Twitter Anticipates Bursts of Requests for Wikipedia Articles

Gabriele Tolomei, Salvatore Orlando
DAIS - Università Ca' Foscari, Venice, Italy
{gabriele.tolomei,orlando}@unive

Diego Ceccarelli, Claudio Lucchese
I.S.T.I. "A. Faedo", CNR, Pisa, Italy
{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. 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. To validate this claim, we apply an EL technique to extract trending entities from a large 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 Twitter actually *leads* Wikipedia by one or more hours.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

Keywords

Entity Linking; Twitter; Wikipedia; Time Series Analysis

1. INTRODUCTION

In the last years research community involved in the social mining field has started studying the relationship between Twitter[1] and Wikipedia[2], as well as between Twitter and other online digital resources. Osborne et al. [7] discuss how Wikipedia can be exploited to filter out spurious real-time events detected on Twitter. Ruiz *et al.* [9] study the problem of correlating microblogging activity from Twitter with stock market events. De Francisci Morales *et al.* [2] recommend interesting news to users by exploiting the information in their Twitter persona. Giummolè *et al.* [3] study the relationships between Twitter trending topics and Google hot queries.

In this paper we aim to study how Twitter and Wikipedia are related by exploiting *named-entities* (such as person names, places, etc.), which are mentioned in user tweets and may become "extraordinary popular". We adopt a simple *Entity Linking* (EL) technique to detect such trending entities in Twitter from their *mentions* (i.e., small fragments of text referring to any named-entity in a knowledge base). More precisely, we use Wikipedia as the referring knowledge base of entities and associated mentions. EL is generally a challenging task, and is even harder when mentions appear in very short texts with not enough surrounding context, such as tweets. The first system to use Wikipedia for entity linking was *Wikify!* [5], while Milne and Witten [6] largely improved this first solution. Since entity relatedness has been recognized as the most important feature to disambiguate entity-linking, Ceccarelli *et al.* [1] discuss how an effective relatedness measure can be learnt from large training sets using a learning-to-rank approach.

The goal of this preliminary work is to investigate whether any relationship exists between trending entities as extracted from Twitter and the request volumes 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 occur later. 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.

2. TIME RELATION BETWEEN TWITTER AND WIKIPEDIA

To motivate our work, we present a pair of real-world examples of *trending entities*, i.e., entities frequently mentioned in user tweets and the corresponding access volumes of the Wikipedia articles associated with those entities. Each plot in Fig. 1 shows a pair of *time series*, one related to Twitter and the other to Wikipedia. The observed values of the time series are the (normalized) hourly *trending score* (i.e., popularity) of the two entities in the first two weeks

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

DUBMOD'13, October 28 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2513577.2513580 ...$15.00.

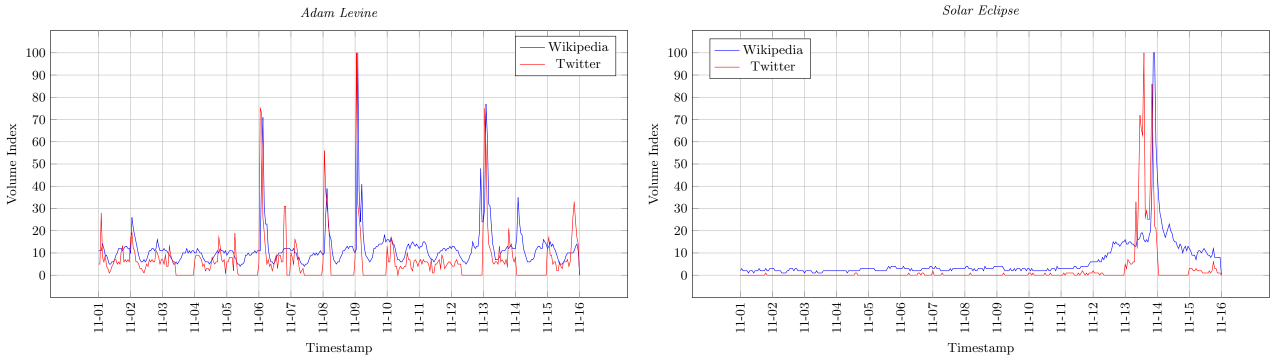

Figure 1: Time series plots of trending entity scores as measured by Twitter and Wikipedia.

of November 2012. The plot on the left shows a pair of time series about the entity Adam Levine[3], who is a famous American singer. 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 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 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).

3. 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 recognize 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 4 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 techniques [6, 4] are not adequate for Twitter, since texts of tweets are 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, used to build pairs of *time series*.

In particular, 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 $\mathcal{T}$: $s_X : \mathcal{E} \times \mathcal{T} \longmapsto \mathbb{N}$ and $s_Y : \mathcal{E} \times \mathcal{T} \longmapsto \mathbb{N}$. 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 *normalized twitter entity score*, which is computed as follows:

$$tes(e_k, t) = \left\lceil \frac{count(e_k, t)}{\max_{t \in \mathcal{T}} count(e_k, t)} \right\rceil * 100, \qquad (1)$$

where $\max_{t \in \mathcal{T}} count(e_k, t)$ 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 *normalized wikipedia entity score*, denoted by

[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.

$wes(e_k, t),\ t \in \mathcal{T}$, as follows:

$$wes(e_k, t) = \left\lceil \frac{n\_reqs(e_k, t)}{\max_{t \in \mathcal{T}} n\_reqs(e_k, t)} \right\rceil * 100. \quad (2)$$

Again, $\max_{t \in \mathcal{T}} n\_reqs(e_k, t)$ 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. We may finally associate with each $e_k \in \mathcal{E}$ 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:

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

4. EXPERIMENTS AND RESULTS

In this section, we describe the experimental setup and the tests conducted on real-world datasets of trending entities from Twitter and Wikipedia.

Raw Twitter Data Crawling.

We collect Twitter data for 15 consecutive days, namely from 2012-11-01 at 00:00AM UTC to 2012-11-15 at 11:59PM UTC, during which (at least) a standing out event occurred, namely the U.S. 2012 Presidential Elections. We use the Twitter Streaming API upgraded to *gardenhose* level, in order to retrieve nearly real-time a sample of 10% of the public tweets[6]. We focus only on tweets coming from the U.S., which hopefully are almost all written in English. As a result, we obtain a total corpus of about 260 million tweets.

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

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

2) We limit the set of detected mentions (and associated candidate entities) to the most meaningful ones. To this end, we exploit 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 [5]. This property permits us to discriminate mentions that refers with a high probability to some entity from those referring to an entity only occasionally. For example, 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 English articles entity. Thereby, we add m to the detected mentions only if $LP(m) > 0.4$.

3) At this stage, we have to link a single entity to each detected mention m. To this end, we sort 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 [6].

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

Wikipedia Page Statistics and Time Series Building. In order to collect statistics about the hourly volumes of requests for Wikipedia articles during the relevant period (the first 15 days of November 2012), we use the standard page view statistics for Wikimedia project[8]. In a nutshell, for each article and each hour, we collect a record that states both the total number of access counts and the total amount of MBs transferred from Wikipedia servers to clients.

For each trending entities $e_k \in \hat{\mathcal{E}}$ discovered in Twitter, we can finally build the two time series made of $24 * 15 = 360$ observations, $\mathcal{X}_k$ and $\mathcal{Y}_k$.

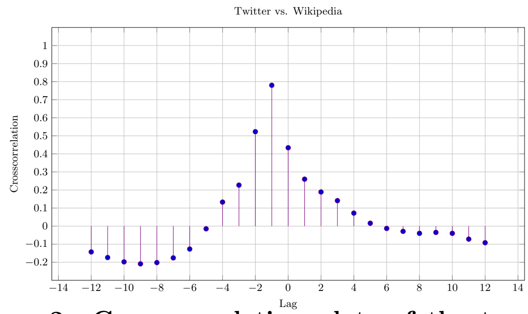

Figure 2: Cross-correlation plots of the two time series relating to the entity Adam Levine.

4.1 Time Series Analysis

We analyze our time series pairs by computing their *cross-correlation*, which we use to show that the Twitter series are predictor of the Wikipedia ones. Let $t \in \mathcal{T} = \langle t_1, t_2, \ldots, t_T \rangle$ be a sequence of T *discrete*, *equally-lasting*, and *equally-spaced* slots, and let δ be a *time lag* δ, such that $t + \delta \in \mathcal{T}$. We first define the *cross-covariance* as: $c_{XY}(\delta) = E[(X_{t+\delta} - \mu_{\mathcal{X}})(Y_t - \mu_{\mathcal{Y}})]$. The *cross-correlation* is the cross-covariance normalized in the range $[-1, 1]$, that is:

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

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.

However, cross-correlation can be safely computed only when the time series are at least *weak stationary* [8]. In fact, measuring the cross-correlation between two non-stationary time series generally leads to wrong conclusion about their actual relation. Other than statistical tests (e.g., *ADF*, *KPSS*), an empirical way to check for (weak) stationarity is to inspect the *autocorrelation* plots of each individual time series $\mathcal{X}_i$ and $\mathcal{Y}_j$, separately. The autocorrelation of a non-stationary variable appears *strongly positive* and *non-noisy*

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

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

[8] http://dumps.wikimedia.org/other/pagecounts-raw/

out to a high number of lags (often 10 or more) meaning it decays slowly. Conversely, the autocorrelation of a stationary variable usually decays into "noise" (e.g., fluctuating behavior) and/or hits negative values within a few lags. We observe this last behavior in all our time series, which thus can be considered weak stationary.

Therefore, we compute the cross-correlation of each pair of time series $(\mathcal{X}_k, \mathcal{Y}_k)$, according to the Eq. 3. We use several lags δ (i.e., $\delta = \pm 1, \pm 2, \pm 3, \ldots$) in order to capture lagged relationships from few hours up to many days. However, the most interesting results are obtained when we search for cross-correlation within 12 hours. After that lag, the cross-correlation becomes generally not significant. In fact, the *maximum* values of cross-correlation are mostly obtained at lag $\delta = -1$, and just within few lags they suddenly drop below the level of significance. To better explain this result, Fig. 2 presents the cross-correlation plot for the two time series from Twitter and Wikipedia associated with the entity Adam Levine. Fig. 3 shows how maximum cross-correlation values computed for *all* our time series are distributed over the hourly lags. From this last plot, more than 40% of the total pairs of time series have their maximum correlation at lag $\delta = -1$. In addition, about two out of three maximum correlation values occur at non-positive lags. This means that trending entities derived from Twitter actually anticipate the volumes of requests that users make for the corresponding Wikipedia articles, namely Twitter *leads* Wikipedia.

Interestingly, the considerations above are fully compliant with our preliminary findings described in Section 2.

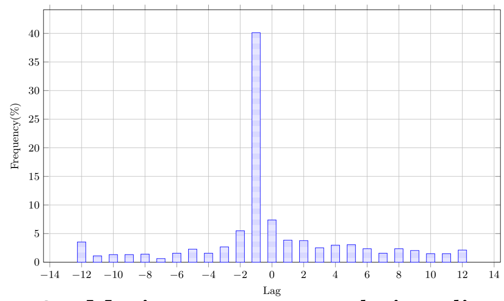

Figure 3: Maximum cross-correlation distributed over hourly lags.

	Lags (δ)								
	-4	-3	-2	-1	0	1	2	3	4
mean	0.16	0.28	0.45	**0.60**	0.35	0.27	0.29	0.23	0.23
stdev	0.18	0.25	0.28	0.27	0.25	0.19	0.21	0.17	0.14

Table 1: Statistics on Cross-correlation.

As the last step, we analyze the values of the maximum cross-correlation we observed for each pair of time series, on the basis of which we built the histograms in Figure 3. Table 1 shows the mean and the standard deviation of these values. First, we observe that at lag $\delta = -1$ we have pretty large correlation values ($mean = 0.60$) as expected. Cross-correlation is still large at lag $\delta = -2$. Second, even if a maximum cross-correlation is observed between pair of time series at different lags δ, its average value becomes increasingly lower ($mean < 0.40$). So the corresponding time series do not appear to be highly correlated. In the next and conclusive section, we give an anecdotal evidence of the possible rationale of this behavior, and we discuss how this opens for possible future work.

5. CONCLUSION AND FUTURE WORK

In this work, we discussed if *trending entities* rising from Twitter may predict the volume of requests for relating Wikipedia articles. To validate this claim, we provided the following contributions. First, we applied an *entity linking* (EL) technique to extract trending entities from a real-world dataset of public tweets. Then, we analyzed 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 revealed that Twitter actually *leads* Wikipedia by a lag of one hour, for more than 40% of the times.

In addition, we manually checked those cases where we observed a poor correlation. Remarkably, we noticed that most of the times this happened because the trending mention of an entity on Twitter is difficult to disambiguate. Indeed, the EL step mapped this trending mention to the *wrong* Wikipedia article/entity. For instance, the mention *Jim Jones* was linked to the Wikipedia article about Jim Jones[9] – a religious leader who founded the "Peoples Temple" – because it has the highest *commonness* (see Section 4). In fact, the correct entity should be the Wikipedia article on another Jim Jones[10] – a rapper and actor. Evidence of this mismatching could be found by looking at the statistics of the two articles limited to our period of observations, as well as directly from the true Wikipedia entity page. This last finding suggested that statistics on Wikipedia page requests might be useful for disambiguating entities, especially when mentions of those occur in short texts with not enough surrounding context, such as tweets. We left this new research challenge as future work.

Acknowledgements

This work was partially supported by the EU projects InGeoCLOUDS (no. 297300), MIDAS (no. 318786), E-CLOUD (no. 325091), the National project PON TETRis (no. PON01 00451), and the Regional project SECURE! (FESR PorCreo 2007-2011).

6. REFERENCES

[1] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Learning relatedness measures for entity linking. In *CIKM*, 2013. ACM.

[2] G. De Francisci Morales and C. Lucchese. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *WSDM*, 2012. ACM.

[3] F. Giummolè, S. Orlando, and G. Tolomei. Trending topics on Twitter improve the prediction of Google hot queries. In *SocialCom*, 2013. ASE/IEEE.

[4] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *SIGIR '11*, p. 765–774, 2011. ACM.

[5] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, p. 233–242, 2007. ACM.

[6] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, p. 509–518, 2008. ACM.

[7] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis. Bieber no more: First Story Detection using Twitter and Wikipedia. *SIGIR 2012 Workshop TAIA*, 2012.

[8] M. B. Priestley. *Spectral analysis and time series*. 1981. Academic Press.

[9] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes. Correlating financial time series with micro-blogging activity. In *WSDM*, p. 513–522, 2012. ACM.

[9] http://en.wikipedia.org/wiki/Jim_Jones

[10] http://en.wikipedia.org/wiki/Jim_Jones_(rapper)