# Hybrid Models for Future Event Prediction

Giuseppe Amodeo*
IASI-CNR, Univ. of L'Aquila
Rome, Italy
gamodeo@fub.it

Roi Blanco
Yahoo! Research
Barcelona, Spain
roi@yahoo-inc.com

Ulf Brefeld
Yahoo! Research
Barcelona, Spain
brefeld@yahoo-inc.com

## Abstract

We present a hybrid method to turn off-the-shelf information retrieval (IR) systems into future event predictors. Given a query, a time series model is trained on the publication dates of the retrieved documents to capture trends and periodicity of the associated events. The periodicity of historic data is used to estimate a probabilistic model to predict future bursts. Finally, a hybrid model is obtained by intertwining the probabilistic and the time-series model. Our empirical results on the New York Times corpus show that autocorrelation functions of time-series suffice to classify queries accurately and that our hybrid models lead to more accurate future event predictions than baseline competitors.

## Categories and Subject Descriptors

H.3.0 [**Information Storage and Retrieval**]: General; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

## General Terms

Theory, Experimentation

## Keywords

Future prediction, information retrieval, event prediction, time series, ARMA, ARIMA, SARIMA, web search

## 1. INTRODUCTION

Predicting the future is one of the oldest goals of mankind. While early approaches were assembled from heuristics and introspection, modern data repositories and information retrieval systems provide means for data-driven model generation and their quantitative analysis.

*This work was performed during an internship at Yahoo! Research and is partially supported by the EU Large Scale Integrated Project LivingKnowledge (contract no. 231126).


*CIKM'11,* October 24–28, 2011, Glasgow, Scotland, UK.


In contrast to previously published query volume based approaches [9, 7, 8], we study content-based models of time. We focus on time series that consist of user generated content retrieved according to a user query. That is, we are not only able to predict *when* a topic will be important in the future but also contribute to *why* this will be the case by analyzing the retrieved content. Additionally, our approach is independent of external resources and can be deployed to any document collection without the need of logging huge amounts of query traffic to perform predictions.

A natural way to deal with time series are the autoregressive moving average (ARMA) models and their derivatives that capture *periodicity* and *trend* of the data. The two criteria are fundamental for our analysis: periodicity determines whether predicting future events is an appropriate means for the actual time series as non-periodic or even random events such as natural disasters are not predictable by definition. The trend of a time series plays an important role as a continuously increasing amount of publications does not automatically imply the dawn of an upcoming event but could also be the result of a growing public interest, as for instance the query *global warming*.

The future event prediction consists of 3 stages. Each time series is classified using state-of-the-art machine learning techniques into four categories *periodic*, *partially-periodic*, *trend-based*, and *random* solely on the basis of its autocorrelation function. For periodic and partially-periodic time series, we adapt a probabilistic model to the periodicity of the autocorrelation function. A hybrid model is then computed by intertwining the probabilistic model with the original time series model for future event prediction.

We empirically evaluate all steps carefully using the New York Times corpus. Our results show that the classification of the time series can be accomplished with high accuracy and that the probabilistic model captures the regularities of periodic and partially-periodic time series very well. We further observe that the future time series predicted by the hybrid model are close to the ground-truth.

The remainder of our paper is structured as follows. Section 2 reviews related work. We present the hybrid model in Section 3 and report on empirical results in Section 4. Section 5 concludes the paper.

## 2. RELATED WORK

Goel et al. [7, 8] track unemployment levels and home sales by examining the number of times related queries have been submitted to a search engine and Choi and Varian [9] study the effects of query volume on future automobile,

home, and retail sales, as well as travel behavior. Chien et al. [5] study semantically related queries based on their temporal correlation. Gamon et al. [6] study the relationship and information flow between news platforms and Radinsky et al. [12] study term co-occurrences to predict upcoming trends. Recent work by Alonso et al. [2] and Catizone et al. [4] suggests that the time dimension can be further exploited by automatically creating time-lines from temporal information extracted from documents. In fact, some applications already make use of the temporal aspect to enhance search result presentation.

Time-series analysis has been adopted also to discover existing relationship among queries starting from the query logs. Zhang et al. [14] present a survey on the use of time series analysis for query logs. Kulkarni et al. [10] extract features for classifying queries according to their popularity and over time and Murata et al. [11] focus on temporal changes in search behavior. Finally, Adar et al. [1] analyzed the correlation of user behavior and external events. Contrarily to these approaches, we solely rely on the contents of the documents to generate the time series.

# 3. HYBRID MODELS FOR PREDICTING FUTURE EVENTS

In this section, we present our hybrid approach to future event prediction where we define an event as *an external and not directly observable incident, influencing the common interest in a topic*. Indicators of events are therefore bursts in the time series or, in other words, peaks. That is, from a practical point, peak prediction and event prediction are identical. Nevertheless, we keep the distinction because events act like latent variables that alter the time series at certain points in time which we then observe as peaks in the data.

Our approach comprises the following steps: Given an indexed document collection, we retrieve a set of documents related to a user query and translate the retrieved content into a time series. The resulting time series is classified to determine whether event prediction is an appropriate means for processing the query. If the time series is not rejected, a peak detection is performed and a probabilistic model is trained to predict future bursts that match our definition of events. Finally, we compute a hybrid model that combines the predicted events with the time series.

## 3.1 From Queries to Time Series

Given an input query, we retrieve the set of sentences that contain mentions of the query terms. Sentences are ranked using BM25 [13]. Finally, we bin the retrieved sentences using their publication dates. The size of the bins depends on the granularity of the data and could for instance be day-wise, week-wise or month-wise. The resulting histogram is our time series $y = y_1, y_2, \ldots, y_T$ that consists of counts of the retrieved documents in the respective time-bins.

## 3.2 Time Series Classification

Obviously, not all queries are suitable for event prediction. To decide whether predicting events is useful for a given time series $y$, we classify $y$ into four categories: (i) *periodic* time series exhibit a general, repetitive pattern, (ii) *partially periodic* time series possess some repetitive regularities which are however not as striking as in the previous case, (iii) *trend-based* time series follow a general increasing or decreasing trend (e.g., 'global warming') which is also the main characteristic, and finally (iv) *random* time series do not exhibit any regularities or structure that could be exploited. If a query is classified as periodic or partially-periodic it is further processed, otherwise discarded.

## 3.3 Historic Peak Detection

The previous section showed that we can derive high-level information from the time series. For event prediction, however, we need to be able to detect bursts of interest. Such information needs are reflected by peaks in the time series. A peak can be defined as follows:

DEFINITION 1 (PEAK). *Let $y$ be a time series and $g$ a function of $y$. An element $y_t$ is called a peak with respect to $g$, if (a) and (b) hold:*
*(a) $y_t$ is a local maximum.*
*(b) $y_t \geq g(y)$.*

The function $g$ realizes a thresholding and guarantees that only popular and isolated peaks are returned. Possible choices are for instance the mean, the median, or a quartile function.

## 3.4 Future Peak Prediction

The periodicity of a time series is determined by the distance (i.e., the number of lags) between maximal values of the autocorrelation function. This is equivalent to measuring the distance between two sign changes. The periodicity is then simply given by the average distance across all sign changes. Nevertheless, in the presence of noise it might happen that some peaks are not detected in the first place and the above described strategy is prone to fail. In the following, we present a robust method to predict peaks in the presence of noise.

We compute the probability of $y_t$ being a peak as follows: Let $b$ be the number of detected peaks and $a$ be the average periodicity, we shift the actual time slice $t$ across the time series by multiples of periodicity $a$ and count the number of encountered peaks,

$$c(y_t) = \sum_{j=0}^{b-1} I[j \cdot a + t], \tag{1}$$

where $I[z]$ is the indicator function returning 1 if $y_z$ is a peak according to Definition 1 and 0 otherwise. After normalization we obtain

$$P(y_t = \text{ peak}) := \frac{c(y_t)}{\max_j c(y_j)}. \tag{2}$$

The measure $P$ can be interpreted as a Bernoulli variable for each $y_t$. Whether a time slice $y_t$ is finally treated as a peak is determined by a drawing $x_t \sim Uniform(0, 1)$ for each time $t$ and accepting $y_t$ as a peak if $P(y_t) \geq x_t$ and rejecting it otherwise. Note that we obtain at least one peak with probability 1 because of the division by the maximum.

## 3.5 Hybrid Models for Event Prediction

The predicted peaks of the previous section are a collection of time-stamps where we expect peaks. We now combine the baseline time series model with the set of predicted peaks.

We devise a simple but effective combination method. Let $Q = \{q_1, \ldots, q_j\}$ be the set of detected peaks in the historical

**Table 1: Query classification results.**

| Features | SVM | BNet | CART | J48 | RF | Log |
|---|---|---|---|---|---|---|
| TS | 34.18 | 53.64 | 57.55 | 49.82 | 31.68 | 52.64 |
| NTS | 54.45 | 44.36 | 32.45 | 36.91 | 34.26 | 40.64 |
| ACF | 78.91 | 76.09 | 76.00 | 71.27 | **84.26** | 78.73 |

**Table 2: Results for the peak detection.**

| | Prec | Rec | F1 |
|---|---|---|---|
| Mean | 67.6 | 92.2 | 78.0 |
| Median | 45.9 | 93.4 | 61.5 |
| 1st Quartile | 77.6 | 80.6 | 79.1 |

data, $P = \{p_1, \ldots, p_k\}$ be the set of predicted peaks, and $y = y_1, \ldots, y_T$ the future part of the time series model. The hybrid method simply substitutes the value of $y_t$ by the average of the detected peaks in the historical data for all $t \in P$. That is, the prediction $\hat{y} = \hat{y}_1, \ldots, \hat{y}_T$ of the hybrid model is given by

$$\hat{y}_t = \begin{cases} \frac{1}{|Q|} \sum_{q \in Q} y_q & : \quad t \in P \\ y_t + k \cdot (\bar{y} - y_t) & : \quad \text{otherwise} \end{cases} \tag{3}$$

for all $t = 1, \ldots, T$, where $\bar{y}$ is the mean of the historical data. The constant $k$ compresses the time series model around the mean to exclude interferences between predicted peaks and large values of $y_t$ that are not peaks and thus acts like a normalization factor. Throughout the paper we use the value $k = 0.4$ that performed well in tests with a small subset of time series.

# 4. EMPIRICAL EVALUATION

In this section we report on our empirical results. We use the New York Times corpus and apply a 3-fold cross validation together with the AIC criterion [3] for model selection in all our experiments. SARIMA models [3] are observed to consistently outperform ARMA and ARIMA models and we discard the latter two from the presentation and refer to SARIMA as the baseline time series model in the remainder.

The New York Times corpus is a publicly available collection of over 1.8 million New York Times articles annotated with rich metadata. It spans between January 1, 1987 and July 19, 2007. We collect queries from past TREC competitions and added a couple of own queries. The final set consists of 108 queries related to events (e.g., "super bowl"), accidents (e.g., "railway accidents"), tv series (e.g., "the simpsons"), and new technologies (e.g., "blackberry"), as well as time-related queries (e.g., "summer"). Using monthly binning for the content, our data comprises 25 periodic, 15 partially periodic, 31 trend-based, and 37 random time series. We use a monthly binning for the New York Times corpus.

## 4.1 Evaluation of Time Series Classification

We translate the queries into time series according to Section 3.1 and classify them manually into the four categories *periodic*, *partially periodic*, *trend-based*, and *random* for evaluation. We compare the predictive performance of Support Vector Machines with RBF kernel (SVM), Bayesian Nets (BNet), logistic regression (Log), and decision trees SCART (CART), random forests (RF), and J48 C4.5 (J48). We use three different sets of features: The time series itself (TS), a normalized variant thereof (NTS), and the autocorrelation function (ACF).

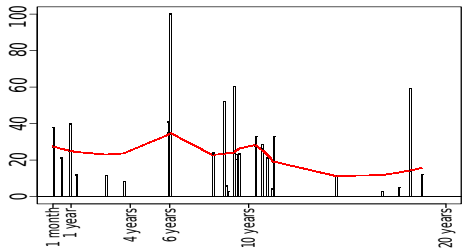


**Figure 1: Feature importance for NYT.**

The results are shown in Table 1. All methods perform best using the autocorrelation function which captures the nature of the problem reasonably well. The best result is attained by random forests with an accuracy of 84% which is significantly better than the performance of its competitors.

Figure 1 shows the most important features of the random forest which correspond to lags of the autocorrelation function. Unsurprisingly, periodicity is captured on a monthly and yearly basis by the classifier. Furthermore, multiples thereof are used for detecting reoccurring events. This is for instance the case for the most important feature which captures correlations of 6 years. While there is no event in the data that is repeated every 6th year, the period is ideal for capturing reoccurring annual, biennial, and triennial events.

## 4.2 Evaluation of Historical Peak Detection

For evaluating our peak detection strategy, we label all peaks in the time series manually. Our algorithm for automatic peak detection is a straight forward implementation of Definition 1. To allow for reusing the results for the peak prediction in the next section, we use only the first half of each time series for detecting the peaks and leave the remainder untouched.

Table 2 shows the results for periodic and partially periodic queries using mean, median, and first quartile as underlying function $g$ (see Definition 1). The left part of the table shows the outcome for all queries. The results can be characterized by an excellent recall and a reasonable prediction, which however not accurate enough by itself.

## 4.3 Future Peak Prediction

We now apply the obtained results to peak and event prediction on the remaining half of the time series. Before we report the results, we need to define some evaluation criteria that are suitable for comparing two sets of peaks.

Besides standard Precision and Recall, we consider weighted variants thereof to take the severeness of erroneous peaks into account. For instance, consider a time series with only a single peak and two predictions, one is very close but not in the same time-stamp as the true peak and other one is far away. Both predictions would realize a precision and recall of zero with respect to the standard definitions. The weighted counterparts w-Precision and w-Recall weight the error by the number of months between the true peak and the prediction to capture the difference in the predictions. Finally, we denote by MPD the mean distance of the predicted peak and the ground-truth.

We compare our approach denoted as *Pred* detailed in Section 3.4 with three baselines. The first is simply predict-

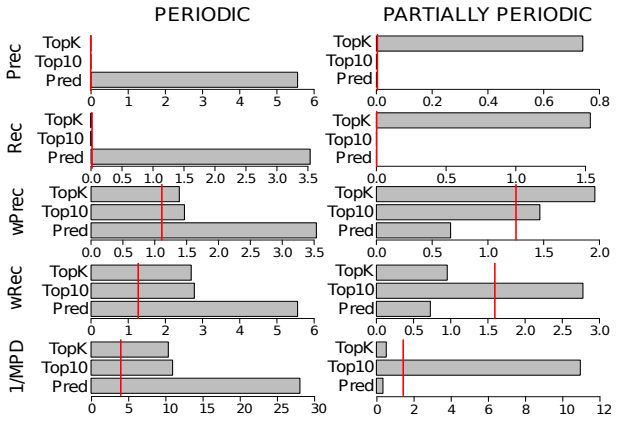


**Figure 2: Results for peak prediction.**

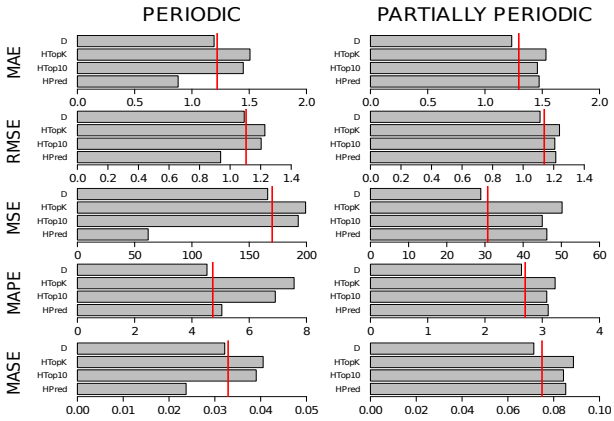


**Figure 3: Evaluation of the hybrid model.**

ing the publication dates of the top-10 retrieved documents as peaks, denoted by *Top-10*. The second baseline called *Top-k* is identical to top-10 but instead of 10 documents, we take $k$, where $k$ is the number of previously detected peaks in the peak detection. The third one is the SARIMA model which is also used to compute the periodicity. Peaks are predicted using the peak detection strategy with a 1st quartile function.

Figure 2 shows the results. The rows are the five evaluation measures and the columns correspond to the four categories. Note that we inverted MPD so that higher values are always better. The SARIMA baseline is indicated by red bars. Our method indicated by *Pred* improves all evaluation metrics for periodic time series significantly. For the partially-periodic category, this is surprisingly not the case. Here, Top-10 and Top-$k$ work significantly better.

## 4.4 Evaluation of the Predicted Time Series

In contrast to the peak prediction of the previous section, the hybrid model outputs a time series, hence we can compute its autocorrelation function, plot correlograms, and apply well-known distance-based performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Scaled Error (MASE).

We compare the hybrid model with the true continuation (the right half) of the time series. We deploy the three different peak prediction methods of the previous section in combination with the hybrid approach (see Equation 3): hybrid+top10 (HTop10), hybrid+top-$k$ (HTopK), as well as hybrid+Pred (HPred). To better understand the difference between peak prediction (case 1 in Equation 3) and smoothing (case 2 in Equation 3), we deploy another baseline, denoted as $D$, that simply smoothes all values of the SARIMA forecasting by applying $\hat{y}_t = y_t + k(\bar{y} - y_t)$ to all time slices. Additionally, the vanilla SARIMA model is included (red bars).

The figure shows the evaluation metrics row-wise and the columns depict the four categories. The SARIMA models compute a conservative prediction with respect to the mean that is, in general, improved using smoothing for periodic trends. Unsurprisingly, simply smoothing the time series around its mean (method $D$) does not change its predictive performance. However, using the hybrid method (HPred) significantly reduces the prediction error of the SARIMA model for periodic queries and performs best in our study.

# 5. CONCLUSIONS

In this paper we studied content-based future event prediction. For a given query, we translated the retrieved documents into a time series so that events become (indirectly) observable in the time series by bursts and peaks. Our approach is twofold: Firstly, time series are classified into four categories to determine whether event prediction can reasonably be applied. We experimented with several classifiers and features derived from the autocorrelation function. The best result was obtained using a random forest together with the autocorrelation function of the time series, yielding an accuracy of 84%.

Secondly, we extracted the periodicity of time series using again the autocorrelation function and estimated a probabilistic model to predict future peaks which are then intertwined with the baseline SARIMA model to produce a more competitive prediction model. The hybrid method is shown to significantly improve future event prediction compared to baseline methods including the original SARIMA model.

# 6. REFERENCES


[1] E. Adar, D.S. Weld, B.N. Bershad, and S.S. Gribble. Why we search: visualizing and predicting user behavior. In *WWW*, 2007.

[2] O. Alonso, M. Gertz, and R. Baeza-Yates. Clustering and exploring search results using timeline constructions. In *CIKM*, 2009.

[3] George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics, 4th edition, 2008.

[4] R. Catizone, A. Dalli, and Y. Wilks. Evaluating automatically generated timelines from the web. In *LREC*, 2006.

[5] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *WWW*, 2005.

[6] M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst, and A. önig. Blews: Using blogs to provide context for news articles. In *ICWSM*, 2008.

[7] S. Goel, J.M. Hofman, S. Lahaie, D.M. Pennock, and D.J. Watts. Predicting consumer behavior with web search. *National Academy of Sciences*, 2010.

[8] S. Goel, D. M. Reeves, D.J. Watts, and D.M. Pennock. Prediction without markets. In *EC*, 2010.

[9] H. Varian H. Choi. Predicting the present with google trends. Technical report, 2009.

[10] A. Kulkarni, J. Teevan, K.M. Svore, and S.T. Dumais. Understanding temporal query dynamics. In *WSDM*, 2011.

[11] M. Murata, H. Toda, Y. Matsuura, R. Kataoka, and T. Mochizuki. Detecting periodic changes in search intentions in a search engine. In *CIKM*, 2010.

[12] K. Radinsky, S. Davidovich, and S. Markovitch. Predicting the news of tomorrow using patterns in web search queries. In *ICWI*, 2008.

[13] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, 1994.

[14] Y. Zhang, B.J. Jansen, and A. Spink. Time series analysis of a web search engine transaction log. *Inf. Processing. Manage.*, 2008.