Trending Topics on Twitter Improve the Prediction of Google Hot Queries

Federica Giummolè
Università Ca' Foscari Venezia
Venezia, Italy
Email: giummole@unive.it

Salvatore Orlando
Università Ca' Foscari Venezia
Venezia, Italy
Email: orlando@unive.it

Gabriele Tolomei
Università Ca' Foscari Venezia
Venezia, Italy
Email: gabriele.tolomei@unive.it

***Abstract*—Once every five minutes, *Twitter* publishes a list of *trending topics* by monitoring and analyzing tweets from its users. Similarly, *Google* makes available hourly a list of *hot queries* that have been issued to the search engine. In this work, we analyze the time series derived from the *daily volume index* of each trend, either by Twitter or Google. Our study on a real-world dataset reveals that about 26% of the trending topics raising from Twitter "as-is" are also found as hot queries issued to Google. Also, we find that about 72% of the *similar* trends appear first on Twitter. Thus, we assess the relation between comparable Twitter and Google trends by testing three classes of time series regression models. We validate the *forecasting* power of Twitter by showing that models, which use Google as the *dependent* variable and Twitter as the *explanatory* variable, retain as significant the past values of Twitter 60% of times.**

***Keywords*-Time series analysis; Time series regression; Social network analysis; Twitter; Trending topics; Google; Hot trends**

I. Introduction

Twitter[1] is one of the most popular online social media and microblogging platform where people share information nearly real-time by reading and writing short text messages called *tweets*. Twitter may rely on a huge amount of user-generated data, which can be analyzed to provide better monetization for the company and new services for the end users (e.g., *personal advertising*).

Some analyses aim at showing what are the *topics* that users are most interested in (or talking of) by using the Twitter platform. To this end, Twitter periodically extracts and publishes a list of the top-10 *trending topics*, namely text strings referring to *social trends*. Each trending topic is the succinct textual representation of a "standing out fact", as extracted from user tweets. It may either refer to a long-lasting or a sudden effect on the volume of tweets (e.g., nba *vs.* election 2012). Unfortunately, the details about the technique Twitter uses to generate its lists of trending topics are not publicly known, even though recent work is able to infer them with high accuracy [1].

Similarly to Twitter, once every hour *Google*[2] releases a list of the top-20 trending search keywords (i.e., the so-called *hot trends* or *hot queries*), which we refer to as *web trends*. Google allows us to quantitatively examine how trending is a hot query. Specifically, it computes the search fraction of a given query in a time range and a geographical region. This analysis indicates the likelihood of a random user to ask for a query from a certain location at a certain time.

The aim of this work is to investigate whether any relationship occurs between social trends as extracted from Twitter and web trends as output by Google. Intuitively, we claim that the same topics that appear as trending on Twitter could *later* become a trending query on Google.

To motivate our research, we show in Fig. 1 a pair of daily-based time series, referring to the first two weeks of November 2012, and regarding the U.S. Presidential Elections event. The main issue concerns the different time granularity of observations of trend volumes as derivable from Twitter and Google. Since Google only publishes a daily-aggregate analysis of its web trend volumes, no finer-grained time series (e.g., hourly-based) can be produced from trend data, and this forces us to compare series on a daily basis only. Despite this issue, the plots in Fig. 1 reveal that a *predictive relation* exists between each pair of Twitter's and Google's trend time series. As opposed to the trend time series, the first-time occurrence of a trend can be derived hourly from both Twitter and Google. Indeed, Twitter updates its set of trending topics once every five minutes whereas Google does it every hour. We found that 66% of times the same trend appears first on Twitter and then on Google.

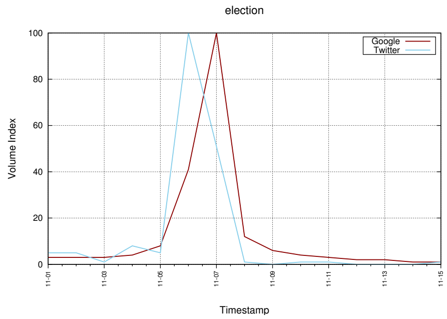

Figure 1. Time series of trend volumes: Twitter vs. Google.

The main contributions of this paper are thus the following: *(i)* a *Trend Bipartite Graph* (TBG) to represent the lexical similarity between any pair of social and web trends,

[1] http://www.twitter.com
[2] http://www.google.com

as extracted from real-world Twitter and Google datasets; *(ii)* the use of TBG to build the Twitter/Google time series to analyze and compare; *(iii)* a statistic regression analysis to measure the ability of Twitter in actually *predicting* and *causing* a Google trend to later occur. Models that include Twitter in the regression function better fit and forecast our time series data.

II. Related Work

In the recent years, we have seen an exponential growth of many online *social network applications*, such as *Flickr*, *MySpace*, *Facebook*, *Google+*, just to name a few. Among them *Twitter* has emerged as one of the most influential online social media service. Thereby, several studies have started analyzing data from Twitter. Many work aim at classifying different types of users, their behaviors, and the relationships occurring among them according to the *following/follower* pattern [2]–[5]. Other studies focus on analyzing the content of the tweets and the way these are related to trending topics. In this last regard, one of the most representative and exhaustive study is proposed by Kwak *et al.* [6]. Among other contributions, the authors describe the relation between tweets and trending topics extracted from Twitter, and trends derived from other media, i.e., search queries on Google and CNN headlines. Although the subject above seems to be highly similar to the one we tackle in this paper, we compare Twitter and Google trends from a very different perspective. Specifically, Kwak *et al.* aim at checking if Twitter trends and Google hot queries overlap (in fact, the authors consider a trending topic and a search keyword a match if the length of the longest common substring is more than 70% of either string). Instead, our goal is to test – through time series regression analysis – if Twitter trending topics can be used to model, explain, and predict the volume of Google hot queries.

Ruiz *et al.* [7] exploit the general findings on Twitter discussed in [6] yet studying the problem of correlating microblogging activity with stock market events.

Using Web data for predicting the behavior of a real time series is another well-investigated topic. However, to the best of our knowledge, the present work is the first attempt trying to relate time series derived *both* from Web search and social network data.

Liu *et al.* [8] propose a unified model to predict the upcoming query trends on the basis of observations extracted from the query log of a commercial search engine. Recent work prove that Web search volume can predict the values of some economic indicators. For instance, Bordino *et al.* [9] show that daily trading volumes of NASDAQ-100 stocks are correlated with daily volumes of queries about the same stocks. Ettredge *et al.* [10] use search logs to predict the job market while Choi and Varian [11] show how Google trends may be used to forecast unemployment levels, car and home sales, and disease prevalence in near real-time [11]. Goel *et al.* [12] show that what users are searching for online can also predict their collective future behavior days or even weeks in advance. Furthermore, Ginsberg *et al.* [13] propose to approximate the flu cases in the U.S. by using a search engine query log whereas Corely *et al.* [14] address a similar problem yet exploiting blog content. Finally, Goel *et al.* [15] claim also that, though correct, these prediction models are not really competitive when compared to models that exploit domain knowledge, and that social media data – though usually very large – are not always statistically representative of the population [16]. In [17] Goel *et al.* show evidence of differences between the distribution of demographic characteristics of a country and that of Twitter users from that country.

III. Social vs. Web Trends Analysis

In this section, we introduce and formalize the main elements and techniques used in our research.

Trend Vocabularies. Let $\mathcal{V}_X = \{x_1, x_2, \ldots, x_n\}$ be the *vocabulary* set of all the *trending topics* as provided by Twitter. Similarly, we denote by $\mathcal{V}_Y = \{y_1, y_2, \ldots, y_m\}$ the *vocabulary* set of all the *hot queries* released by Google.

It is worth remarking that the keywords of both vocabularies are not necessarily single-term, but may be composed of a sequence of terms.

Trend Scores. We refer to $\mathcal{T} = \langle t_1, t_2, \ldots, t_T \rangle$ as a discrete time interval. Functions s_X and s_Y assign *scores* to vocabulary keywords over time $\mathcal{T}$:

$$s_X : \mathcal{V}_X \times \mathcal{T} \longmapsto \mathbb{N},\ s_Y : \mathcal{V}_Y \times \mathcal{T} \longmapsto \mathbb{N}.$$

For each keywords, they indicate the “strength” of its trending in a given time slot, as measured by Twitter and Google.

Trend Time Series. Each Twitter/Google trend can be modeled as a *time series*, composed of t_T *random variables*, namely $\mathcal{X} = \{X_t\}_{t=t_1}^{t_T}$ for Twitter, and $\mathcal{Y} = \{Y_t\}_{t=t_1}^{t_T}$ for Google. Each random variable evaluates to a trending score of a given trend. More formally, let $s_X(x_i, t)$ be the trending score of $x_i \in \mathcal{V}_X$, and $s_Y(y_j, t)$ be the trending score of $y_j \in \mathcal{V}_Y$, as measured at time $t \in \mathcal{T}$. The *observed time series* for $x_i \in \mathcal{V}_X$ and $y_j \in \mathcal{V}_Y$ correspond to the sequences of values assumed by each X_t and Y_t:

$$\mathcal{X}_i = \{X_t = s_X(x_i, t)\}_{t=t_1}^{t_T},\ \ \mathcal{Y}_j = \{Y_t = s_Y(y_j, t)\}_{t=t_1}^{t_T}.$$

Moreover, we define the aggregate time series resulting from each pair of Twitter series $\mathcal{X}_i$ and $\mathcal{X}_j$, as follows:

$$\mathcal{X}_i \uplus \mathcal{X}_j = \{s_X(x_i, t) \oplus s_X(x_j, t)\}_{t=t_1}^{t_T},$$

where $\oplus$ is a normalized sum.

Trend Bipartite Graph (TBG). Analyzing *any* pair of time series derived from a Twitter trend (x_i) and a Google hot query (y_j) might be useless or even misleading. In this study, we focus on series associated with trends that are likely related, namely to vocabulary keywords that are somehow "similar". In addition, we aim at combining groups of series referring to Twitter trends that are alike. To this end, we introduce the *Trend Bipartite Graph* (TBG), as follows.

Definition 3.1 (Trend Bipartite Graph):
Let TBG $= (\mathcal{V}_X, \mathcal{V}_Y, E, w, \eta)$ be a *bipartite graph* where:
– $\mathcal{V}_X$ and $\mathcal{V}_Y$, the vocabularies of Twitter and Google trends, respectively, are the two *disjoint* sets of graph *nodes*[3];
– $E \subseteq \mathcal{V}_X \times \mathcal{V}_Y$ is the set of graph *edges*;
– $w : \mathcal{V}_X \times \mathcal{V}_Y \longmapsto [0, 1]$ is an *edge weighting function*;
– $\eta \in [0, 1]$ is a weight *threshold*, such that $E = \{(x_i, y_j) \in (\mathcal{V}_X, \mathcal{V}_Y) \mid w(x_i, y_j) \geq \eta\}$.
Intuitively, the TBG links any pair of trends from Twitter and Google with an edge weighted by a function w, which measures the pairwise *trend similarity*. The threshold η is in turn used to avoid linking those trends that are low related to each other.

Several trend similarity functions can be used. The simplest approach only looks at the lexical surfaces of trends, thereby computing a string similarity score (e.g., *Levenshtein distance*, *longest common substring*, *n-gram similarity*, etc.). Advanced solutions might take into account the *semantics* of each trend (e.g., by linking trends to referent *entities* of an external knowledge base like *Wikipedia* [18], [19]).

The TBG allows us to identify a set of "comparable" time series pairs, whose associated trends are similar, and which are defined by $\mathcal{S} = \{(\mathcal{X}_i, \mathcal{Y}_j) \mid (x_i, y_j) \in E\}$.

In a nutshell, for each $\mathcal{Y}_j$ we retrieve a set of related series $\mathcal{X}_i$ according to the TBG, and we aggregate them together obtaining $\mathcal{S}_{\mathcal{Y}_j} = \biguplus_{(\mathcal{X}_i, \mathcal{Y}_j) \in \mathcal{S}} \mathcal{X}_i$. We can finally define a set of comparable time series, where the first element of each pair results from the aggregation of several Twitter series: $\mathcal{D} = \{(\mathcal{S}_{\mathcal{Y}_j}, \mathcal{Y}_j) \mid \mathcal{S}_{\mathcal{Y}_j} \neq \emptyset\}$. It is worth remarking that we combine only Twitter series because it is more likely that multiple trending topics refer to the same Google hot query than vice versa. Furthermore, if $\eta = 1$ then $\mathcal{S}$ includes only those x_i that *exactly* matches y_j, namely those trends that are actually *shared* "as-is" between Twitter and Google. If this is the case, it holds that $\mathcal{D} = \mathcal{S}$.

Trend Time Series Regression. Our final goal is to check whether the evolution of a trend from Twitter is *significant* to explain and predict the behavior of its counterpart trend as extracted from Google.

Concretely, given a TBG $= (\mathcal{V}_X, \mathcal{V}_Y, E, w, \eta)$ and the dataset of related pairs of time series $\mathcal{D}$, for each $(\mathcal{X}_i, \mathcal{Y}_j) \in \mathcal{D}$ we evaluate the capability of $\mathcal{X}_i$ (Twitter) in *forecasting/causing* $\mathcal{Y}_j$ (Google). To perform this step, we validate the following *time series regression models* to fit our data:

- *Autoregressive Models*: AR(p);
- *Distributed Lag Models*: DL(q);
- *Autoregressive Distributed Lag Models*: ADL(p, q).

Roughly speaking, AR(p) tries to fit the time series data from the *dependent* variable (Google) using a linear combination of up to its own p past observations, i.e., $Y_t = \alpha + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \epsilon_t$. It turns out that this model does not take into account any Twitter explanatory variable at all. Conversely, DL(q) models the dependent variable only using a linear combination of up to q past observations from a hypothetical explanatory variable (Twitter), i.e., $Y_t = \alpha + \psi_1 X_t + \psi_2 X_{t-1} + \ldots + \psi_{q+1} X_{t-q} + \epsilon_t$. Finally, ADL($p, q$) uses past observations from *both* the dependent and the explanatory variables:

$$Y_t = \alpha + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \\ + \psi_1 X_t + \psi_2 X_{t-1} + \ldots + \psi_{q+1} X_{t-q} + \delta t + \epsilon_t.$$

IV. Experiments

In this section, we describe the experiments we conducted on a real-world dataset of trends from Google and Twitter. The experimental phase is divided into three separate tasks:

A. Raw Data Crawling: to collect both Google and Twitter data, thereby deriving the actual time series of trends.

B. Time Series Datasets Building: to build the time series datasets from the "raw" Google and Twitter data crawled.

C. Time Series Regression Analysis: to conduct the regression analysis for exploring relation between Twitter and Google trends.

A. Raw Data Crawling

In the very first step, we crawl all the data both from Google and Twitter, which are necessary for deriving the final datasets of time series. Concretely, we collect data for fifteen consecutive days, namely from 2012-11-01 at 00:00AM UTC to 2012-11-15 at 11:59PM UTC. However, since Google and Twitter have their own services and access policies for getting data, we describe this task separately.

1) **Google Hot Queries***:* Google has recently released *Google Hot Trends*[4], which is a tool displaying the top-20 *hot*, i.e., fastest rising, queries (search-terms) of the past hour in the United States.

Furthermore, Google provides each hot query with a daily *search volume index*, which is a normalized integer score ranging from 0 to 100 computed as follows. Given a specific range of dates, i.e., $[d_{start}, d_{end}]$, and a hot query q, we denote by $svi(q, d)$ the search volume index of q on day d, such that $d_{start} \leq d \leq d_{end}$. Google assigns $svi(q, d^*) = 100$ on day d^* with the highest search volume traffic of q. For any other day $d' \neq d^*$, $svi(q, d')$ results from normalizing the search volume of q on d' with respect to d^*, and $svi(q, d^*)$.

At the end of the crawling step, we collect 24 daily lists, thereby resulting in $15 * 24 = 288$ lists, each one containing 20 hot queries. After in-depth pre-processing and cleaning

[3] Though the same string may occur in both vocabularies, elements of those two sets are *conceptually* separated.

[4] http://www.google.com/trends/hottrends/atom/hourly

stages, we derive a final vocabulary of 190 unique hot queries, i.e., $|\mathcal{V}_Y| = 190$, which is available for download.[5]

2) **Twitter Data***:* Twitter allows to interact with its platform by exposing a useful REST Application Programming Interface (API).[6] Roughly, two main functionalities are available throughout this API: *Search* and *Streaming*.

We perform two crawling tasks running in parallel: from a side, we use the Search API to retrieve the list of top-10 trending topics, once every 5 minutes. It is worth noting that Twitter refreshes the list of trends *exactly* every 5 minutes, thereby no trend data are lost during this step. On the other hand, we collect a sample of the public tweets throughout the Streaming API.

i) Trending Topics. This crawling task collects 12 top-10 lists of U.S. *trending topics* for each hour, thereby resulting in a total of up to $12 * 10 = 120$ hourly trends. However, to align this dataset of trends with that provided by Google, we need to "collapse" each block of top-10 lists into a combined top-10 hourly list. More generally, this is an instance of the *rank aggregation* problem [20], which aims at combining several ordered lists in a proper and efficient way. However, we here omit further details on this due to space constraints. The final vocabulary of Twitter trends contains 892 entries, i.e., $|\mathcal{V}_X| = 892$, and is available for download.[7]

ii) Public Timelines. The other crawling task we perform concerns the retrieval of a sample of tweets from the public timelines of Twitter. To be consistent with other collections, we focus only on tweets coming from the U.S., which are almost all written in English. As a result, we obtain a total amount of about 260 million tweets, which means that more than 17 million tweets have been crawled per day on average.

B. Time Series Datasets Building

In this section, we discuss how actual time series of trends have been built from the raw datasets collected from Google and Twitter, as detailed above.

Each trend associated with $x_i \in \mathcal{V}_X$, $y_j \in \mathcal{V}_Y$ is observed during a time interval $\mathcal{T}$, which corresponds to the range of dates used to crawl our data.

However, only Google provides each hot query with a score $s_Y(y_j, t)$, namely the *daily search volume index* denoted by $svi(y_j, t)$ (see Section IV-A1). Thus, $\mathcal{T}$ can be divided into single days, i.e., $\mathcal{T} = \{t_1, t_2, \ldots, t_{15}\}$. For each hot query y_j we obtain 15 daily observations, each one equals to the daily search volume index:

$$s_Y(y_j, t) = svi(y_j, t), \quad \text{where } t = t_1, \ldots, t_{15}.$$

Note that these are the finest-grained observations we can obtain for Google hot queries. Therefore, the resulting time series for y_j is $\mathcal{Y}_j = \{Y_t = svi(y_j, t)\}_{t=t_1}^{t_{15}}$.

Since Twitter trends do not come with any score, in order to make Google and Twitter time series comparable, we have to figure out a score also for each Twitter trend, which is similar to the Google's daily search volume index. Thereby, we exploit the whole collection of public tweets obtained as described in Section IV-A2, and for each trend $x_i \in \mathcal{V}_X$ we compute the *daily trend volume index* $tvi(x_i, t),\ t \in \mathcal{T}$, as follows. Let $count(x_i, t)$ be the number of occurrences of the trend x_i in the public set of tweets during the day t. Then, the daily trend volume index is:

$$tvi(x_i, t) = \left\lceil \frac{count(x_i, t)}{\max \bigcup_{t \in \mathcal{T}} count(x_i, t)} \right\rceil * 100, \qquad (1)$$

where $\max \bigcup_{t \in \mathcal{T}} count(x_i, t)$ is the maximum daily count of x_i, as measured across all the days in the interval.

Like Google's search volume index, also Twitter's trend volume index is a normalized integer score ranging from 0 to 100. For each Twitter trend x_i we have 15 daily observations, each one equals to the daily trend volume index:

$$s_X(x_i, t) = tvi(x_i, t), \quad \text{where } t = t_1, \ldots, t_{15}.$$

Finally, the resulting time series for the Twitter trending topic x_i is $\mathcal{X}_i = \{X_t = tvi(x_i, t)\}_{t=t_1}^{t_{15}}$.

To compare Twitter's and Google's time series, we pair those derived from *similar* trends. To this end, we build the *Trend Bipartite Graph* (TBG) starting from the trend vocabularies $\mathcal{V}_X$ and $\mathcal{V}_Y$, according to Section III. We define the edge weighting function w as the string similarity between any pair of Twitter and Google trends x_i and y_j, respectively. Specifically, we use a *normalized longest common substring* score ($nlcs$), defined as follows:

$$nlcs(x_i, y_j) = \frac{|lcs(x_i, y_j)|^2}{|x_i||y_j|},$$

where $|lcs(x_i, y_j)|$ is the length of the longest string of characters that is a substring of both x_i and y_j.

Actually, we use two similarity thresholds, i.e., $\eta_1 = 1.0$ and $\eta_2 = 0.6$, thereby obtaining two graphs $\mathsf{TBG}_1 = (\mathcal{V}_X, \mathcal{V}_Y, E_1, nlcs, \eta_1)$ and $\mathsf{TBG}_2 = (\mathcal{V}_X, \mathcal{V}_Y, E_2, nlcs, \eta_2)$. We denote by $\mathcal{S}_1 = \{(\mathcal{X}_i, \mathcal{Y}_j) \mid (x_i, y_j) \in E_1\}$ and $\mathcal{S}_2 = \{(\mathcal{X}_i, \mathcal{Y}_j) \mid (x_i, y_j) \in E_2\}$ the two sets of time series pairs derived from TBG_1 and TBG_2, respectively.[8] [9] Obviously, $|\mathcal{S}_1| \leq |\mathcal{S}_2|$, since TBG_1 contains a less number of edges: in particular, we find 50 pairs of trends for $\mathcal{S}_1$, and 69 for $\mathcal{S}_2$.

In addition, we aggregate and normalize the series associated with Twitter trends, which are connected to the the same Google hot query in the bipartite graphs. Thus, for the two graphs we generate two sets $\mathcal{D}_1$ and $\mathcal{D}_2$ (see Section III), where $\mathcal{D}_1 = \mathcal{S}_1$, while $\mathcal{D}_2$ contains pairs, each coupling an aggregate Twitter time series with one derived from Google.

[5] http://bit.ly/YiO6AD

[6] https://dev.twitter.com/docs/api

[7] http://bit.ly/120h0tQ

[8] http://bit.ly/YIEsFD

[9] http://bit.ly/YrKnwA

C. Time Series Regression Analysis

We are interested in evaluating two phenomena about social and web trends: *(i) forecasting* and *(ii) causality*. The former refers to the power of Twitter trends in predicting their counterpart Google hot queries whereas the latter goes a step further and tries to determine *causality* between Twitter and Google. Both issues require dealing with time series regression models introduced in Section III.

To assess the first aspect, we consider each pair of time series $(\mathcal{X}_i, \mathcal{Y}_j)$ in our running datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, individually. We start fitting each series to *autoregressive* models, i.e., AR(p). Intuitively, this means that we are trying to explain the behavior of a trend time series from Google $Y_{j,t}$ at day t, by only considering the values of the *same* series as measured up to p days before (i.e., $Y_{j,t-1}, \ldots, Y_{j,t-p}$). Thereby, these models assume Google trends depending *only* on themselves, and Twitter having no influence at all.

On the other hand, we introduce the second class of regression models, namely *distributed lag* DL(q). As opposed to AR(p), DL(q) models try to fit a Google time series from $Y_{j,t}$ at day t, by only looking at the paired time series from Twitter as measured up to q days before (i.e., $X_{j,t}, \ldots X_{j,t-q}$).

We measure how many AR(p) models retain as *significant* their p-lagged component, and, similarly, how many DL(q) models keep as *significant* their q-lagged component. In our experiments, we set the *significance level* $\alpha = 0.05$, and for each class we limit the lag order of the model p and q up to a maximum of 3 days.

As the last step, for each tested model we compute the *adjusted coefficient of determination*, denoted by $\hat{R}^2 \in [0, 1]$ and averaged by all the pairs of time series. This is generally used to describe how well a regression line fits a dataset, and provides a measure of how well future outcomes are likely to be predicted by the model yet penalizing models with too many explanatory terms. In Tab. I, we show the results for both the two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, which can be interpreted as follows. Each entry in the table measures the percentage of models AR(p) and DL(q) where the p- and q-lagged component was significant. Very unlikely, time series derived from Google hot queries can be explained by AR(p) models, namely only using their past values. Indeed, only 10% of the times the negative lag-1 component is significant. Conversely, when DL(q) models are used, the negative lag-1 component of Twitter is significant 60% of the times.

Furthermore, we check if there is any causality between Twitter and Google trends. This is achieved by comparing AR(p) models with *autoregressive distributed lag* models, i.e., ADL(p, q). These models use up to p and q values from *both* the dependent (i.e., Google) and explanatory (i.e., Twitter) trend time series, respectively. Concretely, we measure the ratio of ADL(p, q) models where q-lagged component of the explanatory variable is significant. Clearly, this value is less than (or at most equal to) that we find when comparing AR(p) with DL(q) models because we are now evaluating a relation that is "stronger" than forecasting. Tab. II shows that in more than 40% pairs of time series Twitter trend "causes" Google hot queries if we limit to $q = -1$. This percentage evaluates to about 70% if we restrict only to those pairs who have already shown a significant q-lagged component in the corresponding DL(q) model.

	AR (%)	$\hat{R}^2$	p	DL (%)	$\hat{R}^2$	q
$\mathcal{D}_1$	10.0	0.02	1	**60.0**	**0.75**	1
	18.0	0.04	2	30.0	0.72	2
	14.0	0.03	3	16.0	0.71	3
$\mathcal{D}_2$	13.3	0.02	1	**56.7**	**0.75**	1
	15.0	0.03	2	25.0	0.73	2
	13.3	0.01	3	18.3	0.73	3

Table I
TIME SERIES REGRESSION: AR(p) VS. DL(q).

	$\mathcal{D}_1$		$\mathcal{D}_2$	
(p, q)	ADL (%)	$\hat{R}^2$	ADL (%)	$\hat{R}^2$
$(1, 1)$	**42.0**	**0.79**	**43.3**	**0.80**
$(2, 2)$	18.0	0.76	16.7	0.79
$(3, 3)$	18.0	0.76	18.3	0.78

Table II
TEST FOR CAUSALITY USING ADL(p, q).

By looking at the values of $\hat{R}^2$, ADL$(1, 1)$ is the model that best fit our data, on average. This result sounds reasonable because it mixes the autoregressive component of Google with the prediction of Twitter, as captured one day before.

V. Conclusion and Future Work

In this work, we explored possible relations between *trending topics* rising from Twitter (i.e., *social trends*) and *hot queries* issued to Google (i.e., *web trends*). We claimed that a trending topic on Twitter could later become a hot query on Google as well. Indeed, information flooding nearly real-time across the Twitter social network could anticipate the set of topics that users will be interested in – thereby will search for – in the near future.

To validate our claim, we provided the following contributions. First, we introduced the *Trend Bipartite Graph* (TBG) to represent the lexical similarity between any pair of social and web trends, as extracted from real-world Twitter and Google datasets. The TBG used a *threshold* to link those trends that were most likely related. Then, we measured the ability of Twitter in actually *predicting* and *causing* a Google trend to later occur by conducting an exhaustive comparison of several time series regression models. This step showed that models including Twitter in the regression function better fit and forecast our time series data. Specifically, we found that models, which used Google as the *dependent* variable and Twitter as the *explanatory* variable, retained as significant the past values of Twitter 60% of times. Moreover, we discovered that a Twitter trend *caused* a similar Google trend to later occur about 43% of times. Finally, we showed that the best-performing models were those using past values of *both* Twitter and Google.

As future work, we plan to extend this study by considering trending signals coming from other social and web platforms.

References

[1] L. Hardesty, "Predicting what topics will trend on Twitter," http://bit.ly/PIsBbr, October 2012.

[2] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Proceedings of WebKDD/SNA-KDD '07*. New York, NY, USA: ACM, 2007, pp. 56–65.

[3] B. Krishnamurthy, P. Gill, and M. Arlitt, "A few chirps about twitter," in *Proceedings of WOSN '08*. New York, NY, USA: ACM, 2008, pp. 19–24.

[4] D. Zhao and M. B. Rosson, "How and why people twitter: the role that micro-blogging plays in informal communication at work," in *Proceedings of GROUP '09*. New York, NY, USA: ACM, 2009, pp. 243–252.

[5] B. A. Huberman, D. M. Romero, and F. Wu, "Social networks that matter: Twitter under the microscope," *First Monday*, vol. 14, no. 1, 2009.

[6] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of WWW '10*. New York, NY, USA: ACM, 2010, pp. 591–600.

[7] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proceedigs of WSDM '12*. New York, NY, USA: ACM, 2012, pp. 513–522.

[8] N. Liu, J. Yan, S. Yan, W. Fan, and Z. Chen, "Web query prediction by unifying model," in *Proceedings of ICDMW '08*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 436–441.

[9] I. Bordino, S. Battiston, G. Caldarelli, M. Cristelli, A. Ukkonen, and I. Weber, "Web search queries can predict stock market volumes," *PloS One*, vol. 7, no. 7, p. e40014, 2012.

[10] M. Ettredge, J. Gerdes, and G. Karuga, "Using web-based search data to predict macroeconomic statistics," *Communications of the ACM*, vol. 48, no. 11, pp. 87–92, November 2005.

[11] H. Choi and H. Varian, "Predicting the present with google trends," *Economic Record*, vol. 88, pp. 2–9, 2012.

[12] S. Goel, J. M. Hofman, S. Lahaie, D. M. Pennock, and D. J. Watts, "Predicting consumer behavior with Web search," *Proceedings of the National Academy of Sciences*, vol. 107, no. 41, pp. 17 486–17 490, Oct 2010.

[13] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, pp. 1012–1014, 2009.

[14] C. Corley, A. R. Mikler, K. P. Singh, and D. J. Cook, "Monitoring influenza trends through mining social media," in *Proceedings of BIOCOMP*, 2009, pp. 340–346.

[15] S. Goel, J. M. Hofman, S. Lahaie, D. M. Pennock, and D. J. Watts, "What can search predict?" http://bit.ly/11xxMKf, 2010.

[16] D. Gayo-Avello, "A warning against converting Social Media into the next Literary Digest," *Communications of the ACM*, 2011.

[17] D. Gayo-Avello, P. T. Metaxas, and E. Mustafaraj, "Limits of electoral predictions using twitter," in *Proceedings of ICWSM '11'*, L. A. Adamic, R. A. Baeza-Yates, and S. Counts, Eds. The AAAI Press, 2011.

[18] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *Proceedings of CIKM '07*. New York, NY, USA: ACM, 2007, pp. 233–242.

[19] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee, "Twiner: named entity recognition in targeted twitter stream," in *Proceedings of SIGIR '12*. New York, NY, USA: ACM, 2012, pp. 721–730.

[20] V. Pihur, S. Datta, and S. Datta, "Rankaggreg, an r package for weighted rank aggregation." *BMC Bioinformatics*, vol. 10, 2009.