (very negative) to 4 (very positive). The neutral keywords ($K^{\bigcirc}_{fi}$) are then used for grouping posts and calculating an overall score for each faction.
2) *Adaptive sentiment analysis* [4]. Starting from the positive and negative keywords of each faction ($K^{\oplus}_{fi}$ and $K^{\ominus}_{fi}$), this technique generates two word-polarity dictionaries, which are built from a set of posts containing such positive and negative keywords. Also in this case, a score for each faction is returned.
3) *Emoji-based polarization* [5]. This technique groups the posts of each faction by using keywords ($K^{\bigcirc}_{fi}$), then classifies their sentiment by using emojis and returns a score for each faction.
4) *Hashtag-based polarization* [6]. The posts are classified as in favor of a given faction based on the positive faction keywords ($K^{\oplus}_{fi}$). Then the posts are aggregated by users and the polarization of each user is computed.

To allow a direct comparison with the real percentages, the results obtained by the different techniques have been normalized with respect to the sum of the real ones.

A. 2018 ITALIAN GENERAL ELECTION

Here we discuss the case study carried out to analyze the polarization of a large number of Twitter users during the 2018 Italian general election. Twitter users have been classified using the polarization rules extracted from our methodology and the results have been compared to: i) official results; ii) main opinion polls collected before voting; and iii) other techniques present in the literature.

Italians voted to elect 630 deputies and 315 senators of the XVIII legislature: the results decreed the center-right coalition as the most voted, with about 37% of votes, while the most voted list was the *Movimento 5 Stelle*, which received over 32% of votes. The electorate was composed of 50,782,650 voters for the Chamber of Deputies and 46,663,202 for the Senate[4], with a turnout of about 73%, the lowest in Italian republican history.

In order to assess the validity of the proposed methodology, the analysis we carried out focused on the four most successful political factions, in decreasing order of consensus: *M5S* (*Movimento 5 Stelle*), *PD* (*Partito Democratico*), *LEGA*, *FI* (*Forza Italia*). In the following, we show how the classification model has been trained and discuss the main achieved results.

1) Models training and iteration-level results

IOM-NN has been used to classify 60,782 tweets posted by 21,883 users from February 1, 2018 to March 3, 2018 (the day before the election). By following the approach described in [6] we can assess that collected data is statistically significant for the event under analysis, since:

- All the tweets under analysis have been written in Italian, that means they have the *lang* field set to *it* (Italian). With very few exceptions, the Italian language is used only by Italians people who reside in Italy or abroad[5].
- 92% of the users who set a location in their profile, specified a region in Italy. Moreover, there is a strong correlation between the number of users that can be assigned to a region and the number of people actually living in that region according to official statistics (the Pearson correlation coefficient is equal to 0.8 with a confidence interval of 99%).
- About 98% of the Italian social media users are adults and equally divided by gender (51.2 females and 48.8 males)[6] .

IOM-NN exploits the following positive faction keywords for analyzing the collected data:

- $K^{\oplus}_{M5S}$={#iovotom5s,#m5salgoverno,#dimaiopresidente}
- $K^{\oplus}_{PD}$={#sceglipd,#iovotopd,#pdvinci}
- $K^{\oplus}_{LEGA}$={#4marzovotolega,#iovotolega,#salvinipremier}
- $K^{\oplus}_{FI}$={#berlusconipresidente,#votoforzaitalia, #4marzovotoforzaitalia}

The threshold th and the minimum increment eps have been set to 0.9 and 5% respectively. In our test, the post classification algorithm terminated in 4 iterations by annotating 23,997 tweets, which represents about 39.5% of the total. Table 3 shows the obtained results at each iteration by

[4]http://www.interno.gov.it/it/notizie/elezioni-2018-come-vota-corpo-elettorale-tessera-elettorale (in Italian)
[5]https://en.wikipedia.org/wiki/Italian_language
[6]https://wearesocial.com/it/digital-2019-italia

Iteration	Tweet input	Classified ($\mathbf{C^i}$)	Not classified ($\mathbf{N^i}$)	Perc. of class. tweets	$\frac{\lvert \mathbf{C^i} \rvert}{\lvert \mathbf{N^{i-1}} \rvert}$	Accuracy
0	60,782	3,072	57,710	5.1%	5.1%	-
1	57,710	14,676	43,034	24.1%	25.4%	0.916
2	43,034	4,677	38,357	7.7%	10.9%	0.990
3	38,357	1,572	36,785	2.6%	4.1%	0.992
Total	60,782	23,997	36,785	39.5%	-	-

TABLE 3: Partial results for each iteration achieved by IOM-NN (2018 Italian general election).

	Tweets	Users	M5S%	PD%	LEGA%	FI%	LogAcc	MAPE	MAE
Real percentages	-	-	*32.68*	*18.72*	*17.37*	*14.01*	-	-	
Averages of opinion polls	-	≈1,000	28.10	22.80	13.40	16.40	0.81	0.19	3.74
IOM-NN	***23,997***	***9,942***	***31.64***	***19.89***	***18.45***	***12.80***	***0.94***	***0.06***	***1.13***
Sentiment analysis with NLP	25,299	-	20.84	30.69	13.26	17.99	0.63	0.38	7.98
Adaptive sentiment analysis	53,488	-	21.67	18.28	21.30	21.53	0.73	0.28	5.72
Emoji-based polarization	234	-	32.25	13.76	23.20	13.57	0.84	0.16	2.92
Hashtag-based polarization	3,053	1,589	21.03	28.78	6.70	26.28	0.39	0.60	11.16

TABLE 4: Obtained percentages and accuracy evaluation on the 2018 Italian general election dataset.

specifying the number of classified and unclassified tweets, the ratio $\frac{|C^i|}{|N^{i-1}|}$ and the accuracy of the neural network.

2) Polarization of users and final results

The algorithm described in Section III-C has been used for analyzing the users who have written the 23,997 classified tweets so as to determine their polarization degree towards the considered factions. The first three rows of Table 4 shows a comparison between the official results, the average of the latest polls and the percentages obtained by *IOM-NN*. We evaluated the accuracy through different statistical indexes, comparing the obtained results with the latest opinion polls published before the elections. Considering the four most supported parties, our methodology obtained the following approval percentages: M5S 31.64%, PD 19.89%, LEGA 18.45%, and FI 12.80%. These results are extremely close to the real ones (i.e., M5S 32.68%, PD 18.72%, LEGA 17.37%, FI 14.01%), even more than the average of polls. In addition, the obtained results are characterized by very good values of log accuracy ratio, as well as a negligible value of mean percentage and absolute errors. In particular, our methodology achieved a mean average error (MAE) of 1.13 percentage points and a log accuracy ratio (LogAcc) very close to 1. On the other hand, opinion polls achieved a MAE of 3.74 percentage points and a LogAcc of 0.81, which confirm the ability of the proposed methodology to forecast election results. Figure 5 shows an info-graphic about the comparison of the real percentages, opinions polls and obtained results.

Table 4 also presents the results obtained by the other techniques in the literature (rows 4-7). These techniques have been configured with the positive faction hashtags used by IOM-NN (see $K^{\oplus}_{M5S}$, $K^{\oplus}_{PD}$, $K^{\oplus}_{LEGA}$ and $K^{\oplus}_{FI}$) and the following negative and neutral faction keywords:

- $K^{\ominus}_{M5S}$={#nom5stelle, #rimborsopolim5s, #maim5s} and $K^{\bigcirc}_{M5S}$={m5s, movimento5stelle, dimaio}

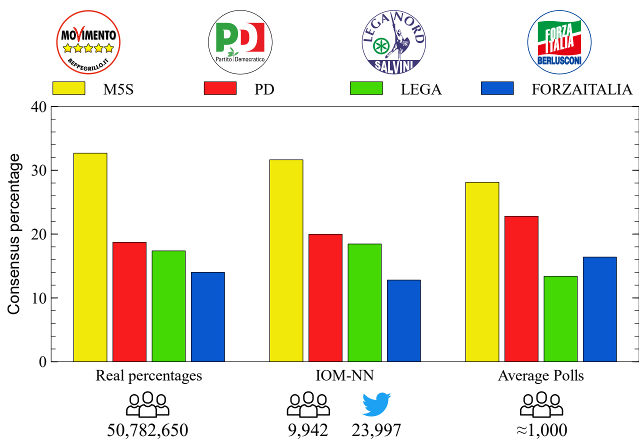

FIGURE 5: Comparison among real percentages, opinions polls and IOM-NN results (2018 Italian general election).

- $K^{\ominus}_{PD}$={#nonvotopd, #maipd, #bastapd} and $K^{\bigcirc}_{PD}$={pd, partitodemocratico, renzi}
- $K^{\ominus}_{LEGA}$={#maiconsalvini, #iononvotolega} and $K^{\bigcirc}_{LEGA}$={lega, salvini, leganord}
- $K^{\ominus}_{FI}$={#maipiuberlusconi, #stopberlusconi} and $K^{\bigcirc}_{FI}$= {forzaitalia, berlusconi}

Compared to such techniques, IOM-NN turned out to be the most accurate in estimating the voting percentages, outperforming the competitors in terms of achieved LogAcc, MAPE and MAE. Compared to the emoji- and hashtag-based techniques, IOM-NN is able to classify a much greater number of tweets and users, which ensures greater statistical representativeness of data and robustness of results.

It should be noticed that, differently from other techniques (i.e., sentiment analysis with NLP and adaptive sentiment analysis), IOM-NN is not volume-based. This means that, since it gives the same weight to each user regardless of the number of published posts, the results obtained are not influenced by users who published a large number of posts.

State	Real percentages C	Real percentages T	Average of opinion polls C	Average of opinion polls T	IOM-NN C	IOM-NN T	Sent. analysis with NLP C	Sent. analysis with NLP T	Adaptive sent. analysis C	Adaptive sent. analysis T	Emoji-based polarization C	Emoji-based polarization T	Hashtag-based polarization C	Hashtag-based polarization T
Colorado	**48.2**	43.3	**43.3**	40.4	**53.3**	38.2	**49.1**	42.4	45.4	46.1	**52.1**	39.4	**49.9**	41.7
Florida	47.8	**49.0**	46.4	**46.6**	49.9	46.9	46.1	**50.7**	49.3	47.5	49.3	47.5	48.4	48.4
Iowa	41.7	**51.1**	41.3	**44.3**	44.6	**48.2**	49.5	43.3	47.0	45.8	46.9	45.9	42.0	**50.8**
Michigan	47.3	**47.5**	45.4	42.0	43.5	**51.4**	49.8	45.0	47.5	47.3	48.2	46.6	38.1	**56.7**
New Hamp.	**47.0**	46.6	**43.3**	42.7	**49.0**	44.6	**48.2**	45.4	45.6	48.0	44.0	49.6	45.5	48.1
N.Carolina	46.2	**49.8**	46.4	46.4	47.8	**48.2**	50.6	45.4	49.2	46.8	48.7	47.3	40.7	**55.3**
Ohio	43.6	**51.7**	42.3	**45.8**	46.9	**48.4**	51.3	44.0	48.4	46.9	50.9	44.4	42.6	**52.7**
Pennsylvania	47.9	**48.6**	46.2	44.3	54.9	41.6	**50.3**	46.2	48.1	**48.4**	54.4	42.1	50.2	46.3
Virginia	**49.8**	44.4	**47.3**	42.3	**53.2**	41.0	49.0	45.2	46.3	47.9	**48.2**	46.0	45.1	49.1
Wisconsin	46.5	**47.2**	46.8	40.3	45.8	**47.9**	48.5	45.2	47.4	46.3	49.8	43.9	40.8	**52.9**
Correctly classified	-		**6/10**		**8/10**		**4/10**		**1/10**		**2/10**		**6/10**	
Tweets	-		-		718,425		775,277		818,403		23,937		409,146	
Users	-		≈10,000		125,891		-		-		-		78,430	
Mean LogAcc	-		0.97		0.93		0.93		0.95		0.93		0.93	
Mean MAPE	-		0.03		0.07		0.07		0.05		0.07		0.07	
Mean MAE	-		1.57		3.19		3.21		2.37		3.48		3.25	

TABLE 5: Obtained percentages and accuracy evaluation on the 2016 US presidential election dataset. For each technique, when the winning candidate is correctly identified, the percentage is written in bold.

Since CoreNLP provides a well-trained model for sentiment analysis only for the English language, all the tweets downloaded in Italian have been translated in English before being processed.

B. 2016 US PRESIDENTIAL ELECTION

After the presentation of the Italian use case, here we discuss the analysis we carried out on the 2016 US presidential election, which was characterized by the rivalry between Hillary Clinton and Donald Trump.

The analysis has been performed on data collected for ten US Swing States: Colorado, Florida, Iowa, Michigan, Ohio, New Hampshire, North Carolina, Pennsylvania, Virginia, and Wisconsin. Swing states are those characterized by greater political uncertainty, in which neither major political party holds a lock on the outcome of presidential elections. These states are considered of strategic importance, as their votes have a high probability of being the deciding factor in a presidential election. For each state, data have been collected through the standard Search Twitter API, which allows for collecting tweets published in a given area or place. Overall about 2.5 million of tweets, posted by 521,291 users, have been collected from October 10, 2016 to November 7, 2016 (the day before the election). From such data we filtered out all the tweets posted by users with a not defined location or with a location that does not belong to any of the considered states. Filtered data (818,403 tweets posted by 141,959 users) are statistically significant for the event under analysis, since:

- All the tweets under analysis have the *lang* field set to *en* (English).
- For each state, there is a strong correlation between the number of analyzed users and the number of people actually living in that state according to official statistics (the Pearson correlation coefficient is equal to 0.95 with a confidence interval of 99%).
- About 94% of the social media users in USA are adults (at least 18 years old) and almost equally divided by gender (42.7% females and 57.3% males)[7].

The following keywords have been used in our experiments:

- $K^{\oplus}_{Clinton}$={#voteHillary, #imwithher, #strongertogether, #hillary2016},
- $K^{\ominus}_{Clinton}$={#neverhillary, #lockherup}
- $K^{\bigcirc}_{Clinton}$={clinton, hillary, democrats, dems}
- $K^{\oplus}_{Trump}$={#voteTrump, #maga, #americafirst, #wakeupamerica}
- $K^{\ominus}_{Trump}$={#nevertrump, #dumpfortrump}
- $K^{\bigcirc}_{Trump}$={trump, donald, republicans, gop}

Table 5 shows the results obtained using IOM-NN in comparison with the real voting percentages, the main opinion polls, and the other related techniques. For each state in the table, we reported the results obtained by the two candidates, where "*C*" stands for Clinton and "*T*" for Trump. In addition, for the winning candidate that has been correctly identified, the value is written in bold. As shown in Figure 6, IOM-NN is able to correctly identify the winning candidate in 8 out of 10 cases, outperforming the opinion polls that correctly classifies 6 out 10 states.

Also in comparison with the other techniques, IOM-NN turned out to be the most accurate in discovering the winning candidate. In fact, emoji- and hashtag-based techniques classified a much smaller amount of tweets and correctly identified the winning candidate in 6 and 3 cases respectively. The results of the adaptive sentiment analysis were very poor, since it correctly identified the winning candidate in only one case, while sentiment analysis with NLP produced right predictions in 4 out 10 cases. Compared to IOM-NN, some techniques (i.e., adaptive sentiment analysis and opinion polls) achieved slightly better results in terms of log

[7]https://www.statista.com/statistics/376128/facebook-global-user-age-distribution/

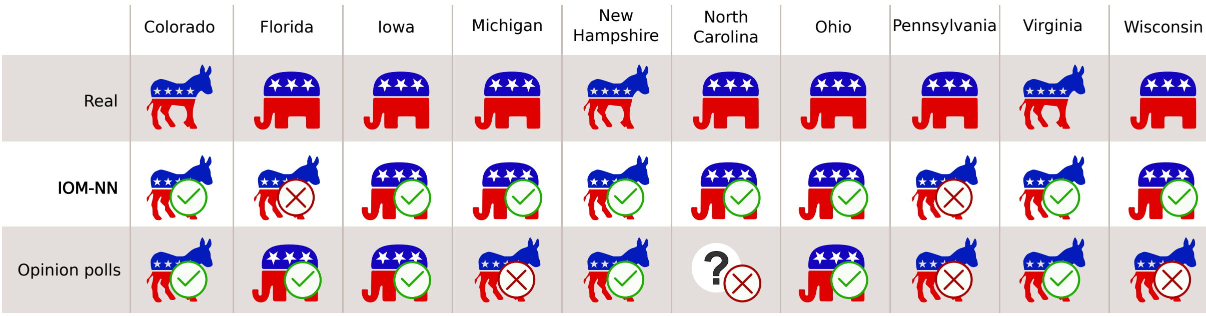

FIGURE 6: Comparison among the real winning candidate and that identified by IOM-NN and opinions polls. The *Democratic Donkey* symbolizes the party of Hillary Clinton, while the *Republican Elephant* that of Donald Trump.

accuracy ratio, MAPE, and MAE, because their predictions are quite balanced by assigning almost the same score to the two candidates in the different states.

V. CONCLUSION

This paper proposes a methodology, named IOM-NN, for estimating the polarization of public opinion regarding political events characterized by the competition of factions or parties. The designed methodology uses an automatic incremental procedure based on feed-forward neural networks for analyzing the posts published by social media users. IOM-NN can be considered an alternative technique to traditional opinion polls since it is able to capture the opinion of a larger number of people more quickly and at a lower cost. In addition, it can capture the public opinion for topics perceived as embarrassing or offensive, for which people are reluctant to declare their true opinion during the polls.

IOM-NN has been validated through two case studies that analyzed the polarization of a large number of Twitter users during the 2018 Italian general election and 2016 US presidential election. The achieved results are very close to the real ones and more accurate than the average of the opinion polls, thus revealing the high accuracy and effectiveness of the proposed approach. Moreover, our approach has been compared to the most relevant techniques used in the literature (sentiment analysis with NLP, adaptive sentiment analysis, emoji- and hashtag- based polarization). Results show that IOM-NN achieved the best accuracy in estimating the polarization of social media users.

As future work, IOM-NN can be adapted to process real-time data, also coming from different sources (e.g, e-commerce and news sites blogs). Furthermore, its effectiveness can be evaluated also in other application domains, such as reputation evaluation of companies and competitive product analysis.

ACKNOWLEDGMENT

This work has been partially supported by the SMART Project, CUP J28C17000150006, funded by Regione Calabria (POR FESR-FSE 2014-2020) and by the ASPIDE Project funded by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 801091.

REFERENCES

[1] L. Belcastro, F. Marozzo, and D. Talia, "Programming models and systems for big data analysis," International Journal of Parallel, Emergent and Distributed Systems, vol. 34, no. 6, pp. 632–652, 2019.
[2] D. Talia, P. Trunfio, and F. Marozzo, Data Analysis in the Cloud. Elsevier, October 2015, iSBN 978-0-12-802881-0.
[3] L. Oikonomou and C. Tjortjis, "A method for predicting the winner of the usa presidential elections using data extracted from twitter," in 2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA_CECNSM). IEEE, 2018, pp. 1–8.
[4] I. El Alaoui, Y. Gahi, R. Messoussi, Y. Chaabi, A. Todoskoff, and A. Kobi, "A novel adaptable approach for sentiment analysis on big social data," Journal of Big Data, vol. 5, no. 1, p. 12, 2018.
[5] D. Chin, A. Zappone, and J. Zhao, "Analyzing twitter sentiment of the 2016 presidential candidates," American Journal Of Science and Research, 2016.
[6] F. Marozzo and A. Bessi, "Analyzing polarization of social media users and news sites during political campaigns," Social Network Analysis and Mining, vol. 8, no. 1, p. 1, 2018.
[7] L. Belcastro, R. Cantini, F. Marozzo, D. Talia, and P. Trunfio, "Discovering political polarization on social media: A case study," in The 15th International Conference on Semantics, Knowledge and Grids, Guangzhou, China, 2019, to appear.
[8] M. Bilal, A. Gani, M. I. Ullah Lali, M. Marjani, and N. Malik, "Social profiling: A review, taxonomy, and challenges," Cyberpsychology, Behavior, and Social Networking, 2019.
[9] K. Jaidka, S. Ahmed, M. Skoric, and M. Hilbert, "Predicting elections from social media: a three-country, three-method comparative study," Asian Journal of Communication, vol. 29, no. 3, pp. 252–273, 2019.
[10] M. Bilal, A. Gani, M. Marjani, and N. Malik, "Predicting elections: Social media data and techniques," in 2019 International Conference on Engineering and Emerging Technologies (ICEET). IEEE, 2019, pp. 1–6.
[11] M. Gaurav, A. Srivastava, A. Kumar, and S. Miller, "Leveraging candidate popularity on twitter to predict election outcome," in Proceedings of the 7th workshop on social network mining and analysis. ACM, 2013, p. 7.
[12] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in Fourth international AAAI conference on weblogs and social media, 2010.
[13] P. Burnap, R. Gibson, L. Sloan, R. Southern, and M. Williams, "140 characters to victory?: Using twitter to predict the uk 2015 general election," Electoral Studies, vol. 41, pp. 230–233, 2016.
[14] F. M. F. Wong, C. W. Tan, S. Sen, and M. Chiang, "Quantifying political leaning from tweets, retweets, and retweeters," IEEE transactions on knowledge and data engineering, vol. 28, no. 8, pp. 2158–2172, 2016.
[15] S. Alashri, S. S. Kandala, V. Bajaj, R. Ravi, K. L. Smith, and K. C. Desouza, "An analysis of sentiments on facebook during the 2016 us presidential election," in 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2016, pp. 795–802.

[16] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit,” in Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 55–60.
[17] M. K. Olorunnimbe and H. L. Viktor, “Tweets as a vote: Exploring political sentiments on twitter for opinion mining,” in International Symposium on Methodologies for Intelligent Systems. Springer, 2015, pp. 180–185.
[18] A. Bovet, F. Morone, and H. A. Makse, “Predicting election trends with twitter: Hillary clinton versus donald trump,” CoRR, vol. abs/1610.01587, 2016.
[19] F. Bodendorf and C. Kaiser, “Detecting opinion leaders and trends in online social networks,” in Proceedings of the 2nd ACM workshop on Social web search and mining. ACM, 2009, pp. 65–68.
[20] E. M. Hafner-Burton and A. H. Montgomery, “Centrality in politics: How networks confer power,” 2010.
[21] P. R. Miller, P. S. Bobkowski, D. Maliniak, and R. B. Rapoport, “Talking politics on facebook: Network centrality and political discussion practices in social media,” Political Research Quarterly, vol. 68, no. 2, pp. 377–391, 2015.