

A Graph Neural Network For Assessing The Affective Coherence Of Twitter Graphs

Georgios Drakopoulos
Ionian University
Dept. of Informatics
c16drak@ionio.gr

Ioanna Giannoukou
University of Patras
School of Mgt. Science and Tech.
igian@upatras.gr

Phivos Mylonas
Ionian University
Dept. of Informatics
fmylonas@ionio.gr

Spyros Sioutas
University of Patras
CEID
sioutas@ceid.upatras.gr

Abstract—Graph neural networks (GNNs) is an emerging class of iterative connectionist models taking full advantage of the interaction patterns in an underlying domain. Depending on their configuration GNNs aggregate local state information to obtain robust estimates of global properties. Since graphs inherently represent high dimensional data, GNNs can effectively perform dimensionality reduction for certain aggregator selections. One such task is assigning sentiment polarity labels to the vertices of a large social network based on local ground truth state vectors containing structural, functional, and affective attributes. Emotions have been long identified as key factors in the overall social network resiliency and determining such labels robustly would be a major indicator of it. As a concrete example, the proposed methodology has been applied to two benchmark graphs obtained from political Twitter with topic sampling regarding the Greek 1821 Independence Revolution and the US 2020 Presidential Elections. Based on the results recommendations for researchers and practitioners are offered.

Index Terms—graph neural networks, social graph resiliency, affective coherency, linked data, high dimensional data

1. Introduction

In the connected era social networks arguably play an instrumental role in reflecting and even shaping up to an extent the public sentiment regarding various social, political, cultural or even historical events [1]. Twitter encourages short replies and high interaction between its accounts and has been steadily playing a central role in elections worldwide such as the 2016 US Republican (or GOP) party elections and later that year the US presidential elections [2] [3].

Large scale graph resiliency can take many forms depending mainly on the application and the information available. For instance, for graph signal processing (GSP) graph topology correlation [4] or a long short term memory (LSTM) network [5] may be suitable, whereas for long supply chains the knowledge of flows and capacities in light of the *max-flow-min-cut* theorem [6] may be appropriate. For social networks the overall sentiment polarity has been proposed as an important coherence factor [7] [8]. To this end affective models like the *emotion wheel* [9] or the *universal emotion theory* [10] have been progressively adapted

to describe individual online behavior in various contexts. This appears to be especially true in Twitter where among others irony, inflammatory comments, apophegmatic retorts, and bots with affective programming have been reported to elicit different emotional responses across accounts [11].

Graph neural networks (GNNs) are connectionist models which can efficiently learn global properties by iteratively updating local state vectors assigned to vertices or edges. Compared to other neural network (NN) architectures they tend to exhibit scalability and robust generalization of local states. Moreover, existing affective coherency models for large graphs either lack scalability [12] or do not properly consider the linked nature of social graphs [13]. This is the twofold principal motivation behind this work.

The primary research contribution of this conference paper is an iterative GNN architecture locally aggregating vertex state vectors comprising of structural, functional, and affective attributes in order to obtain vertex labels pertaining to the respective emotional polarity. The latter has been identified as a key factor for the overall social network resiliency. Two large benchmark graphs obtained from the political Twitter serve as an illustration point, whereas their structural and functional properties are extensively studied. As a secondary objective, based on the results recommendations are given for researchers and practitioners.

The remainder of this work is structured as follows. In section 2 the recent scientific literature regarding graph resiliency metrics and neural network architectures is briefly overviewed. The proposed methodology is explained in detail in section 3. In section 4 the experimental setup is described and the results of the benchmark social graphs along with practical recommendations based on the experimental findings are given. Section 5 concludes this work by recapitulating the main results and outlining possible future research directions. Random variables are represented by capital calligraphic, matrices by capital boldface, and vectors by small boldface letters. Technical acronyms are explained the first time they are encountered in the text. Finally, the notation of this work is summarized in table 1.

2. Previous Work

GNNs are a class of iterative connectionist models which capture the dependencies of an underlying domain and

TABLE 1. NOTATION OF THIS CONFERENCE PAPER.

Symbol	Meaning	First in
\triangleq	Equality by definition	Eq. (1)
$ S $	Set or tuple cardinality	Eq. (9)
$\deg(v)$	Degree of vertex v	Eq. (9)
$E[\mathcal{X}]$	Expected value of r.v. \mathcal{X}	Eq. (13)
$\text{Var}[\mathcal{X}]$	Variance of r.v. \mathcal{X}	Eq. (14)
$\mu[\mathcal{X}]$	Skewness of r.v. \mathcal{X}	Eq. (16)
$\kappa[\mathcal{X}]$	Kurtosis of r.v. \mathcal{X}	Eq. (17)
$\det(\mathbf{M})$	Determinant of matrix \mathbf{M}	Eq. (24)
\mathbf{I}_n	$n \times n$ identity matrix	Eq. (28)
\mathbf{O}_n	$n \times n$ zero matrix	Eq. (30)
$\text{sgn}(x)$	Sign of scalar x	Algo. (1)

construct global property estimators from individual ground truth state vectors. Higher order GNNs are introduced in [14], capsule GNNs have been proposed in [15], and few-shot learning scenarios in [16]. Applications include social recommendation [17] and graph contrasting coding for learning graph representations [18]. Finally, two recent comprehensive reviews on the field are [19] and [20].

Other recent architectures include recurrent neural networks (RNNs) [21]. The primary characteristic is that connections in RNNs form temporal sequences which also serve as mineable patterns [22]. Applications include affective document classification [23], human activity recognition [24], and geometric matrix completion [25].

Convolutional neural networks (CNNs) rely heavily on local weighted and delayed feedback [26]. Predicting the physical trajectory of social agents, whether humans or autonomous vehicles, based on local smoothness constraints [27] and sound classification based on attributes extracted by signal processing techniques [28] are two applications.

Tensor stack networks (TSNs) consist of clusters of feedforward neural networks (FFNNs) where each network in the cluster receives a synaptic weight update based not only on its own local error gradient but also on that of other networks in the cluster [29]. As their name suggests, tensors are their basic building blocks [30]. TSNs have been proposed for graph resiliency [31], gesture interpretation [32], and face recognition [33]. For a TSN toolkit see [34].

Self organizing maps (SOMs) constitute a class of neurons arranged in a low dimensional grid and trained in an unsupervised manner based on a modified Hebbian rule [35]. SOMs are designed to learn the topology of an underlying higher dimensional manifold [36]. Applications include clustering large databases [37], learning functional magnetic resonance images (fMRI) [38], interpreting gene expressions [39], and denoising industrial images [40].

3. Proposed Method

3.1. Graphs As High Dimensional Data

High dimensional data are frequently generated across a broad spectrum of fields ranging from digital medicine and smart city management to finance, e-commerce, space exploration, and social media. Data dimensionality as well as

variability, including missing fields, depends heavily on the underlying field. Nonetheless, the fundamental goal remains essentially the same, namely the description of time-varying entities from different perspectives and possible at various levels of granularity as deemed appropriate.

Social networks are no exception to this rule as they model entities, either individuals or organizations, at Internet scale. Additionally, social graphs, and any type of graphs for that matter, can inherently represent high dimensional data in a recursive and distributed way. Indeed, local information can be augmented with that of connected entities or with information obtained from the local digital communities in order to yield a better sketch of the entity under consideration. Additionally, information from entities can be combined in a bottom-up manner in order to construct a description of the community comprising of these entities.

3.2. Graph Resiliency Evaluation

GNNs start only with knowledge of a locally valid ground truth expressed as a state vector of finite length. In the most common GNN variants such vectors are only assigned to vertices, as is the case here. Then GNNs move gradually from this initial information to a robust global property estimators through an iterative process, implying that a termination criterion must be carefully designed to understand when a steady state is reached. These estimators eventually depend not only on the state vectors but on the local connections as well. Concerning the former, the proposed scheme combines the following categories of features:

- **Structural:** Nearly all interaction including communication and replies in a conversation occur over a social graph. The latter is maintained by explicit *follow* relationships. Since they are one-way, their direction is an important information offering higher granularity compared to two-way relationships like those in Facebook. Although they do not carry significant affective potential, they can denote strong connection or lack thereof between two accounts.
- **Functional:** These attributes pertain to the online activity itself. However, they are indirectly or even remotely related to the affective status of the respective account. Thus they are also considered neutral in terms of affective potential.
- **Affective:** The affective interaction in Twitter and its affective aspects are captured by these attributes. Typically both their sign and magnitude are important to fully understand the emotional polarity of the respective account. In most cases they are easy to determine as most accounts, especially in Twitter, are quite expressive –and occasionally vociferous.

The state vector \mathbf{s}_k associated with each vertex $v_k \in V$ has the structure shown in equation (1). It comprises of attributes for each of the above categories. Specifically, the first two components are structural ($r_{1,k}$ and $r_{2,k}$), followed by five functional ($f_{1,k}$ to $f_{5,k}$), and three more affective ones ($a_{1,k}$ to $a_{3,k}$). Although in the general case state vector

components may well be signed, here \mathbf{s}_k will contain only non-negative ones. Attributes have been selected based on recent scientific literature recommendations [8] [2] [11] and their meaning is explained in table 2.

$$\mathbf{s}_k \triangleq [r_{1,k} \quad r_{2,k} \quad f_{1,k} \dots f_{5,k} \quad a_{1,k} \dots a_{3,k}]^T \quad (1)$$

The proposed GNN is seeks to find a label \hat{l}_k for each v_k indicating its emotional polarity, which is an estimator of the inaccessible true polarity label l_k . To achieve this computational goal, not only the local information stored in a vertex will be exploited, but also the particular connectivity patterns. In turn this allows the GNN to take full advantage of neighboring state vectors. Moreover, the connection itself is treated as additional latent information in the aggregator.

TABLE 2. VERTEX STATE ATTRIBUTES.

Attribute	Meaning
$r_{1,k}$	Number of <i>follow</i> relationships
$r_{2,k}$	Number of inverse <i>follow</i> relationships
$f_{1,k}$	Number of tweets
$f_{2,k}$	Number of retweets
$f_{3,k}$	Number of hashtags in tweets and retweets
$f_{4,k}$	Number of mentions
$f_{5,k}$	Number of replies
$a_{1,k}$	Number of positive hashtags
$a_{2,k}$	Number of negative hashtags
$a_{3,k}$	Number of neutral hashtags

The original label $l_k^{[0]}$ for each v_k is set to the appropriate value as shown in equation (2). The three individual hashtag fractions are determined by the natural language processing (NLP) procedure and of algorithm 1, hence, these starting labels will be referred to as the NLP labels. Observe that these labels are first order estimates.

$$l_k^{[0]} \triangleq \begin{cases} \frac{a_{1,k}}{a_{1,k} + a_{2,k}} & a_{1,k} > a_{2,k} \\ -\frac{a_{2,k}}{a_{1,k} + a_{2,k}} & a_{2,k} > a_{1,k} \\ 0 & a_{1,k} = a_{2,k} \end{cases} \in [-1, 1] \triangleq I_0 \quad (2)$$

Computing the affective sign of a hashtag h_0 requires determining the sign of the difference between the positive and the negative words of it as shown in algorithm 1.

Algorithm 1 Assigning affective polarity to hashtags

Require: Hashtag h_0 coded as a case-sensitive string

Ensure: Sentiment polarity of h_0 is computed

- 1: set counter c_0 to zero
 - 2: **for all** words w of h_0 **do**
 - 3: **if** w is positive according to SentimentAnalysis **then**
 - 4: add one to c_0
 - 5: **else if** w is negative **then**
 - 6: subtract one from c_0
 - 7: **end if**
 - 8: **end for**
 - 9: **return** $\text{sgn}(c_0)$
-

In the j -th iteration labels are updated as in (3):

$$l_k^{[j]} = \varphi \left(\frac{\beta_0}{2} l_k^{[j-1]} + \frac{\beta_0}{2} \sum_i \gamma_i \delta_{i,k} l_i^{[*]} \right) \quad (3)$$

The summation takes place over every neighboring vertex v_i of v_k , where the *follow* direction is ignored. The iteration index $*$ means that if a label has been updated, then the new label is used. The local label receives as much weight as that of the combined neighborhood effect. This prevents vertices, especially the influential ones, to be overtaken by a large neighborhood merely because of the size of the latter.

In equation (3) γ_i is the followers-to-all ratio of (4):

$$\gamma_i \triangleq \frac{r_{2,i}}{r_{1,i} + r_{2,i}} \quad (4)$$

Also $\delta_{i,k}$ is a coefficient denoting online influence as in (5):

$$\delta_{i,k} \triangleq \frac{1}{3} \left(\frac{f_{1,i}}{f_{1,i} + f_{2,i}} + \frac{f_{4,i}}{f_{4,i} + f_{5,i}} + \frac{f_{3,i}}{f_{3,i} + f_{3,k}} \right) \quad (5)$$

The first fraction denotes the fraction of the original content v_i generates by tweeting to the total content it forwards to its connections by tweeting and retweeting. Along a similar line of reasoning, the second fraction measures how many replies are specifically directed to v_i in contrast to casual mentions. The third fraction connects v_i with v_k through pairwise hashtag coherency regardless of affective polarity.

The function $\varphi(\cdot)$ is the hyperbolic tangent of equation (6). It is strongly non-linear, smooth, and maps \mathbb{R} to I_0 .

$$\varphi(\vartheta; \beta_0) \triangleq \tanh(\vartheta; \beta_0) = \frac{e^{\beta_0 \vartheta} - e^{-\beta_0 \vartheta}}{e^{\beta_0 \vartheta} + e^{-\beta_0 \vartheta}} \quad (6)$$

Notice that scaling by β_0 is necessary in order to transform I_0 to a domain where $\varphi(\cdot)$ practically takes all values in I_0 . For the purposes of this work $\beta_0 = 5$ has been selected.

Since the proposed GNN works iteratively, the convergence conditions must be set. The first derivative of the hyperbolic tangent satisfies the condition of equation (7), which places a growth rate cap close to the bounds of I_0 .

$$\dot{\varphi}(\vartheta; \beta_0) = \beta_0 (1 - \varphi^2(\vartheta; \beta_0)) \quad (7)$$

Also, $\varphi(\cdot)$ satisfies the Brouwer fixed point theorem. The latter states that if a mapping $f: I \rightarrow I$ is continuous and the interval I is closed there is one $x^* \in I$ such that:

$$x^* = f(x^*) \quad (8)$$

The relative steady state difference will track convergence. It is the number of labels which have changed more than a threshold η_0 between two iterations to the number of labels. When it is under a threshold η_1 , the iteration terminates.

Algorithm 2 summarizes the operation of the GNN.

Algorithm 2 The proposed GNN

Require: State vectors as in equation (1)

Ensure: Robust affective polarity labels $\{l_k\}$

- 1: obtain initial labels $\{l_k^{[0]}\}$ from algorithm 1
 - 2: **repeat**
 - 3: **for all** vertices v_k **do**
 - 4: update local label l_k as in equation (3)
 - 5: **end for**
 - 6: **until** the relative steady state difference drops under η_1
 - 7: **return** labels $\{l_k\}$
-

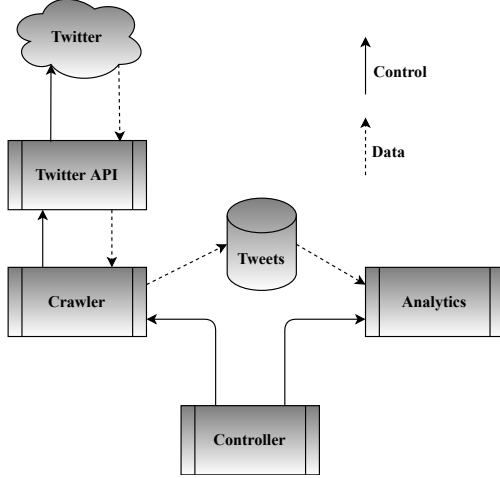


Figure 1. Proposed system architecture.

4. Results

4.1. Experimental Setup

This section describes the infrastructure used to obtain and process the two benchmark graphs. Figure 1 depicts the components well as the various interactions between them.

The experimental setup is shown in table 3. It contains any parameters which do not directly influence the execution of the GNN itself as previously described but still they play a central role in the results of the experiments.

TABLE 3. EXPERIMENTAL SETUP.

Parameter	Value
Label change threshold η_0	0.05
Relative steady state threshold η_1	0.05
Number of runs R_0	1000
Data subset starting size	10% with increments of 10%
Subset selection policy	Random

In order to verify the Twitter application the two OAuth developer keys were obtained for the account running the application and two more for the application itself.

Although algorithmic bias in social media searches is not the focus of this work, the best practice guidelines set forth in [41] and in [42] were integrated into the crawler.

4.2. 1821 Graph

2021 marks an important milestone from the Greek Revolution of Independence of 1821. Although many celebrations are expected to be canceled because of the still raging COVID-19 pandemic, Twitter activity about the events still schedules as well as about the anniversary itself has soared.

The synopsis for the 1821 graph is given in table 4. It shows from top to bottom basic structural, advanced structural, and functional properties. Among structural properties higher order ones are of special interest since they are activity indicators among account groups. Specifically [43]:

- Triangles are of order three for vertices and edges. They constitute the smallest possible communities.
- Squares are of order four for vertices and edges. They are loose communities of size four.
- Four-cliques are of order four for vertices and six for edges. They are tight communities of size four.

TABLE 4. 1821 GRAPH SYNOPSIS.

Property	Value
Number of vertices $ V_h $	132.317
Number of edges $ E_h $	2.225.177
Density ρ_h / Log-density ρ'_h	16.8170 / 1.2393
Completeness ξ_h / Log-completeness ξ'_h	$2.54E^{-4}$ / 0.6196
Number of triangles T_h	446.513
Number of squares S_h	215.387
Number of cliques of size four C_h	102.044
Graph diameter L_h	10
Percentage of vertices reachable at $L_h - 1$	95.33%
Percentage of vertices reachable at $L_h - 2$	93.26%
Percentage of vertices reachable at $L_h - 3$	89.11%
Percentage of vertices reachable at $L_h - 4$	84.73%
Number of favorites F_h	36.994.815
Number of tweets W_h	17.465.844
Number of buckets B_h	$\lceil W_h \rceil$
Sampling interval	8/2020-10/2020

In table 4 the graph density ρ_h is defined as the ratio of the number of the edges to the number of vertices. Because of the Euler graph formula, density is also an approximation of the average graph degree as shown in equation (9):

$$\rho_h \triangleq \frac{|E_h|}{|V_h|} = \frac{2}{|V_h|} \sum_{v_k \in V_h} \deg(v_k) \quad (9)$$

The log-density ρ'_h is similarly defined as the ratio of the respective order of magnitudes of the number of edges to the number of vertices as shown in equation (10):

$$\rho'_h \triangleq \frac{\ln |E_h|}{\ln |V_h|} \quad (10)$$

The completeness ξ_h of a graph is defined as the ratio of the edges of a graph to the edges of a fully connected graph with the same number of vertices as shown in equation (11):

$$\xi_h \triangleq \frac{|E_h|}{\binom{|V_h|}{2}} = \frac{2|E_h|}{|V_h|(|V_h| - 1)} \approx \frac{2\rho_h}{|V_h|} \quad (11)$$

Also the log-completeness ξ'_h is defined as the order of magnitude of the edges of a graph to that of the edges of a fully connected graph with the same number of edges (12):

$$\xi'_h \triangleq \frac{\ln |E_h|}{\ln \binom{|V_h|}{2}} = \frac{\ln |E|}{\ln |V_h| + \ln (|V_h| - 1) - \ln 2} \approx \frac{\rho'_h}{2} \quad (12)$$

Based on the entries of table 4 the 1821 benchmark graph is simultaneously very compact and quite active. Also it abounds with hashtags, which have an elevated semantic importance compared to ordinary tweet terms [44].

The hashtag frequency scree plot is shown in figure 2.

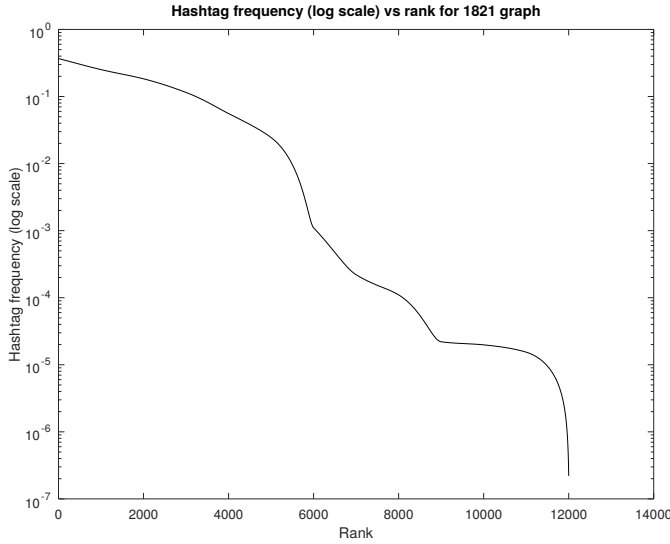


Figure 2. Scree plot for the 1821 graph.

The expected value of the number of hashtags per tweet I_h as counted by the r.v. \mathcal{H} is computed as in equation (13):

$$I_h \triangleq E[\mathcal{H}] \approx \frac{1}{W_h} \sum_{k=1}^{W_h} h_k \quad (13)$$

In the above equation W_h is the total number of tweets in the graph and h_k is the number of the hashtags of the k -th such tweet. Notice that the rightmost side of equation (13) is the sample mean, namely an estimator in the strict signal processing sense of the true stochastic mean of \mathcal{H} . However, since the actual probability theoretic properties of \mathcal{H} are unknown, the sample mean will be used as under mild assumptions it converges in the mean to $E[\mathcal{H}]$.

The variance σ_h^2 of $\text{Var}[\mathcal{H}]$ is also approximated by the sample variance or the natural estimator as shown in (14):

$$\sigma_h^2 \triangleq \text{Var}[\mathcal{H}] \approx \sqrt{\frac{1}{W_h - 1} \sum_{k=1}^{W_h} (h_k - I_h)^2} \quad (14)$$

Once I_h and σ_h^2 are obtained the first question is whether hashtag frequency distribution is Gaussian as in (15):

$$f_{\mathcal{H}}(h) \triangleq \frac{1}{\sigma_h \sqrt{2\pi}} \exp\left(-\frac{(h - I_h)^2}{2\sigma_h^2}\right) \quad (15)$$

This distribution is appealing for the following reasons [45]:

- It has the maximum differential entropy among the distributions of the same variance. Thus, it explains many probabilistic scenaria under this constraint.
- Because of the central limit theorem (CLT), the Gaussian distribution can effectively model the behavior of a large sum of independent r.v.s.

Table 5 presents the results from the four normality tests of [46], which are all negative. A possible explanation for this could be that the hashtags are not chosen independently by Twitter accounts but instead there is high dependency based on hashtag popularity, whether they belong to trending topics, and replies and mentions from other accounts [47]. External events tend also to be treated similarly by sociopolitically aligned accounts and, therefore, there is a significant hashtag overlap [48]. Moreover, partly because of the Zipf law of minimum effort [49] or the homophily phenomeon encountered in social networks [50], accounts tend to copy the hashtags of aligned opinion leaders [51].

TABLE 5. GAUSSIANITY RESULTS FOR TEST GRAPHS.

Test name	1821	US2020
Kolmogorov-Smirnov	No	No
Lilliefors	No	No
Anderson-Darling	No	No
Shapiro-Wilk	No	No

Given the negative Gaussianity results of table 5, other ways to describe \mathcal{H} had to be sought. A possible option lies in the direction of analysis carried out in [43] for understanding the performance of two graph searches for discovering trusted candidates for startups in LinkedIn.

First, the skewness $\mu[\mathcal{H}]$ of \mathcal{H} is computed as in (16):

$$\mu[\mathcal{H}] \triangleq E\left[\left(\frac{\mathcal{H} - E[\mathcal{H}]}{\sqrt{\text{Var}[\mathcal{H}]}}\right)^3\right] \quad (16)$$

For unimodal distributions with finite variance the skewness sign has the following interpretation:

- When skewness is zero, the distribution is balanced.
- When it is positive, then the right tail is heavier.
- When it is negative, then the left tail is heavier.

Additionally, the kurtosis of \mathcal{H} is computed as in (17):

$$\kappa[\mathcal{H}] \triangleq E\left[\left(\frac{\mathcal{H} - E[\mathcal{H}]}{\sqrt{\text{Var}[\mathcal{H}]}}\right)^4\right] = \frac{E[\mathcal{H} - E[\mathcal{H}]]^4}{\text{Var}[\mathcal{H}]^2} \quad (17)$$

Kurtosis is also related to the mass of the tails of a distribution. It is an indicator of the frequency of outlier generation in comparison to the Gaussian distribution.

The results for both benchmark graphs are shown in table 6. The column labeled Uni. indicates whether the corresponding distribution is unimodal, which is instrumental for interpreting the values of skewness and kurtosis. The respective mean hashtag value for the US2020 graph is higher while both benchmark graphs have approximately the

same variance (σ_h and σ_g). This means that tweets from the US2020 graph are expected to have more hashtags on the average. Moreover, the US2020 graph is more skewed and is prone to generate more outliers compared to the 1821 graph. This could be an indication that the former has conversations with more hashtags and that many of them are infrequent.

TABLE 6. SKEWNESS AND KURTOSIS FOR BENCHMARK GRAPHS.

Graph	Uni.	I_h/I_g	σ_h/σ_g	$\mu[\mathcal{H}]$	$\kappa[\mathcal{H}]$
1821	Yes	131.3333	821.8511	3.6518	3.5542
US2020	Yes	154.5000	813.3334	3.8994	3.9817

Another way to characterize the behavior of \mathcal{H} is to construct the scree distribution, namely the frequency of its values vs their respective rank. Along a similar line of reasoning, the log-scrree distribution is the logarithm of the frequency, which is non-zero by definition, vs the respective rank. For the purposes of this work the log-scrree plot will be used. For the 1821 and the the US 2020 Elections graphs the respective log-scrree plots are shown in figures 2 and 3.

A well-known frequency rank model which can explain the linear decay rate at the initial steps of the two scree plots is the power law examined among others in [52]:

$$f_k \triangleq \alpha_0 k^{-\gamma_0}, \quad 1 \leq k \leq B_h \quad (18)$$

In the above equation the positive exponent γ_0 plays a central role, similar to the Lyapunov exponent for energy functions in control theory, as it determines the type of the power law distribution. Specifically:

- When $\gamma_0 \geq 3$, the mean value and variance are finite.
- When $2 \leq \gamma_0 < 3$, only the mean value is finite.
- Otherwise the mean value and variance are infinite.

Moreover, B_h denotes the number of buckets used to cluster the values of \mathcal{H} for the respective frequency ranking. For the purposes of this work B_h has been selected to be:

$$B_h \triangleq \left\lceil \sqrt{W_h} \right\rceil \quad (19)$$

The above selection ensures that each bucket contains a statistically sufficient number of tweet counts h_k [46].

The exponent γ_0 can be estimated by taking the logarithm of both sides of (18) linearizing thus the power law:

$$\ln f_k = \ln \alpha_0 - \gamma_0 \ln k, \quad 1 \leq k \leq B_h \quad (20)$$

Stacking the B_h equations of (20) yields the system:

$$\underbrace{\begin{bmatrix} \ln f_1 \\ \ln f_2 \\ \vdots \\ \ln f_{B_h} \end{bmatrix}}_{\mathbf{f}_p} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & -\ln 2 \\ \vdots & \vdots \\ 1 & -\ln B_h \end{bmatrix}}_{\mathbf{X}_p} \underbrace{\begin{bmatrix} \ln \alpha_0 \\ \gamma_0 \end{bmatrix}}_{\mathbf{b}_p} \quad (21)$$

The least squares (LS) solution for (21) is given by:

$$\hat{\mathbf{b}}_p \triangleq (\mathbf{X}_p^T \mathbf{X}_p)^{-1} \mathbf{X}_p^T \mathbf{f}_p \quad (22)$$

In the general case in equation (22) the square matrix is never inverted as it is a costly and potentially numerically unstable operation. In this case the structure of $\mathbf{X}_p^T \mathbf{X}_p$ is:

$$\mathbf{X}_p^T \mathbf{X}_p = \begin{bmatrix} B_h & -\sum_{k=2}^{B_h} \ln k \\ -\sum_{k=2}^{B_h} \ln k & \sum_{k=2}^{B_h} \ln^2 k \end{bmatrix} \quad (23)$$

Notice that by construction $\mathbf{X}_p^T \mathbf{X}_p$ it is symmetric.

Since the resulting LS system is 2×2 , it suffices to compute two determinants. The determinant of $\mathbf{X}_p^T \mathbf{X}_p$ which determines whether the linear system is undefined is:

$$\det(\mathbf{X}_p^T \mathbf{X}_p) = B_h \sum_{k=2}^{B_h} \ln^2 k - \left(\sum_{k=2}^{B_h} \ln k \right)^2 \quad (24)$$

Matrix \mathbf{Y}_p is defined as:

$$\mathbf{Y}_p \triangleq \begin{bmatrix} B_h & \sum_{k=1}^{B_h} \ln f_k \\ -\sum_{k=2}^{B_h} \ln k & -\sum_{k=2}^{B_h} \ln k \ln f_k \end{bmatrix} \quad (25)$$

Observe that in equation (25) the matrix \mathbf{Y}_p is derived from \mathbf{X}_p by substituting its second column, namely the column corresponding to the second parameter γ_0 , with vector $\mathbf{X}_p^T \mathbf{f}_p$. The determinant of \mathbf{Y}_p is given in (26):

$$\det(\mathbf{Y}_p) = \left(\sum_{k=2}^{B_h} \ln k \right) \left(\sum_{k=1}^{B_h} \ln f_k \right) - B_h \left(\sum_{k=2}^{B_h} \ln k \ln f_k \right) \quad (26)$$

Using determinant properties, the exponent γ_0 is:

$$\gamma_0 \triangleq \frac{\det(\mathbf{Y}_p)}{\det(\mathbf{X}_p^T \mathbf{X}_p)} \quad (27)$$

A similar result holds for the first parameter $\ln \alpha_0$.

The mean square error (MSE) J_p between the actual and the projected frequencies is defined as in equation (28):

$$J_p \triangleq \frac{1}{B_h} \left\| \mathbf{f}_p - \mathbf{X}_p \hat{\mathbf{b}}_p \right\|_2^2 = \frac{1}{B_h} \left\| (\mathbf{I}_{B_h} - \mathbf{H}_p) \mathbf{f}_p \right\|_2^2 \quad (28)$$

In equation (28) the projection matrix \mathbf{H}_p is defined as:

$$\mathbf{H}_p \triangleq \mathbf{X}_p (\mathbf{X}_p^T \mathbf{X}_p)^{-1} \mathbf{X}_p^T \in \mathbb{R}^{B_h \times B_h} \quad (29)$$

Notice that \mathbf{H}_p is a $B_h \times B_h$ orthogonal projection operator and so is $\mathbf{I}_{B_h} - \mathbf{H}_p$ but to a perpendicular space. To see that observe that two successive projections, one with $\mathbf{I}_{B_h} - \mathbf{H}_p$ and one with \mathbf{H}_p , always results in the zero matrix as shown in (30). The same holds if projections are swapped.

$$(\mathbf{I}_{B_h} - \mathbf{H}_p) \mathbf{H}_p = \mathbf{H}_p - \mathbf{H}_p^2 = \mathbf{H}_p - \mathbf{H}_p = \mathbf{O}_{B_h} \quad (30)$$

In light of this, in equation (28) the MSE J_p is essentially the part of the measurements which cannot be projected on the subspace spanned by the columns of \mathbf{H}_p .

Another frequency rank model explaining both the initial almost linear decay and the subsequent sharp cutoff for higher ranks of figures 2 and 3 is that of equation (18) [53]:

$$f_k \triangleq \alpha_0 k^{-\gamma_0} \exp\left(-\frac{k}{\gamma_1}\right), \quad 1 \leq k \leq B_h \quad (31)$$

Taking the natural logarithm of both sides of (31) yields:

$$\ln f_k = \ln \alpha_0 - \gamma_0 \ln k - \frac{k}{\gamma_1}, \quad 1 \leq k \leq B_h \quad (32)$$

Stacking the equations now results in the system (33):

$$\underbrace{\begin{bmatrix} \ln f_1 \\ \ln f_2 \\ \vdots \\ \ln f_{B_h} \end{bmatrix}}_{\mathbf{f}_c} = \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 1 & -\ln 2 & -2 \\ \vdots & \vdots & \vdots \\ 1 & -\ln B_h & -B_h \end{bmatrix}}_{\mathbf{X}_c} \underbrace{\begin{bmatrix} \ln \alpha_0 \\ \gamma_0 \\ \frac{1}{\gamma_1} \end{bmatrix}}_{\mathbf{b}_c} \quad (33)$$

Along a similar line of reasoning the LS solution of the linear system (33) is that of equation (34):

$$\hat{\mathbf{b}}_c = (\mathbf{X}_c^T \mathbf{X}_c)^{-1} \mathbf{X}_c^T \mathbf{f}_c \quad (34)$$

Now the 3×3 coefficient matrix $\mathbf{X}_c^T \mathbf{X}_c$ of the linear system with the normal equations has the structure of (35):

$$\mathbf{X}_c^T \mathbf{X}_c \triangleq \begin{bmatrix} B_h & -\sum_{k=2}^{B_h} \ln k & -\sum_{k=1}^{B_h} k \\ -\sum_{k=2}^{B_h} \ln k & \sum_{k=2}^{B_h} \ln^2 k & \sum_{k=2}^{B_h} k \ln k \\ -\sum_{k=1}^{B_h} k & \sum_{k=2}^{B_h} k \ln k & \sum_{k=1}^{B_h} k^2 \end{bmatrix} \quad (35)$$

Matrix \mathbf{Y}_c^0 of equation (36) is derived from the coefficient matrix of equation (35) by substituting the second column with vector $\mathbf{X}_c^T \mathbf{f}_c$ as in (36):

$$\mathbf{Y}_c^0 \triangleq \begin{bmatrix} B_h & \sum_{k=1}^{B_h} \ln f_k & -\sum_{k=1}^{B_h} k \\ -\sum_{k=2}^{B_h} \ln k & -\sum_{k=2}^{B_h} \ln k \ln f_k & \sum_{k=2}^{B_h} k \ln k \\ -\sum_{k=1}^{B_h} k & -\sum_{k=2}^{B_h} k \ln f_k & \sum_{k=1}^{B_h} k^2 \end{bmatrix} \quad (36)$$

The LS estimation of the exponent γ_0 using determinant properties can be computed as in equation (37):

$$\gamma_0 = \frac{\det(\mathbf{Y}_c^0)}{\det(\mathbf{X}_c^T \mathbf{X}_c)} \quad (37)$$

Along a similar line of reasoning matrix \mathbf{Y}_c^1 is constructed by substituting the third column of matrix $\mathbf{X}_c^T \mathbf{X}_c$ with the vector $\mathbf{X}_c^T \mathbf{f}_c$ as shown in equation (38):

$$\mathbf{Y}_c^1 \triangleq \begin{bmatrix} B_h & -\sum_{k=2}^{B_h} \ln k & \sum_{k=1}^{B_h} \ln f_k \\ -\sum_{k=2}^{B_h} \ln k & \sum_{k=2}^{B_h} \ln^2 k & -\sum_{k=2}^{B_h} \ln k \ln f_k \\ -\sum_{k=1}^{B_h} k & \sum_{k=2}^{B_h} k \ln k & -\sum_{k=2}^{B_h} k \ln f_k \end{bmatrix} \quad (38)$$

The LS estimation of the time constant γ_1 which eventually controls the exponential cut-off is computed as in (39):

$$\gamma_1 = \frac{\det(\mathbf{X}_c^T \mathbf{X}_c)}{\det(\mathbf{Y}_c^1)} \quad (39)$$

Observe that in equation (39) the determinant of matrix $\mathbf{X}_c^T \mathbf{X}_c$ is on the nominator as the system of equation (33) is designed to compute the reciprocal of γ_1 .

The determinant of 3×3 matrices can be efficiently computed with the Sarrus rule [54], which is a special case of Leibniz formula, or with Laplace expansions.

Now the projection operator \mathbf{H}_c which spans the space of projected measurements is defined as in equation (40):

$$\mathbf{H}_c \triangleq \mathbf{X}_c (\mathbf{X}_c^T \mathbf{X}_c)^{-1} \mathbf{X}_c^T \quad (40)$$

The MSE J_c is now computed as in equation (41):

$$J_c \triangleq \frac{1}{B_h} \|\mathbf{f}_c - \mathbf{H}_c \mathbf{f}_c\|_2^2 = \frac{1}{B_h} \|(\mathbf{I}_{B_h} - \mathbf{H}_c) \mathbf{f}_c\|_2^2 \quad (41)$$

The LS solutions of models of equations (18) and (31) have been computed using the actual hashtag frequency measurements for both benchmark graphs as collected from topic sampling. The results are shown in table 7.

TABLE 7. RESULTS FOR THE LS SOLUTIONS.

Graph	Model	γ_0	γ_1	MSE
1821	Power law	2.6511	-	167.4500
1821	Cutoff	2.488	3545.66	143.3333
US2020	Power law	2.8617	-	170.5000
US2020	Cutoff	2.6322	3633.33	145.9811

The exponents γ_0 indicate a relatively graceful reduction. Moreover, the power law yields higher values as the LS fitting yields a straight line with a steeper curve. With the addition of the exponential fall, that line becomes less steep. In any case, the fractional exponents indicate fractal patterns in the distribution of hashtag frequencies such as self-similarity and heavy tail. The latter is confirmed from the skewness and kurtosis results. Concerning the γ_1 time constant, it has similar values for both graphs, which is approximately one quarter of B_h . The last is consistent with the fact that in figures 2 and 3 the cutoffs appear after four time constants. Also, for both benchmarks the cutoff model of (31) attains lower error from the real frequencies.

The models of equations (18) and (31) are not the only ones proposed for hashtag frequency distribution. Another model is the DGX [55] which has been proposed for the degree distribution of large scale free graphs shown in (42):

$$f_k \triangleq \frac{A(\mu_0, \sigma_0)}{k} \exp\left(-\frac{(\ln k - \mu_0)^2}{2\sigma_0^2}\right) \quad (42)$$

In equation (42) the normalization factor A equals:

$$A(\mu_0, \sigma) \triangleq \left(\sum_{k=1}^{+\infty} \frac{1}{k} \exp\left(-\frac{(\ln k - \mu_0)^2}{2\sigma_0^2}\right) \right)^{-1} \quad (43)$$

The DGX model combines the decay rate of the power law with the smoothness and location dependency of the Gaussian kernel. However, estimating the parameters μ_0 and σ_0^2 leads to a system of non-linear differential equations.

4.3. US 2020 Elections Graph

The 2020 US Presidential Elections were conducted in a rather unique political climate with Twitter playing a central role in the respective campaigns of both parties.

The synopsis for the US 2020 Elections graph is given in table 8, which has the same format with table 4. Observe that both benchmark graphs have similar properties.

TABLE 8. US 2020 ELECTIONS GRAPH SYNOPSIS.

Property	Value
Number of vertices V_g	147.881
Number of edges E_g	2.447.224
Density ρ_g / Log-density ρ'_g	16.5486 / 1.2357
Completeness ξ_g / Log-completeness ξ'_g	$2.38E^{-4}$ / 0.6173
Number of triangles T_g	489.773
Number of squares S_g	218.633
Number of cliques of size four C_g	125.806
Graph diameter L_g	11
Percentage of vertices reachable at $L_g - 1$	98.17%
Percentage of vertices reachable at $L_g - 2$	96.44%
Percentage of vertices reachable at $L_g - 3$	91.22%
Percentage of vertices reachable at $L_g - 4$	87.47%
Number of favorites F_g	42.114.509
Number of tweets W_g	22.773.674
Number of buckets B_g	$\lceil \sqrt{W_g} \rceil \approx B_h$
Sampling interval	8/2020-10/2020

The hashtag frequency scree plot for the US Elections 2020 is shown in figure 3. Observe that it is relatively smoother compared to that of the 1821 graph. This can be attributed to the somewhat richer variety of hashtags as well as to their more frequent use compared to the latter.

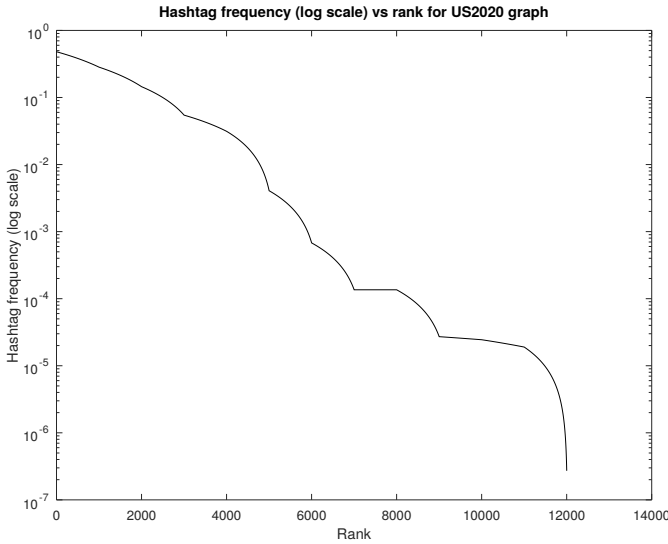


Figure 3. Scree plot for the US2020 graph.

4.4. Performance

Figure 4 shows the number of iterations for both benchmark graphs vs the subset size percentage. It appears that the US2020 requires systematically more iterations to converge. This may be an indication of convoluted interactions between accounts. In both cases scaling is linear for small subsets but this arguably changes for larger ones.

Figure 5 shows the GNN convergence rate for the two benchmark graphs as measured by the relative steady state difference. It can be seen that for the 1821 graph convergence is quicker and perhaps the process could have

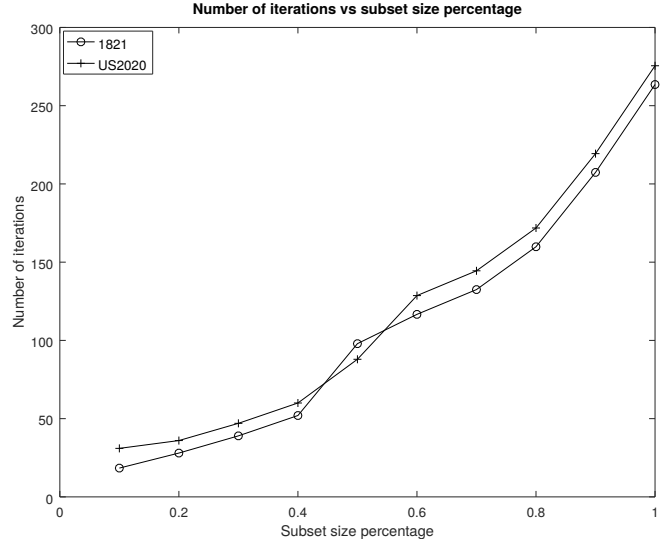


Figure 4. Number of iterations vs subset size.

terminated earlier. On the contrary, convergence is slow for the US2020 graph after an initial stage of little progress.

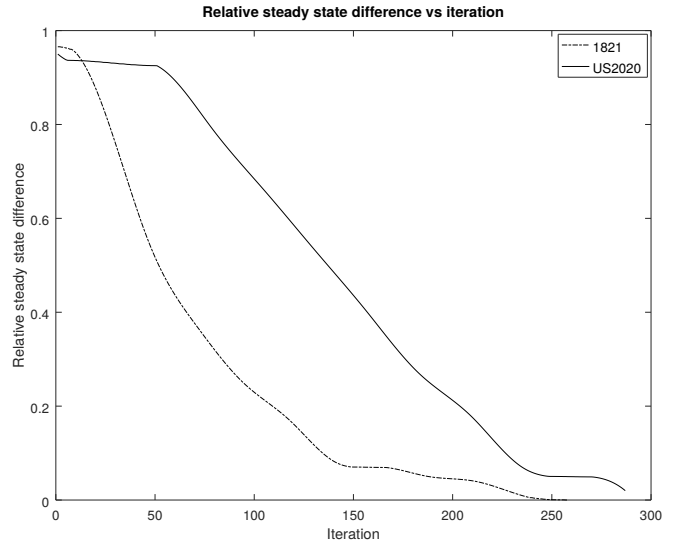


Figure 5. Relative steady state difference vs iteration.

The overall polarity assessment can be defined as the fraction of labels which have changed sign after the application of GNN in the 1821 (left) and the US2020 (right) graphs are shown in table 9. For the former graph it follows that the first order affective polarity classification of algorithm 1 was reasonably successful in identifying positive hashtags as only a third of the accounts initially labeled as positive flipped sign. However, it did not fare that well in the negative case since almost half of the accounts marked as negative changed sign. This can be perhaps explained by its general resiliency with some accounts complaining about individual aspects of the planned events but not about their meaning or about the anniversary itself. The entries

are similar for the US2020 graph where somewhat more accounts flipped signs for both cases.

TABLE 9. FRACTIONS OF SIGN FLIPS (1821/US2020).

	Positive	Negative	Positive	Negative
Positive-GNN	67.33%	32.67%	64.28%	33.72%
Negative-GNN	49.11%	50.89%	45.15%	54.85%

Table 10 shows the total sentiment polarity for the two benchmark graphs. The 1821 graph is more resilient in the sense that there are significantly more accounts with an overall positive stance. The situation is also clear for the US2020 graph but with much less margin.

TABLE 10. SOCIAL GRAPH RESILIENCE.

	Positive	Negative
1821	75.92%	24.08%
US2020	62.18%	37.82%

Given these results, the difference between the NLP and the GNN can perhaps be explained in a much broader sense as follows. Heated Twitter conversations are reported to contain considerable amounts of ironic tweets [56] [57]. Such tweets are not to be taken at face value as their true content hides in the subtext. Therefore, relying on processing them with NLP techniques which are based on syntax alone is unlikely to reveal irony in many cases. On the contrary, GNNs by aggregating locally information about an account yield a more accurate estimation of its real intentions.

4.5. Recommendations

Based on the above results, the following recommendations can be made for researchers and practitioners:

- Higher order methods such as GNNs are more likely to achieve better results.
- The computational complexity of the GNN is not prohibitive for social graphs of the size presented here. Still, for Internet scale graphs distributed implementations should be considered.
- Learning and extracting latent non-trivial knowledge from skewed data is achieved by the GNNs.

5. Conclusions And Future Work

The focus of this conference paper is the assessment of the resilience of large Twitter social graphs based on combination structural, functional, and affective attributes with graph neural networks. Results from two benchmark graphs with similar characteristics have resulted in different results concerning resilience. The higher order of GNNs can mine high dimensional data and discover latent patterns.

Regarding future research directions, the GNN scaling capabilities should be further investigated. Additionally, refined aggregation models factoring friendship direction should be developed. Also, weight schemes involving local influence matrix should be examined.

Acknowledgment

This conference paper is part of the Interreg V-A Greece-Italy Programme 2014-2020 project “Fostering capacities and networking of industrial liaison offices, exploitation of research results, and business support” (ILONET), co-funded by the European Union, European Regional Development Funds (ERDF), and by the national funds of Greece and Italy.

References

- [1] F. B. Keller, D. Schoch, S. Stier, and J. Yang, “Political astroturfing on Twitter: How to coordinate a disinformation campaign,” *Political Communication*, vol. 37, no. 2, pp. 256–280, 2020.
- [2] M. Cornfield, “Empowering the party-crasher: Donald J. Trump, the first 2016 GOP presidential debate, and the twitter marketplace for political campaigns,” *Journal of Political Marketing*, vol. 16, no. 3-4, pp. 212–243, 2017.
- [3] B. L. Ott, “The age of Twitter: Donald J. Trump and the politics of debasement,” *Critical studies in media communication*, vol. 34, no. 1, pp. 59–68, 2017.
- [4] G. Drakopoulos and E. Kafeza, “One dimensional cross-correlation methods for deterministic and stochastic graph signals with a Twitter application in Julia,” in *SEEDA-CECNSM*. IEEE, 2020.
- [5] J. Tang, X. Shu, R. Yan, and L. Zhang, “Coherence constrained graph LSTM for group activity recognition,” *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [6] W. J. Tan, A. N. Zhang, and W. Cai, “A graph-based model to measure structural redundancy for supply chain resilience,” *International Journal of Production Research*, vol. 57, no. 20, pp. 6385–6404, 2019.
- [7] G. Drakopoulos, “Tensor fusion of social structural and functional analytics over Neo4j,” in *IISA*. IEEE, 2016.
- [8] M. Lai, C. Bosco, V. Patti, and D. Virone, “Debate on political reforms in Twitter: A hashtag-driven analysis of political polarization,” in *DSAA*. IEEE, 2015.
- [9] S. Wang, A. Maolinyazi, X. Wu, and X. Meng, “Emo2Vec: Learning emotional embeddings via multi-emotion category,” *ACM Transactions on Internet Technology*, vol. 20, no. 2, pp. 1–17, 2020.
- [10] P. Ekman, “Basic emotions,” *Handbook of cognition and emotion*, vol. 98, no. 45-60, p. 16, 1999.
- [11] B. Mønsted, P. Sapiezynski, E. Ferrara, and S. Lehmann, “Evidence of complex contagion of information in social media: An experiment using Twitter bots,” *PLoS one*, vol. 12, no. 9, 2017.
- [12] L. F. Barrett and A. B. Satpute, “Large-scale brain networks in affective and social neuroscience: towards an integrative functional architecture of the brain,” *Current opinion in neurobiology*, vol. 23, no. 3, pp. 361–372, 2013.
- [13] S. J. Blair, Y. Bi, and M. D. Mulvenna, “Aggregated topic models for increasing social media topic coherence,” *Applied Intelligence*, vol. 50, no. 1, pp. 138–156, 2020.
- [14] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, “Weisfeiler and Leman go neural: Higher-order graph neural networks,” in *AAAI*, vol. 33, 2019, pp. 4602–4609.
- [15] Z. Xinyi and L. Chen, “Capsule graph neural network,” in *International conference on learning representations*, 2018.
- [16] J. Kim, T. Kim, S. Kim, and C. D. Yoo, “Edge-labeling graph neural network for few-shot learning,” in *CVPR*, 2019, pp. 11–20.
- [17] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in *The WWW conference*, 2019, pp. 417–426.

- [18] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *KDD*, 2020, pp. 1150–1160.
- [19] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," 2018.
- [20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [21] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *ICASSP*. IEEE, 2011, pp. 5528–5531.
- [22] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *SLT workshop*. IEEE, 2012, pp. 234–239.
- [23] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.
- [24] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *CVPR*, 2015, pp. 1110–1118.
- [25] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *NIPS*, 2017, pp. 3697–3707.
- [26] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.
- [27] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *CVPR*, 2020, pp. 14424–14432.
- [28] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, "Sound classification using convolutional neural network and tensor deep stacking network," *IEEE Access*, vol. 7, pp. 7717–7727, 2019.
- [29] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2012.
- [30] Y. Bai, J. Fu, T. Zhao, and T. Mei, "Deep attention neural tensor network for visual question answering," in *ECCV*, 2018, pp. 20–35.
- [31] G. Drakopoulos and P. Mylonas, "Evaluating graph resilience with tensor stack networks: A keras implementation," *NCAA*, 2020.
- [32] J. C. Bridgeman and C. T. Chubb, "Hand-waving and interpretive dance: An introductory course on tensor networks," *Journal of Physics A: Mathematical and Theoretical*, vol. 50, no. 22, 2017.
- [33] G. Hu, Y. Hua, Y. Yuan, Z. Zhang, Z. Lu, S. S. Mukherjee, T. M. Hospedales, N. M. Robertson, and Y. Yang, "Attribute-enhanced face recognition with neural tensor fusion networks," in *ICCV*, 2017, pp. 3744–3753.
- [34] D. Palzer and B. Hutchinson, "The tensor deep stacking network toolkit," in *IJCNN*. IEEE, 2015, pp. 1–5.
- [35] J. Kangas, T. Kohonen, and J. Laaksonen, "Variants of self-organizing maps," *IEEE transactions on neural networks*, vol. 1, no. 1, pp. 93–99, 1990.
- [36] K. Kiviluoto, "Topology preservation in self-organizing maps," in *ICNN*, vol. 1. IEEE, 1996, pp. 294–299.
- [37] T. Kohonen, "Exploration of very large databases by self-organizing maps," in *ICNN*, vol. 1. IEEE, 1997, pp. PL1–PL6.
- [38] G. Drakopoulos, I. Giannoukou, P. Mylonas, and S. Sioutas, "On tensor distances for self organizing maps: Clustering cognitive tasks," in *DEXA*. Springer, 2020, pp. 195–210.
- [39] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *PNAS*, vol. 96, no. 6, pp. 2907–2912, 1999.
- [40] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Deep self-organizing maps for unsupervised image classification," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5837–5845, 2019.
- [41] D. S. Palguna, V. Joshi, V. Chakaravarthy, R. Kothari, and L. V. Subramaniam, "Analysis of sampling algorithms for Twitter," in *Joint Conference on Artificial Intelligence*, 2015.
- [42] F. Godin, V. Slavkovicj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using topic models for Twitter hashtag recommendation," in *WWW*, 2013, pp. 593–596.
- [43] G. Drakopoulos, E. Kafeza, P. Mylonas, and H. Al Katheeri, "Building trusted startup teams from LinkedIn attributes: A higher order probabilistic analysis," in *ICTAI*. IEEE, 2020.
- [44] J. Vosecky, D. Jiang, K. W.-T. Leung, K. Xing, and W. Ng, "Integrating social and auxiliary semantics for multifaceted topic modeling in Twitter," *ACM Transactions on Internet Technology*, vol. 14, no. 4, pp. 1–24, 2014.
- [45] K. Moshksar and A. K. Khandani, "Arbitrarily tight bounds on differential entropy of Gaussian mixtures," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3340–3354, 2016.
- [46] N. M. Razali, Y. B. Wah *et al.*, "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, and Anderson-Darling tests," *Journal of statistical modeling and analytics*, vol. 2, no. 1, pp. 21–33, 2011.
- [47] S. Xu and A. Zhou, "Hashtag homophily in twitter network: Examining a controversial cause-related marketing campaign," *Computers in Human Behavior*, vol. 102, pp. 87–96, 2020.
- [48] L. van Vliet, P. Törnberg, and J. Uitermark, "The Twitter parliamentary database: Analyzing Twitter politics across 26 countries," *PLoS one*, vol. 15, no. 9, 2020.
- [49] Y.-y. Zhu, B. Zhang, Q. A. Wang, W. Li, and X. Cai, "The principle of least effort and Zipf distribution," in *Journal of Physics: Conference Series*, vol. 1113, no. 012007, 2018, pp. 1–11.
- [50] S. Šćepanović, I. Mishkovski, B. Gonçalves, T. H. Nguyen, and P. Hui, "Semantic homophily in online communication: Evidence from Twitter," *Online Social Networks and Media*, vol. 2, pp. 1–18, 2017.
- [51] J. A. Caetano, H. S. Lima, M. F. Santos, and H. T. Marques-Neto, "Using sentiment analysis to define twitter political users classes and their homophily during the 2016 American presidential election," *Journal of internet services and applications*, vol. 9, no. 1, pp. 1–15, 2018.
- [52] L. A. Adamic, B. A. Huberman, A. Barabási, R. Albert, H. Jeong, and G. Bianconi, "Power-law distribution of the World Wide Web," *Science*, vol. 287, no. 5461, pp. 2115–2115, 2000.
- [53] D. S. Callaway, M. E. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs," *Physical review letters*, vol. 85, no. 25, pp. 54–68, 2000.
- [54] D. Lorenz and K.-J. Wirths, "Sarrus rules for matrix determinants and dihedral groups," *The College Mathematics Journal*, vol. 49, no. 5, pp. 333–340, 2018.
- [55] Z. Bi, C. Faloutsos, and F. Korn, "The "d_{gx}" distribution for mining massive, skewed data," in *KDD*. ACM, 2001, pp. 17–26.
- [56] J. Á. González, L.-F. Hurtado, and F. Pla, "Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter," *Information Processing & Management*, vol. 57, no. 4, 2020.
- [57] E. Sulis, D. I. H. Farías, P. Rosso, V. Patti, and G. Ruffo, "Figurative messages and affect in Twitter: Differences between# irony,# sarcasm and# not," *Knowledge-Based Systems*, vol. 108, pp. 132–143, 2016.