Sentiment Analysis of Events in Social Media

Conference Paper · September 2019

DOI: 10.1109/ICCP48234.2019.8959677

3 authors:

Alexandru Petrescu
Polytechnic University of Bucharest

Ciprian-Octavian Truică
Aarhus University

Elena Simona Apostol
Polytechnic University of Bucharest

Sentiment Analysis of Events in Social Media

Alexandru Petrescu[1], Ciprian-Octavian Truică[2], Elena-Simona Apostol[3]
Computer Science and Engineering Department, Faculty of Automatic Control and Computers
University Politehnica of Bucharest, Bucharest, Romania
Email: [1]apetrescu0506@stud.acs.upb.ro, [2]ciprian.truica@cs.pub.ro, [3]elena.apostol@cs.pub.ro

***Abstract*—The growing popularity of Online Social Networks has open new research directions and perspectives for content analysis, i.e., Network Analysis and Natural Language Processing. From the perspective of information spread, the Network Analysis community propose Event Detection. This approach focuses on the network features, without an in-depth analysis of the textual content, summarization being a preferred method. Natural Language Processing analyses only the textual content, not integrating the graph-based structure of the network. To address these limitations, we propose a method that bridges the two directions and integrates content-awareness into network-awareness. Our method uses event detection to extract topics of interest and then applies sentiment analysis on each event. The obtained results have high accuracy, proving that our method determines with high precision the overall sentiment of the detected events.**

***Index Terms*—Event Detection, Sentiment Analysis, Social Networks Analysis, MABED, Online LDA, SVM, Logistic Regression**

I. Introduction

With the increasing use of Online Social Networks and the many challenges related to analyzing and mining the textual content generated daily by this new type of media, new methods that extract discussions and topics of interest and the opinion that users have about the need to be developed. Multiple research communities are working on analyzing and making sense of the contact generated daily by these networks, namely the Network Analysis and Natural Language Processing communities. The Network Analysis community develop approaches dealing with information spread and mitigation of harmful content using Event Detection, while the Natural Language Processing community develop methods to analyze the user opinion using Sentiment Analysis. Event Detection (ED) is used to detect the impact and spread of topics on Social Network. Sentiment Analysis (SA) uses supervised and unsupervised learning to determine the polarity of textual data. The dimensions or the actual classification of a text is different for each method, ranging from binary (positive/negative or neutral if we cannot tell) to multi-dimensional approaches (happiness, contempt, surprise and so on).

Our method brings together the research of these two isolated communities, i.e., Network Analysis and Natural Language Processing, by combining Network Analysis with Machine Learning for detecting event sentiments in Online Social Networks. We aim to put the solutions developed in these two areas at work in combination with each other, facilitating the detection and spread of events in a content-aware and network-aware manner.

This paper presents a new approach that combines event detection and sentiment analysis methods on social media networks. Although for our experiments we consider only the Twitter platform, our solution can be easily adapted to other types of social media data. Twitter is a micro-blogging platform that supports only relatively small posts with a maxim size of 140 characters. These posts can also contain images, videos, mentions or tags. Twitter is a widely used platform in textual processing related to social interaction and human behavior patterns. As the relational system of Twitter is modeled like a directional graph, it is helpful for information diffusion or anomaly detection models.

This paper is structured as follows. Section II presents a survey of the current state of the art methods for Event Detection and Sentiment Analysis. The proposed solution and its main functionalities are described in Section III. Likewise, in this section, we briefly describe how each component interacts with the overall platform and how each chosen algorithm works. Section IV showcases the obtained results, analyzes them and gives some directions for future trials, using the chosen Event Detection and Sentiment Analysis methods. In the final section (Section V) of this article, we conclude and we present several new directions and improvements for the proposed solution.

II. Related Work

In this section, we will present an overview of the current state of the art related to Event Detection (ED) and Sentiment Analysis (SA).

A. Event Detection

Information diffusion is the field that analyses how the states of individual nodes in a graph evolve over time [9]. It is used to detect user behavior and information spreading across social networks through monitor the magnitude (the number of users it reaches) and the lifespan of a topic. In the case of Online Social Networks, these topics are named bursty topics [6], i.e., topics that spread information better than the average.

In [13], the authors survey the state-of-the-art methods for ED and build and taxonomy. Their analysis focuses on detecting bursty topics on the Social Media platform Twitter. The analysis includes multiple algorithms and methods, e.g. PT (Peaky Topics) [19] that uses normalized term frequency, TSTE (Temporal and Social Terms Evaluation) [4]

978-1-7281-4914-1/19/$31.00 © 2019 IEEE

that uses a five-step approach considering both temporal and social properties, MACD (Moving Average Convergence Divergence) [15] that uses the trend momentum of a topic, SDNML (Sequentially Discounting Normalized Maximum Likelihood) [21] specialized in tweets with media and URLs, and OLDA (Online Latent Dirichlet Allocation) [2] which incrementally update the topic model at each time slice using the previously generated model as a prior. Experimental results prove that the information is diffused by users that appear as the node for which the graph partitions [11].

Another approach uses machine learning techniques and the inference of time-dependent diffusion probabilities from a multidimensional analysis of individual behaviors and builds communities (sub-graphs) [12]. To build these sub-graphs and to have a better view of the information propagation, three dimensions are considered: semantics, social, and time. The experimental results that using machine learning to infer the information diffusion has high accuracy on the tested datasets.

MABED (Mention-Anomaly-Based Event Detection) [10] is a method that uses user mentions to detect events and their impact on Social Networks. The approach considers for each user at most 2 levels of followers (i.e., the direct followers of the user and the direct followers of the followers) to build communities. The experimental results prove that MABED has higher topic readability and temporal precision than the algorithm used as the baseline, i.e., PS (Peakiness Score) [19] and EDCoW ((Event Detection with Clustering of Wavelet-based Signals) [23].

B. Sentiment Analysis

SA is a field in Natural Language Processing that analyzes user opinions and emotions from written language [16]. There are two main approaches for SA: i) Lexicon Based (unsupervised methods that use word polarity to classify textual data), and ii) Machine Learning (supervised methods that use polarity labeled dataset to build a model).

In [20], the authors present a comprehensive study of the two main SA approaches. The experiments show that the approach that achieves the best results uses Multivariate Analysis of Variance (MANOVA) to exact features. This method of feature extraction manages to highlight the relation between dependent and independent variables for the task of SA.

SA was used to detect the opinions presented in tweets [22]. In this study, the authors used a scale from 0 to 5 to represent the sentiment intensity for both positive and negative tweets, where 0 is the lack of intensity and 5 means a strong intensity. The approach also uses the concept of Peak Hours (i.e., 5 hours before and after the maximum volume hour) to detect events. The results show that events with negative sentiments are more frequent, whereas positive predictions were accurate only during peak hours.

In [14], the authors proposed the use of metadata extracted from tweets (i.e., hashtags, emoticons, and abbreviations) to enhance the feature space for supervised methods. This method uses a hybrid approach combining multiple lexicons containing the metadata with classification algorithms. The experimental results show that this hybrid approach increases the accuracy of detecting positive, negative, and neutral sentiments.

The overuse of adjectives in a phrase influence the accuracy of sentiment detection. To overcome this limitation, one method proposes to classify sentences with multiple adjectives in the 2 classes (i.e., positive and negative) [18]. This method shows that with the increase of adjectives the accuracy of detecting neutral tweets decreases. For the experimental validation, the authors used SVM and Naïve Bayes on two small datasets, between 100 and 2917 sentences.

III. METHODOLOGY

In this section, we present our proposed solution for accurate event and sentiment detection in social media. This solution offers two classes of tasks: Event Detection (ED) and Sentiment Analysis (SA) tasks. Both ED and SA uses a raw frequency-based embedding as a method to word vectors. This approach simply counts the appearance of each word in a document using NLTK Twitter Tokenizer.

A. Proposed Architecture

The architecture of the solution is presented in Figure 1.Its main functionality and modules are as follows:

- *Training Engine* is used for the training step of the ML models. It can be run either locally if the machine is powerful enough or on a stand-alone remote machine that can be accessed through the internet to trigger an ED task.
- *TwitterMining Module* is constantly listening for twitter data, in order to fetch and pass it on to the CosmosDB managed the environment.
- *Storage Module* uses multiple data sources depending on the task at hand: Azure Storage for data persistence, CSV for training modules and JSON for twitter mining.
- *AzureML Module* uses the models trained by the *Training Engine* and offers Web API interfaces for them. It also grabs data from the *Storage Module* and exposes several APIs to the WEB Manager Module.
- *WEB Manager Module* offers both server and client-side functionalities. It is managing all the other modules and also the user requests. Based on the nature of the user request, it can either submit a task to the *AzureML Module*, look in the *Metadata Module* for cached results or present the data with the help of the *PowerBI Module*.
- *Metadata Module* is used as a data source for the *PowerBI Module* and as a cache server by the *WEB Manager Module*.
- *PowerBI Module* uses the data from the tasks submitted by the *WEB Manager Module* and feeds reports to it.

B. Event Detection Tasks

For ED, we use the Mention-Anomaly-Based Event Detection (MABED) and Online LDA (OLDA) algorithms. Both approaches detect bursty topics either on cleaned or raw text. The cleaned text results by removing the stop words and any non-character from the raw text and applying

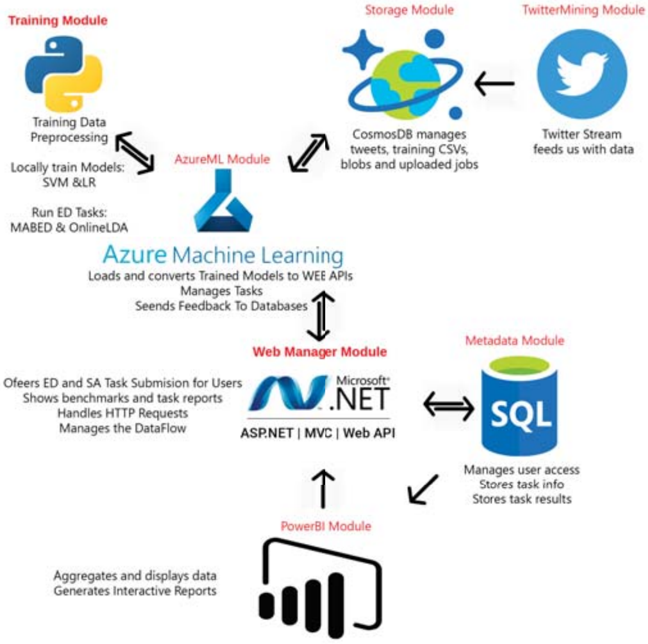

Figure 1: Architecture diagram

lemmatization. After this preprocessing step, we can see some relevant results, as shown in Figure 2, otherwise, all the stop words would have been seen as important.

Mention-Anomaly-Based Event Detection [10] is an efficient statistical method used for detecting events in social networks since it is immune to social media bias consisting of unrelated texts on a given topic. Although social networks can get spammy, meaning that messages with no actual intent are posted around certain hours, this method has proved good in filtering the irrelevant content. MABED is not always viable since, without external context, it can sometimes distort some type of events.

MABED [10] uses Equation (1) to detect bursty topics, with the components from Equation (2), Equation (3) and Equation (4).

$$w_q = \frac{\rho_{O_{t,t'_q}} + 1}{2} \tag{1}$$

$$\rho_{O_{t,t_{q'}}} = \frac{\sum_{i=a+1}^{b} \cdot A_{t,t_{q'}}}{(b-a-1) \cdot A_t \cdot A_{t_{q'}}} \tag{2}$$

$$A_t^2 = \frac{\sum_{i=a+1}^{b} \cdot (N_t^i - N_t^{i-1})^2}{(b-a-1) \cdot A_t \cdot A_{t_{q'}}} \tag{3}$$

$$A_{t'_q}^2 = \frac{\sum_{i=a+1}^{b} \cdot (N_{t'_q} - N_{t'_q}^{i-1})^2}{(b-a-1) \cdot A_t \cdot A_{t_{q'}}} \tag{4}$$

Where:

1) t and t_q are therms
2) $I = [a; b]$ is the time interval
3) N_t^i and $N_{t'_q} t^i$ the time-series with the correlation coefficient proposed in [8]

OnlineLDA [2], is an update on Latent Dirichlet Allocation [3] which uses a non-Markov on-line LDA Gibbs sampler topic model. This model is capable of detecting bursty topics by capturing thematic patterns and identifying emerging topics and their changes over time. This approach is sometimes better than the original LDA approach, as it updates the previous model at each time frame.

OLDA is a hierarchical Bayesian network that relates words and documents through latent topics using a probabilistic formula for a given term to be assigned to a given topic in the (current) context Equation ((5)).

$$P(z_i = j | z\neg_i, w_{di}, \alpha, \beta) \propto \frac{C_{w_{\neg i}j}^{VK} + \beta_{w_i,j}}{\sum_{v=1}^{V} C_{v_{\neg i}j}^{VK} + \beta_{v,j}} \times \frac{C_{d_{\neg i}j}^{DK} + \alpha_{w_i,j}}{\sum_{v=1}^{V} C_{d_{\neg i}j}^{DK} + \alpha_{d,j}} \tag{5}$$

Where:

- D is the total number of documents; K is the number of topics
- V is the total number of unique words
- $C_{w_{\neg i}j}^{VK}$ is the number of times word *w* is assigned to topic *j*, not including the current token instance *i*
- $C_{d_{\neg i}j}^{DK}$ is the number of times topic *j* is assigned to some word token in document *d*, not including the current instance *i*
- $z_{\neg i}$ are all other word tokens; w_{di} are the unique word associated with the *i*-th token in document *d* at the current time
- β_k is the V vector of priors for topic *k* at the current time
- α_k is the K vector of priors for document *k* at the current time

C. Sentiment Analysis Tasks

Support vector machines (SVM) [5] is a supervised learning algorithm used in ML for classifications. Given a tagged dataset, SVM builds a function that separates the space of this given dataset in two classes In SA, this type of model is used for binary classification, where the classes are positive and negative representing the tweets' sentiments. For the SA tasks we employ a linear SVM that uses the function from Equation (6) that tries to separate the values of the training set using Equation (7). We consider that -1 and 1 are the numerical equivalent of the two classes.

$$f(x) = w^T * x + b \tag{6}$$

$$y_i * f(w_i^T * x_i + b) = 1 \tag{7}$$

Where:

- x_i is the "i"-th example
- y_i is the actual class of the the example x_i
- w is the weight matrix that we need to compute
- b is the bias

Logistic regression [7] is a statistical model, widely used, that in its basic form uses a logistic function to model a binary classification for the given dataset. It will try to build a function, a linear one for this case that will summarize the input data. Logistic regression is a supervised method that behaves better than Linear Regression [17], when applied on a sparse dataset, and builds classification models within a probabilistic context [1]. We consider that a dataset is sparse when the matrix of its representation has a lot of zeros and a few one values, like ours, will be. If we represent the words depending on their appearance into a certain block of text (tweet) or not we will get a sparse matrix since the tweets have at most 140 characters.

The probability for this last presented model is given by Equation (8), which has the same notations as to the ones from Equation 6. As can be observed, this is a Sigmoid-type function.

$$P_w(y = \pm 1|x) \equiv \frac{1}{1 + e^{-y \cdot w^T \cdot x}} \tag{8}$$

D. Sentiment Analysis of Events

The ED step produces a sentence that will be the topic of an event. This result contains the key-words of all the tweets that are included in that topic. This implies that we can apply SA over the results, in order to guess if the event has a positive or negative impact over the network.

Combining the magnitude and lifespan of the events with the sentiment label, we can create a better view of the tweet dataset and understanding of the user behavior and the topics of interest. For example, we can determine if negative topics have a higher impact or longer duration. Also, if we consider the individual tweets belonging to an event in a topic we can construct communities based on their opinions.

Algorithm 1 Sentiment Analysis of Events Algorithms

Require: a document set D
Ensure: a event-sentiment set E_S
1: $E_S = \emptyset$
2: $D_C = EDTP(D)$
3: $E = ED(D_C)$
4: **for each** $e_i \in E$ **do**
5: $\quad D^{(i)} = extract(e_i, D)$
6: $\quad D_C^{(i)} = SATP(D^{(i)})$
7: $\quad S = \emptyset$
8: $\quad$ **for** $d \in D_C^{(i)}$ **do**
9: $\quad\quad S = S \cup s | S = SA(d)$
10: $\quad s_j^{(i)} = MaxSA(S)$
11: $\quad E_S = E_S \cup (e_i, s_j^{(i)})$
12: **return** E_S

The proposed method is implemented through Algorithm 1. The input is a set of documents D whereas the output is a event-sentiment set $E_S = \{(e_i, s_j^{(i)}) | e_i \in E \wedge s_j^{(i)} \in S\}$ that contains the sentiment $s_j^{(i)} \in S$ for each event $e_i \in E$, where E are the events and S are the sentiments and. The fist step initializes the event-sentiment set (Line 1), then uses text prepossessing for ED ($EDTP$) (Line 2), and extract the list of events (Line 3). During the iteration of the list of events E (Line 4) we extract the list of documents for the event $D^{(i)} = extract(e_i, D)$ (Line 5), apply the text preprocessing steps for SA and extract the clean document set $D_C^{(i)} = SATP(D^{(i)})$ (Line 6). An empty list $S = \emptyset$ (Line 7) for storing the sentiment for each document is initialized. For each document d in set $D_C^{(i)}$ detects the sentiment (Lines 8 and 9) and update the list $S = S \cup s | S = SA(d)$. After these steps, we label the events with the sentiment with the most number of occurrences (Lines 10 and 11).

IV. Experimental Results

In this section, first, we present the used dataset for Event Detection (ED) and Sentiment Analysis (SA) tasks. Secondly, we analyze the results obtained for the two types of tasks individually. Finally, we present the result of detecting the sentiments for each individual detected event. The source code is publicly available on GitHub [1]

A. Dataset

As a dataset, we used Sentiment140 [2]. This dataset has the following fields: i) the *text*, the *date*, and the sentiment *label*. The *text* field is used in both tasks, after preprocessing is applied. The *date* is used in the ED task and the *label* is use only in the SA task.

To improve the accuracy for detecting the magnitude and lifespan of an event, we use text preprocessing. Thus, for the ED task, we apply the following text preprocessing steps: *i)* lowercase the text, *ii)* split the text into tokens, *iii)* remove stop words, *iv)* remove punctuation, *v)* lemmatize each word and extract the lemma, *vi)* weighed each term using the raw frequency ($f_{t,d}$), i.e., the number of times a term appears in a document. For testing, we constructed three different prepossessed texts: *a)* MT (Minimal Text) uses only steps i), ii), iii), and vi) for prepossessing, *b)* PT (Pure Text) uses only steps i), ii), iii), iv), and vi) for preprocessing; and *c)* CT (Clean Text) includes all the preprocessing steps. By employing these steps, we manage to minimize the vocabulary and improve the accuracy of the ED methods. Figure 2 presents the word cloud after text preprocessing.

For the SA task, we apply the following text preprocessing steps: *i)* split the text into tokens, *ii)* remove punctuation, *ii)* weighed each term using the raw frequency ($f_{t,d}$). Using these steps we obtain the SCT (Sentiment Clean Text). In order to achieve better accuracy we further preprocess the text to obtain SFE (Sentiment Clean Text with Feature Engineering): *i)* keep the case of words, duplicate letter in words, and stop words to preserve the sentiment polarity, *ii)* expand contractions (e.g., ”don’t” becomes ”do not”), *iii)* enhance the text using feature engineering (FE) by combining the negation with the words in front.

[1]Repository https://github.com/xander96/EventDetection-SentimentAnalysis
[2]Sentiment140 http://sentiment140.com/

Figure 2: Event Detection Text Preprocessing Word Cloud

We applied Event Detection on the entire dataset, while we split the corpus into three subsets to see how each Sentiment Analysis algorithm scales. Thus, for the Sentiment Analysis tasks we have the following subsets: *i)* C_1 contains 20 000 random tweets with the labels equally distributed so it would not over-fit, *ii)* C_2 contains 500 000 random tweets with the labels equally distributed so it would not over-fit and *iii)* C_3 containing the entire dataset of 1 600 233 tweets. Table I presents the number of features without and with FE.

Table I: Number of features for Sentiment Analysis

Dataset	Features	FE Features
C_1	30 651	31 018
C_2	307 890	311 240
C_3	684 492	691 646

B. Event Detection

We set both algorithms to return 50 events, each event summarized by the top-10 most relevant terms. The lifespan of the events is between April and July 2009. For this set of experiments, we used the three text prepossessing strategies for ED, i.e, MT, PT, and CT.

1) MABED: MABED manages to extract accurate events and bursty topics (Figure 3), but the topic of readability is influenced by the preprocessing strategy we employed. The text processing is done with MT and PT give more human-readable results, while when using CT, MABED manages to summarize the topics, even more, providing higher magnitudes and life span.

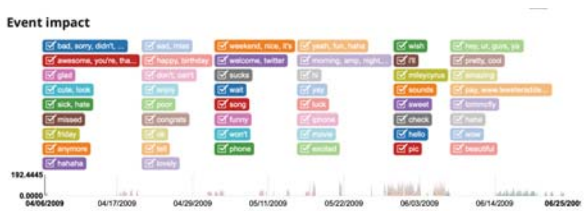

Figure 3: MABED - Event Impact

2) OLDA: OLDA produces topics that are more human-readable than the ones of MABED (Figure 4). To summarize a topic, OLDA uses the entire vocabulary. As in the case of MABED, OLDA with CT manages to better generalize the topics and cover more tweets.

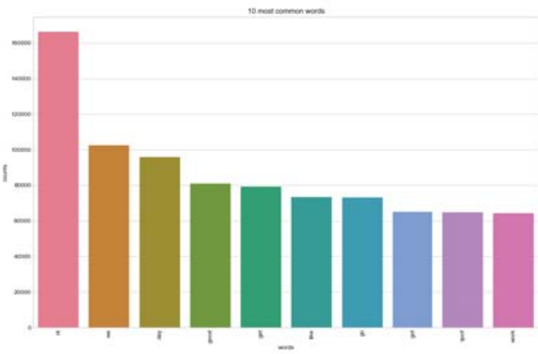

Figure 4: OLDA - Most used words

3) Results Comparison: We can observe that MABED and OLDA manage to detect different emerging events when analyzing the most representative topic keywords using the text preprocessing CT, although some are the same (Figure II). MABED provides better results, as it also uses fewer words.

C. Sentiment Analysis

For the Sentiment Analysis tasks, we use Logistic Regression (LR) and Support Vector Machine (SVM) together with SCT and SFE text preprocessing strategies. We use the Area Under the Curve (AUC), Accuracy, Precision, and Recall for evaluating the results, and K-Folds for the evaluation scores.

1) LR Results: The first set of experiments for LR uses SCT (Table III). The evaluation scores improve with the size of the text.

When using LR with SFE (Table IV), the evaluation scores worsen for the small subset of documents C_1, but they improve with the scale.

Comparing the two experiments, we can observe that the improvement in the evaluation scores is minimal w.r.t. the text preprocessing strategy we employed, i.e., overall for all the measures ~ 0.004. Furthermore, the scores improvement w.r.t. the scale is also minimal, i.e., overall for all the measures ~ 0.05. Thus we can conclude that the number of features does not impact any of the measures.

2) SVM: For the SVM approach, we have used a **linear kernel** as the classes are **balanced**. We set the penalty parameter of the error term $C = 0.1$, value obtained after hyperparameter tuning.

The first set of experiments for SVM uses SCT (Table III). The evaluation scores improve with the increasing number of used features, with an overall increase of almost ~ 0.5.

When using SVM with SFE (Table IV) we observe the same ascending trend of the score improvement w.r.t. the scale of the dataset.

As in the case of LR, we can observe that the improvement in the evaluation scores is minimal w.r.t. the text preprocessing strategy we employed, i.e., overall for all the measures ~ 0.01. Furthermore, the overall score improvement for all the

Table II: Similar topics example for MABED and OLDA

MABED Topic	OLDA Topic
nt sleep wa na morning gon early class getting school	work sleep hour tired done early get go still woke
movie look read wa song watching getting	sorry movie hear watch goodnight exam season mileycyrus love watched
phone wa miss ah finally quot doe lt facebook amp	twitter friend best leaving wont bye facebook know lol let
nt night feel show wa hour fun doe	fun bit show tonight much little bbq going passed wine

Table III: LR with SCT

Dataset	AUC	Accuracy	Precision	Recall
C_1	0.756	0.756	0.703	0.766
C_2	0.790	0.790	0.727	0.802
C_3	0.800	0.800	0.739	0.810

Table IV: LR with SFE

Dataset	AUC	Accuracy	Precision	Recall
C_1	0.752	0.752	0.690	0.752
C_2	0.794	0.794	0.730	0.807
C_3	0.804	0.804	0.742	0.815

Table V: SVM with SCT

Dataset	AUC	Accuracy	Precision	Recall
C_1	0.803	0.729	0.707	0.772
C_2	0.852	0.778	0.799	0.742
C_3	0.876	0.803	0.795	0.820

Table VI: SVM with SFE

Dataset	AUC	Accuracy	Precision	Recall
C_1	0.816	0.748	0.725	0.796
C_2	0.857	0.781	0.763	0.815
C_3	0.877	0.806	0.795	0.825

measures w.r.t. the scale is also minimal, i.e. ~ 0.05. Thus we can conclude once more than the number of features does not impact any of the measures.

3) Results Comparison: SVM provides better evaluation scores than LR w.r.t. the number of features, but the performance improvement is minimal, i.e., ~ 0.05. Using SFE improves the evaluation scores for both algorithms. In conclusion, we affirm that both algorithms present comparable evaluation scores results. From a runtime perspective, LR is much faster than SVM (Table VII), as the model construction takes minutes (m) for LR, while for SVM it takes days (D). We can conclude the LR algorithm is the best choice for constructing the SA model. Although the SVM achieves a very small increase in evaluation score over LR, the waiting time for building the model is not feasible.

Table VII: SVM vs. LR runtime comparison

	SCT			SFE		
Algorithm	C_1	C_2	C_3	C_1	C_2	C_3
LR	2 m	5 m	20 m	1.33 m	4 m	11 m
SVM	2 D	32 D	65 D	1 D	4 D	62 D

D. Sentiment in Tweets

To analyze the accuracy of the proposed method, we run the Event Detection tasks on the entire dataset. For these experiments, we use the CT text preprocessing strategy, discussed in subsection IV-A. After this step is completed, we determine the subsets of tweets belonging to each event, on which we apply the SFE text preprocessing strategy. For each subset of processed tweets, we use the proposed SA algorithms and compute the evaluation scores. The results are presented in Table VIII. The overall best scores for Sentiment Analysis of events are obtained when combining MABED with LR. Although, there is only a small gap between the resulted evaluation scores.

Table VIII: Average scores for Sentiment Analysis of Events

ED	SA	AUC	Accuracy	Precision	Recall
MABED	LR	0.768	0.793	0.570	0.720
	SVM	0.743	0.781	0.545	0.666
OLDA	LR	0.764	0.794	0.661	0.803
	SVM	0.747	0.781	0.644	0.777

To better understand this gap between the scores, we analyze the results individually. Thus, for each event detected in the previous experiment, we extract the topics and labeled them with the most recurring sentiment. We also extract the most recurring sentiment detected by the SA algorithms for each event. Tables IX and X present a subset of the sentiment detected for each event. The analysis of the results shows that our method (Algorithm 1) is accurate and is correctly identifying the sentiment for each event.

V. Conclusions

In this paper, we present a new approach that combines Event Detection and Sentiment Analysis for extracting the sentiments of events that appear in Social Networks. By combining Network Analysis with Machine Learning, our solution brings together two otherwise isolated communities, i.e., Network Analysis and Natural Language Processing.

Event Detection algorithms manage to extract different bursty topics. The experiment results show that OLDA increases topic readability and coherence, while MABED detects a wider range of topics.

The evaluation of Sentiment Analysis algorithms provides better inside in the construction of the model. Although, with the increase of the dataset size the gap between the evaluation scores obtained by algorithms increases, this increase does not prove to be a real gain when taking into account the runtime.

The method we proposed for extracting the sentiments of the social events shows interesting results. The overall best results are obtained when combining MABED with LR, although the increase in the evaluation scores is minimal in comparison with the other tested approaches. Furthermore, the experiments prove that our method manages to correctly determine the overall sentiment for an event.

Table IX: Sentiment Analysis of Events detect with MABED

MABED Topic	True Label	SVM Prediction	LR Prediction
nt sleep wa na morning gon early class getting school	NEGATIVE	NEGATIVE	NEGATIVE
movie look read wa song watching getting	POSITIVE	POSITIVE	POSITIVE
phone wa miss ah finally quot doe lt facebook amp	NEGATIVE	NEGATIVE	NEGATIVE
nt night feel show wa hour fun doe	NEGATIVE	NEGATIVE	NEGATIVE
course saw party quot lt dude look em hey	POSITIVE	POSITIVE	POSITIVE

Table X: Sentiment Analysis of Events detect with OLDA

Topic OLDA	True Label	SVM Prediction	LR Prediction
work sleep hour tired done early get go still woke	NEGATIVE	NEGATIVE	NEGATIVE
sorry movie hear watch goodnight exam season mileycyrus love watched	NEGATIVE	NEGATIVE	NEGATIVE
twitter friend best leaving wont bye facebook know lol let	POSITIVE	POSITIVE	POSITIVE
fun bit show tonight much little bbq going passed wine	POSITIVE	POSITIVE	POSITIVE
happy looking birthday party forward bday http today great american	POSITIVE	POSITIVE	POSITIVE

As future work, we aim to improve accuracy by using Deep Learning methods for Sentiment Analysis. We also want to implement unsupervised methods for detecting the sentiment of the event. Furthermore, we plan to use better feature engineering and weighting schemes for the text preprocessing strategies.

ACKNOWLEDGEMENT

This research was funded by grant No. PN-III-P1-1.2-PCCDI-2017-0734.

REFERENCES

[1] E. Adeli, X. Li, D. Kwon, Y. Zhang, and K. Pohl, "Logistic regression confined by cardinality-constrained sample and feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[2] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *IEEE International Conference on Data Mining*. IEEE, 2008, pp. 3–12.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[4] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *International Workshop on Multimedia Wata Mining*. ACM, 2010, p. 4.

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[6] Q. Diao, J. Jiang, F. Zhu, and E.-P. Lim, "Finding bursty topics from microblogs," in *Association for Computational Linguistic*, 2012, pp. 536–544.

[7] R. S. D.W. Hosmer Jr., S. Lemeshow, *Applied Logistic Regression, 3rd Edition*. John Wiley & Sons, 2013.

[8] O. Erdem, E. Ceyhan, and Y. Varli, "A new correlation coefficient for bivariate time-series data," *Physica A: Statistical Mechanics and its Applications*, vol. 414, pp. 274–284, 2014.

[9] A. Goswami and A. Kumar, *Event Detection Using Twitter Platform*. Springer International Publishing, 2019, pp. 429–480.

[10] A. Guille and C. Favre, "Mention-anomaly-based event detection and tracking in twitter," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2014, pp. 375–382.

[11] A. Guille, C. Favre, H. Hacid, and D. A. Zighed, "Sondy: An open source platform for social dynamics mining and analysis," in *ACM SIGMOD International Conference on Management of Data*, 2013, pp. 1005–1008.

[12] A. Guille and H. Hacid, "A predictive model for the temporal dynamics of information diffusion in online social networks," in *International Conference on World Wide Web*. ACM, 2012, pp. 1145–1152.

[13] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM Sigmod Record*, vol. 42, no. 2, pp. 17–28, 2013.

[14] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *AAAI International Conference on Weblogs and Social Media*, 2011.

[15] R. Lu and Q. Yang, "Trend analysis of news topics on twitter," *International Journal of Machine Learning and Computing*, vol. 2, no. 3, p. 327, 2012.

[16] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[17] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2012, vol. 821.

[18] A. Pak and P. Paroubek, "Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2010, pp. 436–439.

[19] D. A. Shamma, L. Kennedy, and E. F. Churchill, "Peaks and persistence: Modeling the shape of microblog conversations," in *ACM Conference on Computer Supported Cooperative Work*, 2011, pp. 355–358.

[20] D. Sirbu, A. Secui, M. Dascalu, S. A. Crossley, S. Ruseti, and S. Trausan-Matu, "Extracting gamers' opinions from reviews," in *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2016, pp. 227–232.

[21] T. Takahashi, R. Tomioka, and K. Yamanishi, "Discovering emerging topics in social streams via link anomaly detection," in *IEEE International Conference on Data Mining*. IEEE, 2011, pp. 1230–1235.

[22] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 406–418, 2011.

[23] J. Weng and B.-S. Lee, "Event detection in twitter," in *AAAI International Conference on Weblogs and Social Media*, 2011.