Applying a Passive Network Reconstruction Technique to Twitter Data in Order to Identify Trend Setters

Vasu Chetty[†], Nathan Woodbury[‡], Jacob Brewer[*], Kenneth Lee[II], and Sean Warnick[◇]

Abstract— **In this work we apply a systems-theoretic approach to identifying trend setters on Twitter. A network reconstruction algorithm was applied to Twitter data to determine causal relationships among topics discussed by popular Twitter users. Causal relationships in this context means that the topics tweeted by a single user influences the topics tweeted by another user, regardless of sentiment. A user that causally influences other users, without themselves being strongly influenced is identified as a trendsetter. This work seeks to identify potential trendsetters among popular Twitter users and demonstrating that causal influence does not always directly correlate with a user's popularity in terms of followers–demonstrating that popularity alone may not be sufficient for identifying trendsetters on Twitter.**

Twitter is a popular online social networking site that allows users to communicate both publicly and privately using short messages–known as *tweets*–with a 140-character limit. Twitter has become an important platform to study due to its substantial growth over the last decade, with over 300 million users in 2016, an order of magnitude increase since 2010, and over 500 million tweets recorded daily [1].

Twitter users, like many social media sites, form relationships in several ways:

1) **Follower**: Twitter can be described by a directed graph detailing which users are subscribed to follow tweets from other users. Each user has a personalized feed where tweets from users they have followed appear. This relationship appears useful for detailing influence of Twitter users since it details the number of users who are likely to read their tweets. Researchers, however, have discovered that using high follower count alone to detail influence is a weak assumption and other important Twitter relationships need to be taken into account in order to define influence [2].
2) **Mentions**: Mention relationships occur when a Twitter user, which we will call User A, includes the user handle (i.e. their username, including the @ tag so that it links back to the user) of another Twitter user, which we will call User B. This indicates that User A is trying to contact User B, is talking about User B, or is responding to User B. This is a good metric for influence since it means many people are discussing User B or in discussions with User B. However, this can be potentially be abused by smaller groups of users continually tweeting mentions of each other, something Twitter tries to minimize through its acceptable use policies by limiting multiple accounts by users, etc. [3].
3) **Retweets**: Whenever User A *retweets* a tweet from User B, the followers of User A views User B's tweet on their feeds. This means the tweets of User B, i.e. their ideas, are potentially propagated to a new set of users.
4) **Hashtags**: Hashtags are words that begin with the # symbol and are used to highlight the topic of their tweet. Common hashtags among multiple users means those users are tweeting about the same topic, even though they might not be directly related by any of the previous three relationships.

Many companies–such as Klout, PeerIndex, and Kred–have tried to numerically assign an influence score for users based on some combination of the above metrics. These companies are all websites that aggregate a user's online activity on popular social media sites, such as Twitter, to score their influence.

Many researchers have also studied influence and information propagation among Twitter users with approaches such as:

1) Applying clustering algorithms to graphs created by follower relationships [4]
2) Using a combination of the follower, mention, and retweet metrics, while taking into account the user's total number of tweets [5]
3) Tracking the progress of a single topic among followers and followers of followers [6]
4) Using statistical methods to determine information propagation based on human-labeled tweets [7]

While these research approaches and software solutions help model the propagation and magnitude of social influence, they may not shed adequate light on the agent that originally triggered the social movement. For example, one of these existing approaches may conclude that Larry King initiated a viral social media wave with a specific post, however, King's post may have been inspired or triggered by a less noticeable social media user.

This work is the first, to the best of our knowledge, to apply systems-theoretic techniques to the problem of

This material is based on research sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency (HSARPA), Cyber Security Division (DHS S&T/HSARPA/CSD), LRBAA 12-07 via contract numbersHSHQDC-13-C-B0052 and HSHQDC-15-C-B0056. All material in this paper represents the position of the authors and not necessarily that of DHS.

Department of Computer and Information Sciences, Brigham Young University - Hawaii. [†] vasu.n.chetty@gmail.com, [II] leechinhongkenneth@gmail.com

Information and Decision Algorithms Laboratories, Brigham Young University. [‡] nathanswoodbury@gmail.com, [*] jakebrucebrewer@gmail.com, [◇] sean.warnick@gmail.com

identifying influence among users. We detail a potential influence network based on the timing in which users discuss various topics over time and compare the network to follower metrics to demonstrate similarities and differences between this new technique and existing metrics.

Our approach differs from other approaches by looking at how users tweet topics *over time*, ignoring standard metrics such as follower, mentions, retweets, and hashtags. Influence is then determined by finding users who continually discuss topics before other users adopt the topic. We only look at the relationships among popular users (20 randomly selected English-tweeting users from the top 100 users based on number of followers) to identify trendsetters among a group of users that are known to be influential by some metric.

Topics are automatically assigned to each user's tweets using natural language processing (NLP) techniques. Finding trends and applying topic modeling to Twitter has been well-studied in the literature [8], [9] and parts of these procedures are applied in this work to convert tweets over time into signals over time.

Section I defines the dynamical structure function, a mathematical system representation used to detail the causal relationships among manifest variables. Sections II-A and II-B walk through the process of turning tweets by users over time into signals over times by assigning topics to each tweet using standard natural language processing (NLP) methods. Section II-C details the network reconstruction procedure used on the converted Twitter data. Results are presented in Section III. Finally, conclusions and potential future work is described in Section IV.

I. Background

The dynamical structure function (DSF) is a convenient way to represent the signal structure of a linear time-invariant (LTI) system [10], [11], [12]. Note that, throughout this work, all references to systems are in actuality references to LTI systems.

Let $\mathbb{P}(z)$ be the space of all rational polynomials in terms of $z \in \mathbb{C}$. The DSF of a discrete-time LTI system of m inputs and p outputs is characterized by the pair $(Q(z), P(z))$ where $P(z) \in \mathbb{P}^{p\times m}(z)$ and $Q(z) \in \mathbb{P}^{p\times p}(z)$, and where all diagonal entries $Q_{ii}(z)$ of $Q(z)$ are constrained to be zero. If $Y(z)$ and $U(z)$ are the frequency-domain representation of the outputs and inputs of some system, then the DSF relates inputs and outputs through the following equation:

$$Y(z) = Q(z)Y(z) + P(z)U(z), \tag{1}$$

where entries (often called *links* or *modules*) in $P(z)$ define the direct causal mapping from individual inputs to individual outputs and entries in $Q(z)$ define the direct causal mapping from individual outputs to other outputs.

Note that the transfer function matrix $G(z)$ is the black-box mapping from inputs to outputs given by

$$Y(z) = G(z)U(z). \tag{2}$$

By solving for $Y(z)$ in (1) we can define the relationship between the transfer function and the DSF in the following equation:

$$G(z) = (I - Q(z))^{-1}P(z). \tag{3}$$

The equation in (3) highlights the fact that the DSF is a left factorization of the transfer function and potentially contains more structural information about the system than the transfer function (see [13], [14]). Since DSFs can abstract away some of the information contained in the transfer function, a DSF can be reconstructed from input-output data while only requiring low a priori knowledge about the system, as opposed to the reconstruction of a state space representation.

II. Methodology

The process for determining the causal relationships among Twitter users is as follows:

1) Collect data from 20 randomly selected popular users on Twitter.
2) Implement natural language techniques to convert tweets over time into a signal of numeric values over time.
3) Use a network reconstruction technique to determine causal relationships among the selected users, as given by a DSF.

A. Twitter Data Collection

Twenty users were selected at random from the top one hundred Twitter users. Twitter's streaming API was implemented using *tweepy*, a Python wrapper on the Twitter API. The users tweets were collected for roughly six weeks, from early January 2017 to mid-February 2017. During that period 2,295 tweets were collected, the highest volume user composed 370 tweets and the lowest volume user composed 23 tweets.

B. Twitter Data Preprocesing Procedure

Once the tweets were collected, the procedure for converting them to numeric signals was conducted as follows:

1) We combined the tweets of a single user into a single document.
2) We then tokenized and cleaned the documents by removing urls, stopwords, punctuation, mentions (words beginning with @), and short words (3 letters or less). A few custom four letter words were also removed (with, from, this, have, and that) and the resulting words were lemmatized using the *nltk* library in Python.
3) After cleaning the documents, we used the Hierarchical Dirichlet Process (HDP), which is a built-in function in the *gensim* Python library, to determine topics in documents. Note that unlike the Latent Dirichlet Allocation (LDA), the HDP does not require the number of topics to be specified a priori. However, we must still choose a reasonable threshold for the number of topics. Gensim details each topic using a list of ten relevant words. We chose the topics whose relevant words described a large percentage of the topic.

4) Given the topics, we then applied word2vec (again through *gensim*) to the topics as well as the tokenized tweets in order to convert these tweets into numeric data. The word2vec model was first trained on all words in all cleaned tweets before being applied to the topics.
5) We then compared the similarity of each word in each tweet to each of the ten relevant words associated with each topic. We assign each tweet a topic based on the largest similarity between any word in a tweet and any relevant word in a topic using the word2vec similarity function in *gensim*. The topic label for each tweet was projected down onto the reals using the 2-norm since it maintains a notion of distance between each tweet.
6) Given a topic for each tweet, we use 24 hours as our time step in order to get a signal of vectors over time. If a person tweeted more than once over the course of a 24-hour period, we averaged the topic score for each tweet.

Once the tweets for each user were transformed into signals over time, or time-series data, we were ready to apply a network reconstruction algorithm to determine causal relationships among the users.

C. Network Reconstruction of Twitter Data

A network reconstruction problem takes time-series input-output data and finds the unique DSF of the form (1) that best fits the data. This approach is standard in the network reconstruction literature, using either the DSF or a similar representation [15], [16], and has been used to recover biochemical reaction networks from data [17], [18], [19].

In particular, we are seeking to find the unique $Q(z)$ that best fits the output data $Y(z)$ such that

$$Y(z) = Q(z)Y(z) + P(z)U(z), \tag{4}$$

where $P(z)$ and $U(z)$ are unknown. The network reconstruction algorithm used in this work is derived from that presented in [20]. The algorithm proceeds as follows:

1) Let $\mathcal{D}_y \in \mathbb{R}^{T\times p}$ contain the p measured outputs from the system at times $t = 1, 2, \dots, T$. Define $y(t)^\mathsf{T}$ to be the t'th row of $\mathcal{D}_y$. In this work, $\mathcal{D}_y$ contains the Twitter data processed as described in Section II-B.
2) Choose some $r \le T$, where r is the estimated point at which all $Q_{ij}(t) = 0$ for $t > r$. For this work, we choose $r = T$.
3) Construct vector $\hat{y} \in \mathbb{R}^{p(T-1)}$ and matrix $\hat{L} \in \mathbb{R}^{p(T-1)\times rp^2}$ such that

$$\hat{y} = \begin{bmatrix} y(2)^\mathsf{T} & \cdots & y(2)^\mathsf{T} & \cdots & y(T)^\mathsf{T} \cdots & y(T)^\mathsf{T} \end{bmatrix}^\mathsf{T}, \tag{5}$$

and

$$\hat{L} = \begin{bmatrix} y(1)^\mathsf{T} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & y(1)^\mathsf{T} & \cdots & 0 & 0 & 0 \\ & \vdots & & & & \vdots & \\ y(r)^\mathsf{T} & 0 & 0 & \cdots & y(1)^\mathsf{T} & 0 & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & y(r)^\mathsf{T} & \cdots & 0 & 0 & y(1)^\mathsf{T} \\ & \vdots & & & & \vdots & \\ y(T-1)^\mathsf{T} & 0 & 0 & \cdots & y(T-r)^\mathsf{T} & 0 & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & y(T-1)^\mathsf{T} & \cdots & 0 & 0 & y(T-r)^\mathsf{T} \end{bmatrix}. \tag{6}$$

4) Let $\vec{q} \in \mathbb{R}^{rp^2}$ be a placeholder for all of the unknown values in $Q(t)$ (note that, as we are assuming that $Q(t) = 0$ for $t > r$, there are only rp^2 of these unknowns) such that:

$$\vec{q} = \begin{bmatrix} \vec{q}(1)^\mathsf{T} & \cdots & \vec{q}(r)^\mathsf{T} \end{bmatrix}^\mathsf{T}. \tag{7}$$

5) Since $Q(t)$ is hollow, we can throw away all entries in $\vec{q}(t)$ corresponding to each $Q_{ii}(t)$. Let $\hat{x} \in \mathbb{R}^{r(p^2-p)}$ be the resulting vector. We also need to remove all columns in $\hat{L}$ corresponding to the entries removed from $\vec{q}$. Let $\hat{M} \in \mathbb{R}^{p(T-1)\times r(p^2-p)}$ be the resulting matrix.
6) We can now rewrite (4) as

$$\hat{y} = \hat{M}\hat{x} + e. \tag{8}$$

We seek to minimize $\|e\|_2$. We use ordinary least squares to find a $\hat{x}$ that best fits this data, or in other words, we find the $\vec{x}$ that solves the problem

$$\hat{x}^* = \arg\min_{\hat{x}} \|e\|_2 = \arg\min_{\hat{x}} \|\hat{y} - \hat{M}\hat{x}\|_2. \tag{9}$$

7) For each link (i, j), extract the items from $\hat{x}^*$ corresponding to that link into $\tilde{Q}_{ij}(t)$.
8) The convolutional model of each link (i, j) is given by

$$Q_{ij}(t) = a_{q_{ij}}\delta_{(t,0)} + \sum_{n=1}^{w_{q_{ij}}} b_{n,q_{ij}}\left(c_{n,q_{ij}}\right)^t, \tag{10}$$

where $c_{n,q_{ij}} \in \mathbb{R}$, $-1 < c_{n,q_{ij}} < 1$ (since the dynamics on all links are assumed to be stable), $w_{q_{ij}} \in \mathbb{N}$ is the number of delays in link (i, j), $b_{n,q_{ij}} \in \mathbb{R}$, $a_{q_{ij}} = -\sum_{n=0}^{w_{q_{ij}}} b_{n,q_{ij}}$, and $\delta_{(t,0)}$ is the Kronecker delta.
For each link, we fit a model of the form (10) by choosing the parameters $b_{n,q_{ij}}$ and $c_{n,q_{ij}}$ to minimize the error

$$\epsilon = \sum_{t=0}^{r} |Q_{ij}(t) - \tilde{Q}_{ij}(t)|. \tag{11}$$

This problem can be solved using a non-convex optimization technique, such as an evolutionary algorithm. Note that, for tractability, we choose $w_{q_{i,j}} = 6$ and bound $-20 \le b_{n,q_{ij}} \le 20$. The bounds on $b_{n,q_{ij}}$ are rarely reached; nonetheless, it may be possible to slightly improve the quality of the results by increasing these bounds or increasing $w_{q_{i,j}}$.

9) Finally, we build $Q(z)$ from (10) such that

$$\begin{aligned} \alpha_{n,q_{ij}} &= b_{n,q_{ij}} c_{n,q_{ij}}, \\ \beta_{n,q_{ij}} &= c_{n,q_{ij}}, \\ Q_{ij}(z) &= \sum_{n=1}^{w_{q_{ij}}} \frac{\alpha_{n,q_{ij}}}{z - \beta_{n,q_{ij}}}. \end{aligned} \tag{12}$$

D. On the Reconstructability of Twitter Networks

The network reconstruction algorithm depends on a few assumptions of the data and the systems which generate the data in order to be able to reconstruct a network. We discuss those assumptions and their applicability to the Twitter data here.

- **Linearity:** Network reconstruction as presented in this work assumes that the dynamics of the underlying network are linear. However, it is almost guaranteed that the actual dynamics of the network are nonlinear. Therefore, in performing a network reconstruction on Twitter data, we are implicitly assuming that the dynamics are near enough to linear to be approximated by a linear network.
- **Stability:** In order to use the reconstruction algorithm on Twitter data, we must assume that links in $Q(z)$ and $P(z)$ are stable, or in other words, that the impulse responses $Q(t)$ and $P(t)$ converge to zero as $t \to \infty$. The fact that the impulse responses in Figure 1 tend towards zero is good evidence that it is safe to make this assumption.
- **Strict Causality:** The reconstruction algorithm assumes that dynamics are strictly causal, meaning that tweets in the present can only influence other individuals strictly in the future. Since the Twitter data is captured at a daily resolution, this implies that tweets must take a full day to influence the network. This assumption likely does not hold completely as tweets may affect other individuals faster than a day. However, we still will build an approximate network by enforcing strict causality in the learned dynamics.
- **Informativity:** Informativity states that $\hat{M}$ must be injective in order to reconstruct. The data $\mathcal{D}_y$ was rich enough that it was in the range of $\hat{M}$.
- **Measurement Precision:** The reconstruction algorithm has been shown to be particularly sensitive to additive noise on output measurements. We can assume that there is no noise in the system, as we are capturing the tweets exactly; however, the function of turning tweets into numeric signals implies that the network reconstructed from the resultant data is a network of that function and not necessarily the original twitter network.
- **Data Quantity:** We require a T large enough to ensure that we don't overfit the least squares model in (8). We also need to choose r large enough that the dynamics of each $Q_{ij}(t)$ have time to converge to zero. As indicated in Figure 1, the impulse responses tend to have enough time to converge; however, the quality of the reconstruction may be improved by choosing a larger r, which is only possible if a larger T is available as well.

III. Results

The impulse responses of the reconstructed twitter dynamics are shown in Figures 1 and 2. Figure 1 contains a sample of 3 of the 380 impulse responses $Q_{ij}(t)$ reconstructed from the twitter data. The red dots contain the elements of $\vec{q}$ which were extracted from $\vec{x}$ in step 7 of the reconstruction algorithm. Notice how these are converging towards zero as t increases, indicating that the dynamics of this network really are stable. Notice also that they have time to converge to zero within $t \leq r = 34$ timesteps, indicating that we have selected enough data to properly reconstruct the network. The blue lines in this figure are the convolutional model of the impulse response, fit from $\vec{q}$ in step 8 of the reconstruction algorithm.

Let $M \in \mathbb{R}^{p \times p}$ be defined such that $m_{ij} = \|Q_{ij}(z)\|_\infty$, meaning M contains the link magnitudes reconstructed in $Q(z)$. Let $N \in \mathbb{R}^{p \times p}$ be defined such that $n_{ij} = \frac{m_{ij}}{\max_{k,l} m_{kl}}$, or in other words, N contains the relative link magnitudes of the reconstructed network. Matrix N thus forms a weighted adjacency matrix, and can be used to plot the reconstructed network. This plot is shown in Figure 2. Note, $n_{ij} \neq 0$ signifies a link from j to i, the transpose of the typical adjacency matrix convention.

Note that a non-zero magnitude on a link from person to person does not necessarily signify a direct influence of one person on another (though it could). For example, a non-zero link showing an influence from Justin Bieber to Donald Trump does indicate that a tweet from Justin Bieber will cause a reaction from Donald Trump. However, this does not signify that Justin Bieber's tweets directly influence Donald Trump's tweets. Rather it signifies that Justin Bieber's tweets influence his followers that are not represented in this network, which in turn influence other followers, and so on in a network chain to Donald Trump that does not pass through any of the other people in this network.

Matrix N can also be used to compute the individuals who are the largest influencers. The influencer score of an individual is the column sum of N related to that individual, normalized such that the largest influencer has a score of 1. Matrix N can also be used to determine who are the most influenced by others in this network. The influenced score of an individual is the row sum of N related to that individual, also normalized. We also compute a trendsetter score as the ratio of the influencer score to the influenced score, again normalized.

The top three influencers in this network, as shown in Figure 3, are

- Larry King
- Kev Adams
- Justin Bieber

The top three influenced, as shown in Figure 3, are

- Larry King

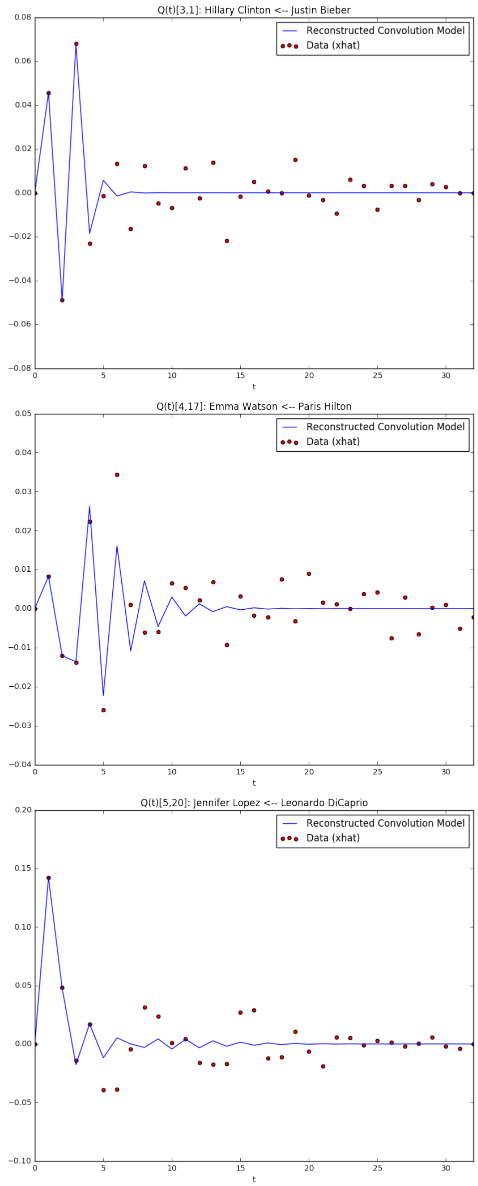

Fig. 1. Sample impulse responses from the reconstruction algorithm applied to the Twitter data. Red dots are the elements of $q_{ij}(t)$ extracted from $\vec{x}$ in step 7 and the blue line is the convolutional model $Q_{ij}(t)$ fit in step 8.

- Stephen Fry
- Justin Bieber

The top three trendsetters, as shown in Figure 3, are

- Tyra Banks
- Cara Delevingne
- Kev Adams

Notice from Figure 3 that in several cases, individuals with high influencer scores also have large numbers of followers, and those with high influenced scores tend to also follow many accounts. For example, if we consider only Tyra Banks, Joe Rogan, Cameron Dallas, Phillip Schofield, Kat Von D, and Cara Delevingne, the correlation between the influencer score and the log of the number of followers is 0.66 and the correlation between the influenced score and the log of

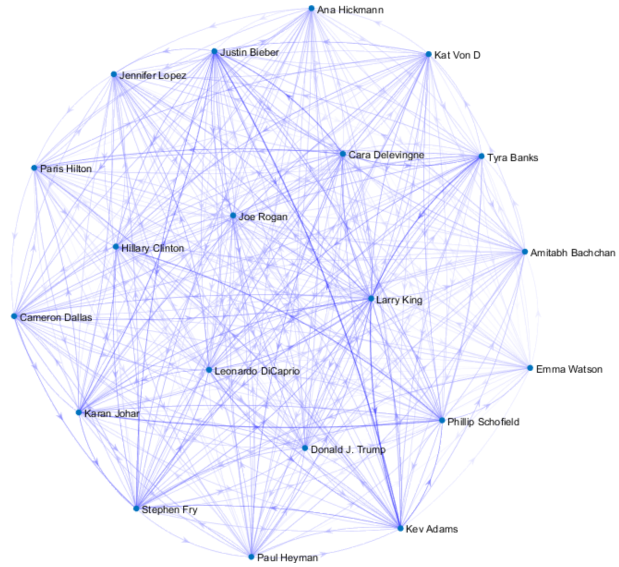

Fig. 2. The relative magnitude of links, $\|Q_{ij}(z)\|_\infty$, reconstructed by the reconstruction algorithm on the Twitter data. The thickness of each line is proportional to its magnitude.

the number following is 0.74. Since number of followers and number following were not included in the dataset used for reconstruction, this is evidence that the reconstruction algorithm is functioning as desired.

That said this relationship does not hold in general (the correlation between influencer score and log of followers is -0.19 for all individuals in this network, and the correlation between influenced score and log of number following is 0.46), indicating that the network reconstruction process is finding properties of the network that are not possible to see from follower/following score alone.

We can also compute correlations between individuals in this network. Let C be a matrix of correlations, where $c_{ij} = c_{ji}$ is the correlation of column i in $\mathcal{D}_y$ with column j. Figure 4 is a heatmap representation of C on this twitter data. Notice how Joe Rogan and Donald Trump are highly correlated with several other members of this network. However, neither are among the top influencers or top influenced within this network, indicating that network reconstruction is not simply discovering correlations.

IV. Conclusions and Future Work

Therefore, we have developed a new metric for scoring influence within a network of Twitter users based on a systems-theoretic approach to network reconstruction. Given the metric, we determined a method for finding trendsetters not based on a follower-following relationship for Twitter users.

This work should extend easily to social media applications beyond Twitter, although there are several areas that could be changed to possibly improve the results, such as:

1) Increasing number of Twitter users considered and number of tweets collected,
2) Trying different step sizes smaller or large than 24 hours,

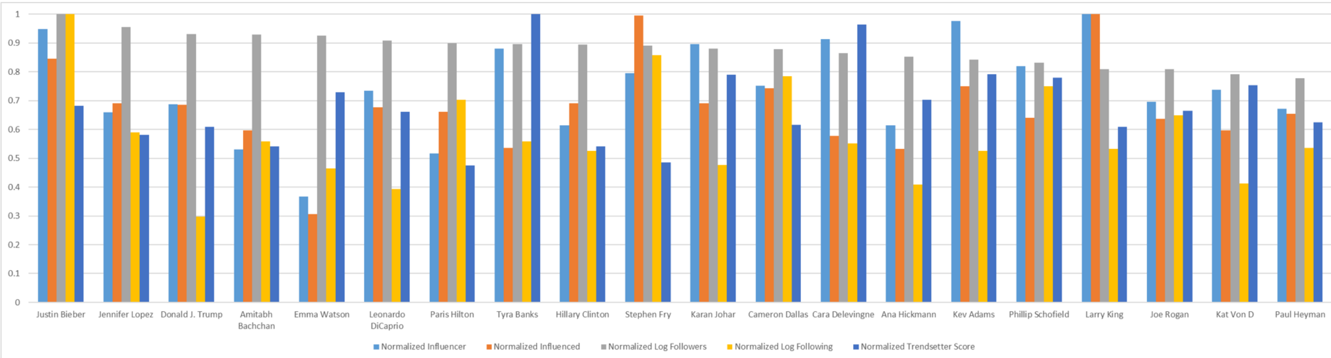

Fig. 3. Various scores for individuals in the reconstructed twitter network. The first column (light blue) is the normalized influencer score for each individual, which are the column sums of matrix N. The second column (orange) is the normalized influenced score for each individual, which are the row sums of matrix N. The third column (gray) is the log of the number of followers for each individual as reported by twitter, normalized. The fourth column (yellow) is the log of the number of people each individual is following as reported by twitter, normalized. The final column (dark blue) is the trendsetter score, which is the ratio of influencer score to influenced score, normalized.

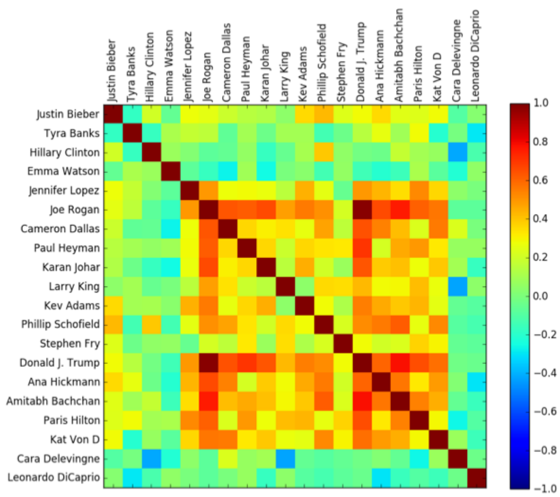

Fig. 4. Pairwise correlations between individuals from the Twitter data.

3) Topic selection using HDP, identifying whether simple selection of some number of popular topics would be better for labeling the tweet than our current procedure,
4) Using a projection method other than the 2-norm that better maintains the notion of distance between the word2vec vectors,
5) Further development of the robust, blind, passive network reconstruction method for dynamical structure functions focusing especially on whether the properties of stability or strict causality could be removed

References

[1] R. Krikorian, "New Tweets per Second Record, and How! Twitter Official Blog," August 16 2013.
[2] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring User Influence in Twitter: The Million Follower Fallacy." *Icwsm*, vol. 10, no. 10-17, p. 30, 2010.
[3] The Twitter Rules. Accessed: 2017-02-22. [Online]. Available: https://support.twitter.com/articles/18311
[4] P. Feng, "Measuring user influence on twitter using modified k-shell decomposition," 2011.
[5] I. Anger and C. Kittl, "Measuring influence on twitter," in *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. ACM, 2011, p. 31.
[6] S. Ye and S. F. Wu, "Measuring message propagation and social influence on twitter. com," in *International Conference on Social Informatics*. Springer, 2010, pp. 216–231.
[7] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 65–74.
[8] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 80–88.
[9] A. Zubiaga, D. Spina, R. Martinez, and V. Fresno, "Real-time classification of twitter trends," *Journal of the Association for Information Science and Technology*, vol. 66, no. 3, pp. 462–473, 2015.
[10] J. Gonçalves and S. Warnick, "Necessary and sufficient conditions for dynamical structure reconstruction of lti networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1670–1674, 2008.
[11] E. Yeung, J. Goncalves, H. Sandberg, and S. Warnick, "Mathematical relationships between representations of structure in linear interconnected dynamical systems," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 4348–4353.
[12] Y. Yuan, A. Rai, E. Yeung, G.-B. Stan, S. Warnick, and J. Gonçalves, "A minimal realization technique for the dynamical structure function of a class of lti systems," *IEEE Transactions on Control of Network Systems*, 2015.
[13] V. Chetty and S. Warnick, "Network Semantics of Dynamical Systems," *Conference on Decision and Control*, 2015.
[14] S. Warnick, "Shared Hidden State and Network Representations of Interconnected Dynamical Systems," in *53rd Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL, 2015.
[15] D. Materassi and M. V. Salapaka, "On the Problem of Reconstructing an Unknown Topology via Locality Properties of the Wiener Filter," *Transactions on Automatic Control*, vol. 57, no. 7, pp. 1765–1777, 2012.
[16] D. Hayden, Y. Yuan, and J. Gonçalves, "Robust Network Reconstruction in Polynomial Time," *Conference on Decision and Control*, 2012.
[17] J. Adebayo, T. Southwick, V. Chetty, E. Yeung, Y. Yuan, J. Gonçalves, J. Grose, J. Prince, G.-B. Stan, and S. Warnick, "Dynamical Structure Function Identifiability Conditions Enabling Signal Structure Reconstruction," *Conference on Decision and Control*, 2012.
[18] Y. Yuan, G.-B. Stan, S. Warnick, and J. Gonçalves, "Robust Dynamical Network Structure Reconstruction," *Automatica, special issue on Systems Biology*, vol. 47, no. 6, pp. 1230–1235, 2011.
[19] C. Ward, E. Yeung, T. Brown, B. Durtschi, S. Weyerman, R. Howes, J. Goncalves, H. Sandberg, and S. Warnick, "A comparison of network reconstruction methods for chemical reaction networks," in *Proceedings of the Foundations for Systems Biology and Engineering Conference*, 2009, pp. 197–200.
[20] V. Chetty, J. Eliason, and S. Warnick, "Passive Reconstruction of Non-Target-Specific Discrete-Time LTI Systems," *American Control Conference*, 2016.