Twitter Boosts the Prediction of Requests for Wikipedia Articles

Gabriele Tolomei, Salvatore Orlando
Università Ca' Foscari Venezia, Italy
Email: {orlando,gabriele.tolomei}@unive.it

Diego Ceccarelli, Claudio Lucchese
ISTI-CNR, Pisa, Italy
Email: {diego.ceccarelli,claudio.lucchese}@isti.cnr.it

***Abstract*—Most of the tweets that users exchange on *Twitter* make implicit mentions of *named-entities*, which in turn can be mapped to corresponding *Wikipedia* articles using proper *Entity Linking* (EL) techniques. Furthermore, some of those become *trending entities* on Twitter due to a long-lasting or a sudden effect on the volume of tweets where they are mentioned. We argue that the set of trending entities discovered from Twitter may help predict the volume of requests for relating Wikipedia articles. The rationale of our intuition is that the nearly real-time nature of Twitter could select in advance the set of articles that users will later look up on Wikipedia. To validate this claim, in this work we provide the following contributions. First, we apply an EL technique to extract trending entities from a real-world dataset of public tweets. Then, we analyze the time series derived from the *hourly trending score* (i.e., an index of popularity) of each entity as measured by Twitter and Wikipedia, respectively. Our results reveals that...**

I. Introduction

Twitter[1] is a popular online social media and microblogging platform where people share information nearly real-time by posting so-called *tweets*. Each user tweet is a text message limited to 140 characters that may contain opinions or feelings on something related to either personal or public interests.

Tweets make often mentions of *named-entities*, such as person names, places, etc. Some of these entities may become "extraordinary popular" on Twitter, due to a long-lasting or a sudden effect on the volume of tweets where they are mentioned. This may be a signal that something relating to those entities has taken or is taking place, and in this work we investigate deeply this intuition by contrasting these Twitter signals with those coming from other sources.

However, we first need an effective method to detect such *trending entities* in Twitter. A possible way is to exploit the list of so-called *trending topics*, which Twitter extracts by its own once every five minutes. This is a set of keyword strings which refer to presumably popular or standing-out facts. Unfortunately, no details about the algorithm used to label a keyword as "trending" were publicly disclosed by Twitter. Moreover, trending topics as fired by Twitter not necessarily link to a well-known knowledge base of entities.

Due to the above reasons, in this work we focus on a simple *Entity Linking* (EL) technique to detect trending entities from Twitter. This solution aims to identify entities from their *mentions* (i.e., small fragments of text referring to any named entity in a knowledge base) occurring in a large corpus of tweets. More precisely, it uses *Wikipedia*[2] as the referring knowledge base of entities and associated mentions. EL is generally a challenging task, and it is even harder when mentions appear in very short texts with not enough surrounding context, such as tweets. In the end, we consider trending entities on Twitter those that are the most frequently mentioned, and which in turn correspond to Wikipedia articles.

The final goal of this research is to investigate whether any relationship exists between trending entities as extracted from Twitter and the trending requests for the corresponding Wikipedia articles. Intuitively, we claim that if an entity appears as trending on Twitter, then a growth of requests for its corresponding Wikipedia article could *later* occur. Furthermore, we also expect the temporal behavior of a trending entity on Twitter might influence, and thus help predict, the access volume to the entity's Wikipedia article. The rationale of this intuition is that information spreading nearly real-time over the Twitter social network could anticipate the set of topics that users will be interested in – and thereby will look up on Wikipedia – in the next future.

Though we do not discuss how our results could be exploited here, we argue that they may lead to several optimization strategies, e.g., the preemptive caching of Wikipedia articles related to entities that started to be trending, or the automatic resolution of ambiguous queries to Wikipedia, which usually lead to multiple articles, since an article related to a trending entity is the most likely result to be returned.

The rest of this paper is organized as follows. Firstly, Section II shows the motivation of our work through a preliminary study of some real-world examples. In Section III, we give an overview of the most valuable work on entity linking using Wikipedia, social network analysis, especially focused on Twitter, and time series regression from Web data. In Section IV we discuss some useful concepts of time series analysis that will be used all along the paper. Section V describes the research steps we pursue to explore whether and what relationships occur between trending entities on Twitter and Wikipedia. Section VI presents the experiments we conducted, and discusses the results we obtained. Finally, in Section VII we summarize our work and point out possible future research directions.

II. Time Relation between Twitter and Wikipedia

To motivate our work, we present some real-world examples of *trending entities*. We start our discussion from Fig. II where each plot shows a pair of *time series*. Each pair presents the *hourly scores* of a trending entity, as measured by Twitter and Wikipedia during the first two weeks of November 2012.

At this stage, we only aim to motivate informally why our

[1] http://www.twitter.com
[2] http://en.wikipedia.org

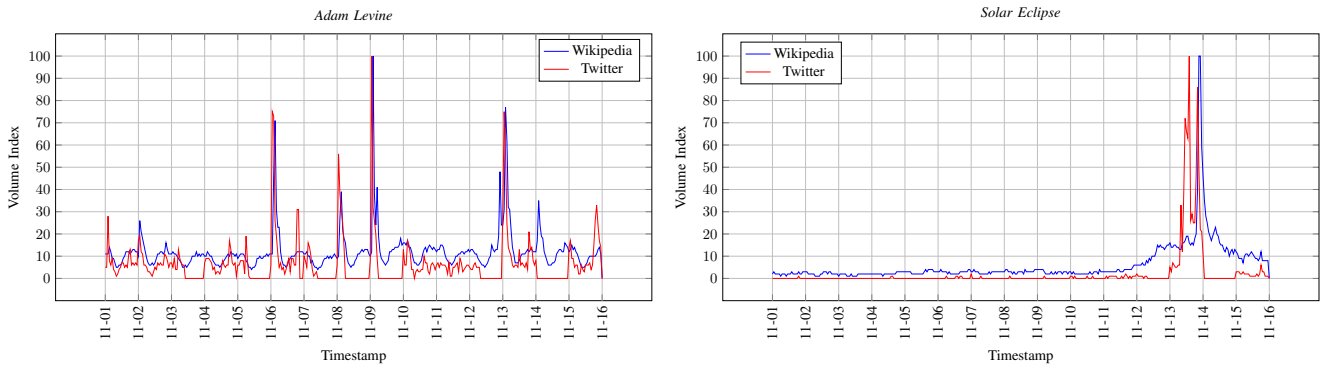

Fig. 1. Time series plots of trending entity scores as measured by Twitter and Wikipedia.

research idea is worthwhile to explore, and the concepts of *trending entity time series* and *trending entity score* will be clarified in the upcoming sections. For now, simply consider each plot as representing the time behavior of two popular entities. Moreover, let the observed values of the series be the (normalized) hourly measures of "popularity" of those entities.

The plot on the left shows a pair of time series about the entity Adam Levine[3], who is a famous American singer and the front man of the "Maroon 5" music band. The other plot shows a pair of time series concerning the entity Solar Eclipse[4], which occurred on last November 13th.

First, it is evident that Twitter and Wikipedia exhibit similar trending scores in both pairs of time series, a part from an almost-constant scaling factor.

Second, if we check what happened to Adam Levine just in correspondence of the three main peaks of Twitter trending scores, we discover that some key events occurred to him, as he was one of the judges of the American reality talent show "The Voice". More precisely, those key events are: live playoffs, the interview at the "Ellen TV Show", and the top-12 live performances. Similarly, the second entity reaches the maximum value of popularity on Twitter just when the actual event was happening (i.e., during the solar eclipse). Therefore, in both pair of time series Twitter truly reveals nearly real-time what hot event is happening.

Third, and even more remarkable, Twitter seems to always anticipates Wikipedia, and this is more evident especially for the highest values of trending entity scores. However, this is shown differently by the two trending entities. Indeed, in the first example Twitter is able to forecast the behavior of Wikipedia one or two hours in advance, even for smaller trending scores[5]. Conversely, for the second entity, Twitter predicts the maximum trending score of Wikipedia largely in advance (i.e., about 12 hours).

We argue that the rationale of these two different forecasting behaviors of Twitter could be the following. When an entity refers to a famous personality, such as Adam Levine, it is reasonable that people start mentioning him on Twitter as something about him is happening (e.g., his performance on a TV show), thus increasing his trending on the social network. Thereby, we can imagine that this popularity pushes people to request for the Wikipedia article of Adam Levine, in order to get information about him as soon as possible (i.e., within a couple of hours that something occurred). Besides, when an entity represents an extraordinary event, still it starts becoming popular on Twitter as the event is running, but this raises people's need for information on Wikipedia less quickly.

Due to all the motivations above, we are strongly convinced that our claim is worth to investigate further.

[3]http://en.wikipedia.org/wiki/Adam_Levine

[4]http://en.wikipedia.org/wiki/Solar_eclipse_of_November_13,_2012

[5]This value is not easily visible from the plot due to the 1-hour scale on the x-axis.

III. Related Work

Three lines of research are actually covered and addressed in this work: *(i)* Entity Linking using Wikipedia, *(ii)* Analysis of Social Network Data, and *(iii)* Time Series Regression from Web Data. In the following, we discuss all of them separately.

Entity Linking using Wikipedia. The first system to use Wikipedia for entity linking is *Wikify!* [1], which works in two separates stages. The first one, i.e., *detection*, aims at identifying the mentions of an entity (Wikipedia article), and the next phase, i.e., *disambiguation*, ensures that the detected mention links to the appropriate article. Milne and Witten [2] largely improves this solution, by exploiting the interdependence between different name mentions as the sum of their pair-wise dependencies. They disambiguate and finally determine the referent entity of a name mention by comparing each of its candidate referents with other name mentions referent entities. In [3] the authors overcome some deficiencies of the previous methods, and propose a graph-based collective entity linking method, which models and exploits the global (rather than the pair-wise) interdependence between different entity linking decisions.

Unfortunately, all the methods above work well on large text documents, where a relatedness measure between candidate entities can be exploited to disambiguate the linking task [2]. Entity linking in Twitter is in fact much harder, since tweets are usually short and informal in nature, and often contain grammatical errors and misspellings. To this end, Li *et al.* [4] propose a novel technique tailored for Twitter that aggregates information gathered from the Web to build both local and global contexts for tweets.

In this paper, we use a pretty simple Wikipedia-based entity linker for Twitter. As a matter of fact, the focus of the paper is not on entity linking, but on the analysis of the time series extracted on the basis of the most trending detected entities. However, we plan to test specialized entity recognizers like the one discussed in [4] as a future work.

Analysis of Social Network Data. In the recent years, we have seen the sudden rising and exponential growth of many online *social network applications*, such as *Flickr*, *MySpace*,

Facebook, *Google+*, just to name a few. Besides all the above, *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 categorizing different types of users, their behaviors, and the relationships occurring among them according to the *following/follower* pattern [5]–[8].

Other studies focus on analyzing the content of the tweets (e.g., to get insights about opinions and/or sentiments [9]), 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.* [10]. Among other things, there authors describe the relationship between tweets and trending topics as extracted from Twitter, and trends derived from other media, i.e., query volume on Google and CNN headlines.

Osborne *et al.* [11] discuss how Wikipedia can be exploited to filter out spurious real-time events detected on Twitter. Among other results, the authors find that there is a delay of one or two hours between events breaking on Twitter and the time when people start to search Wikipedia for information about it. However, differently from our work, they do not use this outcome to perform any prediction analysis of time series.

Ruiz *et al.* [12] study the problem of correlating microblogging activity from Twitter with stock market events. To achieve this goal, they use a graph representation of tweets, whose nodes are different objects (e.g., tweets, users, hashtags, and URLs) and edges model relationships between these objects.

Time Series Regression from Web Data. Using Web data for predicting the behavior of a real time series is a well-investigated topic. However, to the best of our knowledge, this work is the first attempt trying to relate time series derived *both* from Web (i.e., Wikipedia) and social network (i.e., Twitter).

Recent work has proven that Web search volume can predict the *present* values of some economic indicators. For instance, Ettredge *et al.* [13] use search logs to predict the job market while Choi and Varian [14] show how Google trends may be used to forecast unemployment levels, car and home sales, and disease prevalence in near real-time [14].

Finally, Ginsberg *et al.* [15] propose to approximate the flu cases in the U.S. by using a search engine query log whereas Corely *et al.* [16] address a similar problem yet exploiting Web blog content.

IV. Time Series Analysis

In this section, we introduce some basic concepts and notations about *time series*. Then, we discuss a set of techniques used in this paper for extracting knowledge from the time series concerning Twitter and Wikipedia trending entities.

A. Basic Concepts and Notations

Let $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ be a parameter space representing time and containing T *discrete*, *equally-lasting*, and *equally-spaced* slots.

A *time series* is a time-ordered sequence of *random variables* defined on the same probability space and indexed by time slots:

$$\mathcal{X} = \{X_t,\ t \in \mathcal{T}\} = \{X_t\}_{t=t_1}^{t_T}. \tag{1}$$

A time series can be usefully described through the first and second-order moments (i.e., the *mean* and the *variance*) of its composing random variables. To this end, let $E[\cdot]$ be the *expectation operator*. Thus, we define $\mu_t = E[X_t]$ and $\sigma_t^2 = \text{Var}(X_t) = E[X_t - \mu_t]^2$ the mean and variance of each X_t ($t \in \mathcal{T}$) as functions of time.

A crucial issue when dealing with time series concerns *stationarity*. We define a time series *strictly stationary* if its statistical properties do not change over time. Formally, $\mathcal{X} = \{X_t,\ t \in \mathcal{T}\}$ is strictly stationary if, for any $\{t_1, \ldots, t_q\} \subseteq \mathcal{T}$ and any τ, the joint distribution of $X_{t_1}, \ldots, X_{t_q}$ is the *same* as the joint distribution of $X_{t_1+\tau}, \ldots, X_{t_q+\tau}$.

Since this is an extremely strong property, which means that *all* moments of *all degrees* of the series are the same *anywhere*, independent of time, the *second order* or *weak* stationarity is instead often used.

A time series $\mathcal{X} = \{X_t,\ t \in \mathcal{T}\}$ is *weakly stationary* if the means and variances of its random variables are constant over time, and for any $t_i, t_j \in \mathcal{T}$ the *covariance* between X_{t_i} and X_{t_j} is *finite* and only depends on the time *lag* $\delta = j - i$. Eventually, any time series that is not stationary, either strictly or weakly, is called *non-stationary*.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two (weakly) stationary time series. By definition, the means and variances of all their random variables are constant, and we denote them by $\mu_\mathcal{X}, \mu_\mathcal{Y}$ and $\sigma_\mathcal{X}^2, \sigma_\mathcal{Y}^2$, respectively. Given $t \in \mathcal{T}$ and a lag δ, such that $t + \delta \in \mathcal{T}$, we define the *cross-covariance* as:

$$c_{XY}(\delta) = E[(X_{t+\delta} - \mu_\mathcal{X})(Y_t - \mu_\mathcal{Y})]. \tag{2}$$

Furthermore, we compute the *cross-correlation* as the cross-covariance normalized in the range $[-1, 1]$, as follows:

$$r_{XY}(\delta) = \frac{c_{XY}(\delta)}{\sqrt{\sigma_\mathcal{X}^2 \cdot \sigma_\mathcal{Y}^2}} = \frac{c_{XY}(\delta)}{\sigma_\mathcal{X} \cdot \sigma_\mathcal{Y}}, \tag{3}$$

where $\sigma_\mathcal{X}$ and $\sigma_\mathcal{Y}$ are the *standard deviations* of $\mathcal{X}$ and $\mathcal{Y}$.

Intuitively, the cross-correlation gives hints about the presence of correlation between two time series when time-shifted by the lag δ (i.e., *lagged relationship*).

In particular, when one or more $X_{t+\delta}$ are predictors of Y_t and $\delta < 0$, we say that X *leads* Y. Conversely, when one or more $X_{t+\delta}$ are predictors of Y_t and $\delta > 0$, we say that X *lags* Y (or, equivalently, Y *leads* X).

Many problems related to time series analysis deal with identifying which variable is leading and which is lagging. Some others assume a certain variable (X) to be leading of another one (Y), namely they aim to use the values of X to *predict* future values of Y.

However, it is worth remarking that cross-correlation is designed for stationary time series (at least in a weak sense). Indeed, estimating the cross-correlation between two non-stationary time series may lead to a misleading evaluation of their actual lagged relationship.

B. Time Series Regression

Regression analysis is one of the most powerful statistical tools for modeling relationships among variables. In its most general form, a *regression model* aims at relating a *dependent* variable Y to a parametric function of a set of *independent* variables (or *inputs*) $X_1, \ldots, X_r$.

The widest used is the *linear regression model*, which assumes that Y can be written as the sum of two terms. The first one is a deterministic component depending on

$X_1, \ldots, X_r$ and linear in the parameters. The second term represents a random error component including all the influent factors on Y that are not considered in the deterministic component. Using matrix notation, it can be written as follows:

$$Y = \mathbf{X}\boldsymbol{\beta} + \epsilon,$$

where $Y = (Y_1, \ldots, Y_k)^T$ is a random vector, $\mathbf{X} = (x_{i,j})$ is a full rank $k \times r$ matrix of observed values for $X_1, \ldots, X_r$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_r)^T$ is an unknown r-dimensional parameter and $\epsilon = (\epsilon_1, \ldots, \epsilon_k)^T$ is the error component that is assumed to have a multivariate normal distribution with zero mean and uncorrelated (thus independent) components, $\epsilon \sim \mathcal{N}_k(0, \sigma^2 I_k)$, with I_k the identity matrix of order k.

The most common technique for estimating the coefficients $\boldsymbol{\beta}$ is *Ordinary Least Squares* (OLS), where values $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_r)^T$ are chosen so as to *minimize* the residual sum of squared [17], [18]:

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}\{(Y - \mathbf{X}\boldsymbol{\beta})^T(Y - \mathbf{X}\boldsymbol{\beta})\}.$$

The resulting OLS estimator is thus computed as follows:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y.$$

The linear regression model can be fruitfully used for modeling relationships among time series. However, in this context new issues arise. In particular, one variable can influence another with a specific time lag. Furthermore, special care should be taken when dealing with non-stationary time series, since *spurious regression* may occur [19], [20].

In this work, we focus on two different classes of time series regression models that are briefly described below.

Autoregressive Models (AR). The simplest regression model for time series is the one relating a variable (Y_t) *only* to a linear combination of p of its lags ($Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$). We call it autoregressive model, which is typically denoted by AR(p) and where p is the *lag order* of the model:

$$Y_t = \alpha + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \epsilon_t. \quad (4)$$

Autoregressive Distributed Lag Models (ADL). Some analyses require using a regression model that has *both* lags of dependent and explanatory variables, or what we call autoregressive distributed lag model, denoted by ADL(p, q):

$$\begin{aligned} Y_t = \alpha + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \\ + \psi_1 X_t + \psi_2 X_{t-1} + \ldots + \psi_{q+1} X_{t-q} + \delta t + \epsilon_t. \end{aligned} \quad (5)$$

A right estimation and interpretation of an ADL(p, q) model depends on whether the time series variables Y_t and X_t are stationary or not. Indeed, if the variables are stationary the model parameters can be safely estimated by using OLS. Conversely, if Y_t and X_t are non-stationary OLS can lead to wrong parameters estimation, or what we have referred to as *spurious regression problem*.
Anyway, spurious regression vanishes if Y_t and X_t are *cointegrated* [21], and the OLS estimation still works fine even for non-stationary time series.

V. Twitter vs. Wikipedia Trending Entities

In this section, we discuss how we extract, analyze, and contrast trending entities, as observed in Twitter and Wikipedia.

A common way to automatically cross-reference text documents (like tweets) and Wikipedia is to use the latter as a resource for automatic keyword extraction and word sense disambiguation. More specifically, the whole set of Wikipedia articles can be seen as a set of unique and distinct *entities* $\mathcal{E} = \{e_1, \ldots, e_W\}$, where $|\mathcal{E}| = W$ is the total number of Wikipedia articles. We aim to use $\mathcal{E}$ as a common *vocabulary* not only in Wikipedia but also in Twitter, in order to identify time series associated with each entity in the two contexts.

Entity Linking in Twitter using Wikipedia. To identify the correct entities occurring in a tweet, we need to link mentions of those entities in the text with their referent entities in the knowledge base, i.e., Wikipedia in our case. To this end, we define a controlled vocabulary of *mentions* M_e, for each $e \in \mathcal{E}$ of Wikipedia. We build M_e by using the title of the Wikipedia article about entity e, along with the set of anchor texts of internal Wikipedia hyperlinks pointing to such article. We denote with M be the *vocabulary* of all the possible mentions of Wikipedia entities.
In general, given any two entities e and e', it holds that $M_e \cap M_{e'} \neq \emptyset$, and thus the same mention can be used as an anchor text to hyperlink distinct Wikipedia articles. Therefore, given a mention $m \in M$ detected in a document/tweet D, we may have a set of *candidate* entities $C_m = \{e \mid m \in M_e\} \subseteq \mathcal{E}$. The *Entity Linking Problem* aims to disambiguate such entity references: for each mention m discovered in D, we have to identify the correct entity $\hat{e} \in C_m$.
In Section VI we discuss the disambiguation technique we actually use for entity linking. Since we need to identify trending entities in a large corpus of tweets, a simple method suffices for our purposes. In addition, it is worth remarking that more sophisticated technique [2], [3] are not adequate for Twitter, since texts of tweets is too short.

Trending Entity Score. We refer to $\mathcal{T} = \langle t_1, t_2, \ldots, t_T \rangle$ as the sequence of T *discrete*, *equally-lasting*, and *equally-spaced* slots already defined in Section IV. We introduce two functions, s_X and s_Y, which assign *scores* to each entity in the vocabulary ($e \in \mathcal{E}$), as observed at each time slot in $\mathcal{T}$:

$$s_X : \mathcal{E} \times \mathcal{T} \longmapsto \mathbb{N},\ s_Y : \mathcal{E} \times \mathcal{T} \longmapsto \mathbb{N}.$$

For each entity, s_X and s_Y indicate the "strength" of its trending in a given time slot, as measured by Twitter and Wikipedia, respectively. We define the two following normalized integer scores, ranging from 0 to 100.

1) Twitter Trending Entity Score. Let $e_k \in \mathcal{E}$ be a trending entity, and let $count(e_k, t)$ be the number of occurrences of e_k in a sample of public tweets as observed during t. Then, we denote by $tes(e_k, t),\ t \in \mathcal{T}$ the *twitter entity score*, which is computed as follows:

$$tes(e_k, t) = \left\lceil \frac{count(e_k, t)}{\arg\max_{t \in \mathcal{T}} count(e_k, t)} \right\rceil * 100, \quad (6)$$

where $\arg\max_{t \in \mathcal{T}} count(e_k, t)$ is a normalization factor that evaluates to the maximum count of e_k over *all* the observations

in $\mathcal{T}$. Finally, we use the *twitter entity score* to evaluate the function s_X, i.e., $s_X(e_k, t) = tes(e_k, t),$ where $t = t_1, \ldots, t_T$.

2) Wikipedia Trending Entity Score. Let $e_k \in \mathcal{E}$ be a trending entity, and let $n\_reqs(e_k, t)$ be the number of requests for the Wikipedia article of e_k as measured during t. We compute the *wikipedia entity score*, denoted by $wes(e_k, t),\ t \in \mathcal{T}$, as follows:

$$wes(e_k, t) = \left\lceil \frac{n\_reqs(e_k, t)}{\arg\max_{t \in \mathcal{T}} n\_reqs(e_k, t)} \right\rceil * 100. \quad (7)$$

Again, $\arg\max_{t \in \mathcal{T}} n\_reqs(e_k, t)$ is a normalization factor that evaluates to the maximum number of requests for the Wikipedia article of e_k over *all* the observations in $\mathcal{T}$. Finally, we use the *wikipedia entity score* to evaluate the function s_Y, i.e., $s_Y(e_k, t) = wes(e_k, t),$ where $t = t_1, \ldots, t_T$.

Trending Entity Time Series. Now that we have clarified what we meant for trending entity score, we are ready to discuss how each trending entity is modeled by a *time series*. Indeed, we may think of each $e_k \in \mathcal{E}$ as a *pair* of time series, namely $\mathcal{X}_k = \{X_t\}_{t=t_1}^{t_T}$ derived from Twitter, and $\mathcal{Y}_k = \{Y_t\}_{t=t_1}^{t_T}$ derived from Wikipedia. Both $\mathcal{X}_k$ and $\mathcal{Y}_k$ are composed of t_T *random variables*, and each random variable evaluates to the Twitter and Wikipedia entity scores, respectively.
More formally, let $s_X(e_k, t)$ and $s_Y(e_k, t)$ be the Twitter and Wikipedia trending scores of $e_k \in \mathcal{E}$, as measured at time $t \in \mathcal{T}$. The pair of *observed time series* for $e_k \in \mathcal{E}$ correspond to the sequences of values assumed by each X_t and Y_t:

$$\mathcal{X}_k = \{X_t = s_X(e_k, t)\}_{t=t_1}^{t_T}, \quad \mathcal{Y}_k = \{Y_t = s_Y(e_k, t)\}_{t=t_1}^{t_T}.$$

Finally, among all the possible pairs of time series, in this paper we are only interested in comparing those who are actually referring to the *same* trending entities. In other words, we hereinafter consider a set of pairs $\mathcal{D} = \{(\mathcal{X}_i, \mathcal{Y}_j) \mid e_i = e_j\}$.

Trending Entity Forecasting and Causality. Our final goal is to check whether the temporal evolution of a trending entity from Twitter is *significant* to explain and predict the temporal progress of the requests made for accessing to the corresponding Wikipedia article. In order to validate our hypothesis, we perform the following tests for each $(\mathcal{X}_k, \mathcal{Y}_k) \in \mathcal{D}$.
Firstly, we evaluate the capability of the time series $\mathcal{X}_k$ (Twitter) in *forecasting* the time series $\mathcal{Y}_k$ (Wikipedia). Moreover, we measure the *causality* between $\mathcal{X}_k$ and $\mathcal{Y}_k$ by testing for the *Granger causality* [22].
To achieve both tasks, we compare the two classes of regression models discussed in Section IV-B. Eventually, we aim to show which model best fits our data, on average.

VI. Experiments and Results

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

(i) Raw Data Crawling: to collect both Twitter and Wikipedia data to derive the actual time series of trending entities.

(ii) Wikipedia Entity Linking: to link named-entity mentions occurring on Twitter with the corresponding set of Wikipedia articles.

(ii) Time Series Building: to create the actual time series from the "raw" Twitter and Wikipedia data which have been previously crawled.

(iv) Time Series Regression Analysis: to conduct time series regression analysis for exploring relation between Twitter and Wikipedia trending entities.

A. Raw Data Crawling

In the very first step, we crawled all the data both from Twitter and Wikipedia, which were necessary for deriving the final dataset of time series. We collected data for fifteen consecutive days, namely from 2012-11-01 at 00:00AM UTC to 2012-11-15 at 11:59PM UTC.

We deliberately chose this time range because we knew in advance that, at least a standing out event would have occurred, namely the U.S. 2012 Presidential Elections. However, since Twitter and Wikipedia have their own services and access policies for retrieving data, we describe this task separately.

Twitter Public Tweets. Twitter allows developers 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*, each one having its own policies to limit the rate of requests that a client is allowed to perform.
We used the Streaming API in order to retrieve a sample of the public tweets nearly real-time. In addition, we upgraded the default access policy of Streaming API to *gardenhose* level to avoid the standard limit of API calls. In this way, the sample of Twitter timelines we crawled was randomly-selected from a large collection of about 10% of the whole public tweets (instead of 1%).
We thus focused only on tweets coming from the U.S., which hopefully were almost all written in English. As a result, we obtained a total corpus of about 260 million tweets.

B. Wikipedia Entity Linking

In order to extract the set of *trending entities* from this huge Twitter dataset, we exploited the Wikipedia 04/03/2013 dump[7] and applied the following multi-step technique:

1) For each hourly time slot, we considered all the tweets posted in the meanwhile. For each tweet, we extracted all the possible n-grams, $n = 1, \ldots, 6$, and we looked-up for them in the controlled vocabulary of mentions M. For each detected mention m, we identified the set of candidate entities $C_m \subseteq \mathcal{E}$.

2) We limited the set of detected mentions (and associated candidate entities) to the most meaningful ones. To this end, we exploited the *link probability* of a mention m, denoted by $LP(m)$, which is defined as the number of times m occurs as an anchor text in Wikipedia divided by its total number of occurrences in all the Wikipedia pages [1]. This property permits to discriminate mentions that refers with a high probability to some entity from those which may refer to an entity only in particular contexts. For instance, the mention "the" occurs a huge number of times in Wikipedia, but only in a few cases it is used as an anchor text to the

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

[7] http://dumps.wikimedia.org/enwiki/20130403/enwiki-20130403-pages-articles.xml.bz2

`English articles` entity. Thereby, we added m to the detected mentions only if $LP(m) > 0.4$.

3) At this stage, we had to link a single entity to each detected mention m. To this end, we sorted C_m using the *commonness* (i.e., prior probability) of each candidate $e \in C_m$. The *commonness* of e, denoted by $CP(e)$, is defined as the ratio between the number of times m is used as an anchor text to actually refer to e, and the total number of times m is used as an anchor in Wikipedia [2].

4) Once detected the set of all the entities appearing in our collection of tweets, we counted the number of times each entity was mentioned in the corpus on each hourly time slot. Finally, we considered the top-50 most frequent entities on each hour, and we obtained our running vocabulary of trending entities $\hat{\mathcal{E}} \subseteq \mathcal{E}$, namely $1,280$ unique entities.

Wikipedia Page Statistics. Wikipedia provides a lot of statistical tools both via its official Wikimedia Foundation analytics team and through third-party volunteers. For the sake of our purpose, we used the standard page view statistics for Wikimedia projects[8], which contained raw page access data for *all* Wikipedia projects. We chose this option because, to the best of our knowledge, it was the only that allowed us to get hourly page access data. In fact, other available statistics concern only daily-aggregated data[9].
We downloaded the set of 360 page statistics files, one for each hour within the fifteen days of observations. Each hourly file contained one line for each page request statistics of that hour, and each line looks like the following real, four-column, white-spaced record:

```
en Barack_Obama 2599 248007182
```

The record above traces data about the requests for the English Wikipedia article (1st column) of Barack Obama (2nd column), which was accessed 2599 times during that hour (3rd column), and required the total transferring of about 250MB of contents (4th column).

C. Time Series Building

In this section, we discuss how actual time series of trending entities were built from the raw datasets collected from Twitter and Wikipedia, as detailed above.

To this end, we computed the trending entity scores as measured by Twitter (s_X) and Wikipedia (s_Y). In a nutshell, for each entry e_k in our vocabulary of trending entities $\hat{\mathcal{E}}$, we computed both its Twitter and Wikipedia entity scores, as discussed in Eq. 6 and Eq. 7. More precisely, to generate the Twitter time series associated with e_k we used the normalized hourly count of mentions of e_k occurring in the corpus of crawled tweets. In order to built the Wikipedia time series for e_k we instead retrieve the normalized hourly number of requests for the Wikipedia article of e_k, as provided by the Wikipedia page statistics.

Since our analysis spanned fifteen days, the total number of hourly observations turned out to be $24*15 = 360$, namely the sequence of time slots $\mathcal{T}$ had exactly 360 intervals. Therefore, we came up with the following pair of time series for each $e_k \in \hat{\mathcal{E}}$:

$$\mathcal{X}_k = \{X_t = s_X(e_k, t)\}_{t=t_1}^{t_{360}}, \quad \mathcal{Y}_k = \{Y_t = s_Y(e_k, t)\}_{t=t_1}^{t_{360}}.$$

Finally, we focused on the set of pairs of time series $\mathcal{D} = \{(\mathcal{X}_i, \mathcal{Y}_j) \mid e_i = e_j\}$, as specified in Section V. The following section is completely devoted to test and assess any relation between Twitter and Wikipedia time series of trending entities.

D. Time Series Regression Analysis

Our experiments comprised the following steps: *(i) Test for weak stationarity*, *(ii) Cross-correlation*, and *(iii) Forecasting and Causality*.

Test for Weak Stationarity. First, we tested if time series on our dataset were stationary or not. To this end, we inspected the *autocorrelation* of each individual time series $\mathcal{X}_i$ and $\mathcal{Y}_j$, separately. Indeed, the autocorrelation of a non-stationary variable appears *strongly positive* and *non-noisy* out to a high number of lags (often 10 or more) meaning it is slow to decay. Conversely, the autocorrelation of a stationary variable usually decays into "noise" (e.g., fluctuating behavior) and/or hits negative values within a few lags.
According to this, all the time series in $\mathcal{D}$ turned out to be stationary. This is reasonable considering that our time series have both lower and upper bounds set to 0 and 100, respectively, thereby no *trending*[10] (either increasing or decreasing) nor *seasonality* could occur.
Fig. 2 shows the autocorrelation plots for the two time series of the trending entity Adam Levine, which were derived from Twitter (a) and Wikipedia (b), using up to 24-hour lags.

Cross-correlation. An immediate way to discover the relationship between two time series is to measure their *cross-correlation* (see Section IV-A). Intuitively, the cross-correlation provides hints about the presence of correlation between the random variables associated with two time series at a given time lag (i.e., *lagged relationship*).
In addition, the test for stationarity we conducted above revealed that all the time series in our dataset were at least weak stationary. Thereby, this guaranteed that cross-correlation could be safely computed to estimate the lagged relationships between our time series.

We indeed computed the cross-correlation of each pair of time series $(\mathcal{X}_k, \mathcal{Y}_k) \in \mathcal{D}$, according to the Eq. 3. We used several lags δ (i.e., $\delta = \pm 3, \pm 6, \pm 12, \ldots$) in order to capture lagged relationships from few hours up to many days. However, the most interesting results were obtained when we searched for cross-correlation within 12 hours. After that lag, the cross-correlation became generally not significant. In fact, the *maximum* values of cross-correlation were mostly obtained at lag $\delta = -1$, and just within two or three lags they suddenly dropped below the level of significance. To give a better idea of this result, Fig. 2 (c) presents the cross-correlation plot for the two time series from Twitter and Wikipedia associated with the trending entity Adam Levine.
Fig. 3 shows how maximum cross-correlation values computed for *all* the time series in $\mathcal{D}$ were distributed over the hourly lags. From this last plot, more than 40% of the total pairs of

[8] http://dumps.wikimedia.org/other/pagecounts-raw/
[9] http://toolserver.org/~emw/wikistats/

[10] Here the term "*trending*" refers to a characteristic of time series, and *not* to our trends.

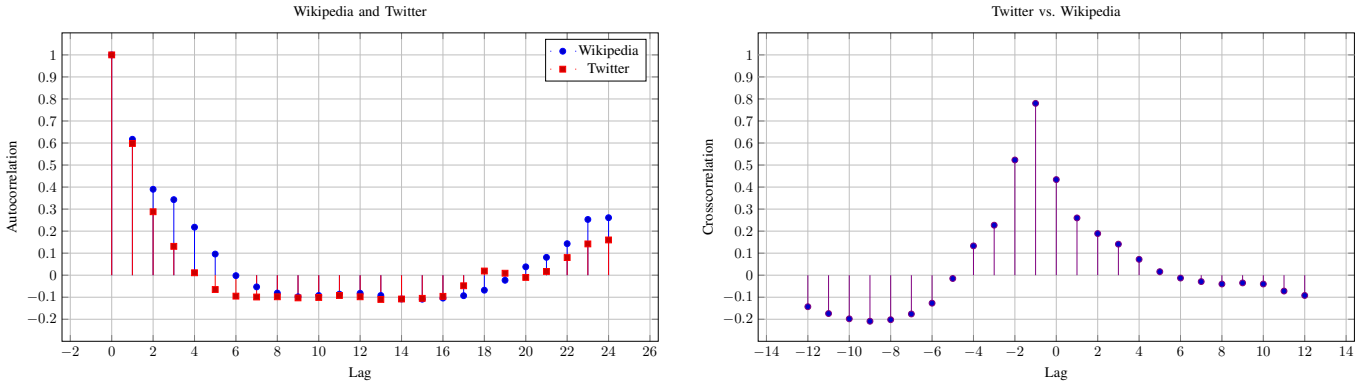

Fig. 2. Autocorrelation (a), and Cross-correlation plots (b) of the time series for the trending entity Adam Levine

time series in $\mathcal{D}$ have their maximum correlation at lag $\delta = -1$. In addition, about two out of three maximum correlation values occurred at non-positive lags. This means that trending entities derived from Twitter actually anticipated the volumes of requests that users made for the corresponding Wikipedia articles, namely Twitter *leads* Wikipedia.

Finally, it is worth noting that considerations above are compliant with our preliminary findings described in Section II.

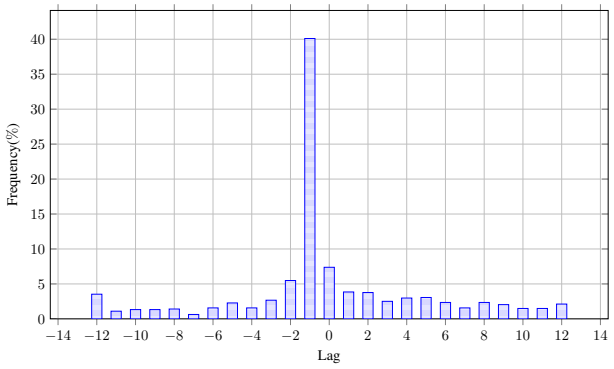

Fig. 3. Maximum cross-correlation distributed over hourly lags.

Forecasting and Causality. From the discussion on cross-correlation in the previous paragraph, we realized that a strong lagged relationship exists between the time series of trending entities as derived from Twitter and Wikipedia. More precisely, we found that Twitter *leads* Wikipedia.

In this last experimental stage, we aimed to evaluate other two phenomena arising from the result above: *(i) forecasting* and *(ii) causality*. The former refers to the power of Twitter in *predicting* access to Wikipedia articles whereas the latter goes a step further and tries to determine *causality* between Twitter and Wikipedia by performing a *Granger-causality* test [22]. Both issues required dealing with the time series regression models described in Section IV-B.

To assess the first aspect, we examined each pair of time series $(\mathcal{X}_k, \mathcal{Y}_k)$ in our running dataset $\mathcal{D}$. We started fitting each series to *autoregressive* models, i.e., AR(p), which we considered as our *baseline* models. Intuitively, this means that we were explaining the hourly volume of requests for a Wikipedia article $Y_{k,t}$, by only considering the values of the *same* series as measured up to p hours before (i.e., $Y_{k,t-1}, \ldots, Y_{k,t-p}$). Thereby, these models assumed Wikipedia trending entities were dependent *only* on themselves, and Twitter having no influence at all.

In fact, to assess how lagged values of Twitter time series actually help forecast Wikipedia time series, we exploited the third class of regression models introduced in Section IV-B, namely *autoregressive distributed lag* ADL(p, q). As opposed to AR(p), ADL(p, q) tries to fit a Wikipedia time series $Y_{k,t}$ using up to p past values of the same series (i.e., $Y_{k,t-1}, \ldots, Y_{k,t-p}$) *and* up to q past values of the paired time series from Twitter (i.e., $X_{k,t}, \ldots X_{k,t-q}$).

In order to evaluate how Twitter is able to forecast Wikipedia, we computed how many ADL(p, q) models retained as *significant* their q-lagged component. For instance, if we focus on ADL($1, 1$) we found that the lag-1 component of Twitter turned out to be significant 55.9% of times. As expected, this percentage decreased as the number of considered q lags was growing. This measure gave evidence that, in the more than half of the cases, lagged values of Twitter are actually useful to model our time series.

Significance was determined by computing the *p-value*,[11] which is the probability of observing a test statistic at least as large as the one calculated assuming the *null hypothesis* is true. To choose whether the null hypothesis is in fact true or false, a trade-off on the the *p-value* is needed. Typically, the null hypothesis is rejected if *p-value* is below a *significance level* α, which, for our experiments, was set to 0.05. In our case, the null hypothesis was that the q-lagged component was *not* significant. Thereby, rejecting the null hypothesis meant to consider such coefficients useful for the model to fit the data.

In addition, we compared the ability of AR(p) and ADL(p, q) models in fitting our data. In fact, we compared their *adjusted coefficient of determination*, denoted by $R^2 \in [0, 1]$ and averaged by all the pairs of time series. This index 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. The greater R^2 for a model the better it fits the data. Actually, R^2 considers the number of explanatory terms in the model, so that it increases only if the newly introduced term improves the model more than would be expected by chance. From the results shown in Tab. I, ADL(p, q) models always outperformed AR(p).

As the last contribution, we checked if there was any causality between Twitter and Wikipedia trending entities. This was achieved by running a test for *Granger-causality* [22], which roughly states the following: given two time series $\mathcal{X}$ and $\mathcal{Y}$, if lagged values of $\mathcal{X}$ help (i.e., are significant to) predict current values of $\mathcal{Y}$ in a forecast built from lagged values of both $\mathcal{X}$ and $\mathcal{Y}$ (i.e., an ADL(p, q) model), then $\mathcal{X}$ is said to "granger cause" $\mathcal{Y}$. Since causality can be bi-directional,

[11] Here the p of *p-value* is totally unrelated to the lag order p of autoregressive models.

TABLE I. TIME SERIES REGRESSION: AR(p) VS. ADL(p, q).

AR(p)		ADL(p, q)	
p	R^2	p, q	R^2
1	0.39	1, 1	0.50
2	0.41	2, 2	0.52
3	0.41	3, 3	**0.53**
4	**0.42**	4, 4	0.53
5	0.42	5, 5	0.53
6	0.42	6, 6	0.53

namely both $\mathcal{X}$ can cause $\mathcal{Y}$ and vice versa, we must check for unidirectional causality to conclude that one of the two actually "came first" [**?**]. In other words, given that we wanted to test if only Twitter ($\mathcal{X}_k$) "granger-causes" Wikipedia ($\mathcal{Y}_k$), we had to reject the non-causality of $\mathcal{X}_k$ to $\mathcal{Y}_k$, and accept the non-causality for the opposite direction. To this end, we computed the number of times Twitter caused Wikipedia among all those ADL(p, q) models having their q-lagged component significant. The outcomes of this is presented in Tab. **??**, which shows that up to 61.9% of the ADL(p, q) models with significant q-lagged component identifies also time series where Twitter cause Wikipedia.

VII. CONCLUSION AND FUTURE WORK

In this work we provided the following contributions. First, we presented an *entity linking* technique to recognize mentions of named-entities as of Wikipedia articles appearing on a huge corpus of real-world user tweets from Twitter.

Then, we focused on the top-most frequent hourly entities, which we called *trending entities*. Therefore, we claimed that the "popularity" of a trending entity on Twitter may help predict the number of requests for the corresponding Wikipedia article. The rationale of this intuition is that information spreading nearly real-time over the Twitter social network could anticipate the set of topics that users will be interested in – and thereby will look up on Wikipedia – in the next future.

To validate our claim, we conducted an extensive time series regression analysis, where time series were derived from the real-world set of trending entities, as observed during fifteen consecutive days.

Thus, we assessed the *forecasting* power of Twitter by finding that the models that use Wikipedia as the *dependent* variable and Twitter as the *explanatory* variable retain as significant the past values of Twitter 60% of times. Moreover, we discovered that a trending entity on Twitter *causes* a similar Google trend to later occur about 43% of times. Eventually, we showed that the best-performing models are those using past values of *both* Twitter and Wikipedia.

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

REFERENCES

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

[2] D. Milne and I. H. Witten, "Learning to link with wikipedia," in *CIKM '08*. New York, NY, USA: ACM, 2008, pp. 509–518.

[3] X. Han, L. Sun, and J. Zhao, "Collective entity linking in web text: a graph-based method," in *SIGIR '11*. New York, NY, USA: ACM, 2011, pp. 765–774.

[4] 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 *SIGIR '12*. New York, NY, USA: ACM, 2012, pp. 721–730.

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

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

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

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

[9] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," *JASIST*, vol. 60, no. 11, pp. 2169–2188, November 2009.

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

[11] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis, "Bieber no more: First Story Detection using Twitter and Wikipedia," *SIGIR 2012 Workshop on Time-aware Information Access (#TAIA2012)*, 2012.

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

[13] 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.

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

[15] 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.

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

[17] M. H. Kutner, C. J. Nachtsheim, and J. Neter, *Applied Linear Regression Models*, Fourth International ed. McGraw-Hill/Irwin, September 2004.

[18] N. Ravishanker and D. Dey, *A First Course in Linear Model Theory*, ser. Chapman & Hall/CRC Texts in Statistical Science. CRC PressINC, 2013.

[19] G. U. Yule, "Why do we Sometimes get Nonsense-Correlations between Time-Series?–A Study in Sampling and the Nature of Time-Series," *Journal of the Royal Statistical Society*, vol. 89, no. 1, pp. 1–63, January 1926.

[20] C. W. J. Granger, "Some properties of time series data and their use in econometric model specification," *Journal of Econometrics*, vol. 16, no. 1, pp. 121–130, May 1981.

[21] R. F. Engle and C. W. J. Granger, "Co-integration and error correction: Representation, estimation, and testing," *Econometrica*, vol. 55, no. 2, pp. 251–76, March 1987.

[22] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, July 1969.