Why We Search: Visualizing and Predicting User Behavior

Eytan Adar, Daniel S. Weld, Brian N. Bershad, Steven D. Gribble
University of Washington, CSE
101 Paul G. Allen Center, Seattle, WA 98195-2350
{eadar, weld, bershad, gribble}@cs.washington.edu

ABSTRACT
The aggregation and comparison of behavioral patterns on the WWW represent a tremendous opportunity for understanding past behaviors and predicting future behaviors. In this paper, we take a first step at achieving this goal. We present a large scale study correlating the behaviors of Internet users on multiple systems ranging in size from 27 million queries to 14 million blog posts to 20,000 news articles. We formalize a model for events in these time-varying datasets and study their correlation. We have created an interface for analyzing the datasets, which includes a novel visual artifact, the *DTWRadar*, for summarizing differences between time series. Using our tool we identify a number of behavioral properties that allow us to understand the predictive power of patterns of use.

Categories and Subject Descriptors
H.2.8 [**Information Systems**] Database Management – *Data Mining*, G.3 [**Mathematics of Computing**] Probability and Statistics – *Time Series Analysis*

General Terms
Algorithms, Measurement, Experimentation, Human Factors

Keywords
User Behavior, DTW, Data Mining, Visualization

1. INTRODUCTION
Though there is a tremendous amount of research on the behavior of users on different web-based systems, there is almost no work on correlating these behaviors. Whether it is by surfing the web, posting on blogs, searching in search engines, or participating in social media systems, users leave traces of their interests all over the web. While interest in some things is constant (e.g. someone, somewhere needs access to their online bank), at other times it is periodic or spontaneously peaks in response to a breaking news event [14]. The effect of these events are ripples of behavioral changes as users rush to search for more information, consume news media, post in blogs, and participate in collaborative systems. However, the size of the response and how quickly users react depends on both the population and the medium. The reaction can be virtually instantaneous at one extreme (e.g. a search engine query), or require a great deal of time (e.g. the posting of a well researched news article). The goal of our work is to predict and explain behaviors by understanding how, when, and why these variations occur.

The ability to predict and classify behavioral reactions would have wide consequences on everything from the scientific understanding of sociological phenomena to the engineered optimization of search engines. The work described in this paper fits into our larger vision of broad, automated prediction by providing the infrastructure and tools necessary to explore how different Internet systems react in relation to each other. Though, we do not yet support *automated* predictions, our tools can provide answers to *user directed* questions such as: when do blog posts lead search behavior? Do users on different search engines react the same way to different news? Can the broadcast of a TV show impact search?

In this work, we develop a model for measuring responses in different web systems, both structured and unstructured, and a mechanism for comparing those responses using Dynamic Time Warping (DTW). Additionally, we create a visual artifact called a *DTWRadar* for summarizing and searching the differences between multiple time-series. Our tools and algorithms are motivated by a need to quantify and explore the event space in a human driven fashion as well as automated exploratory analysis. Using our tools and unique datasets, we conclude with a number of general findings on the relationship of search to other web-based behaviors.

Our work is based on the use of 6 datasets ranging in size from 10^7 search engine queries, to millions of blog posts, and to 1000's of votes on a specialized website. To our knowledge, this is the first large scale study correlating the behavior of multiple web-based systems over the same period.

In Section 2 we begin with a discussion of related work. We continue in Section 3 by introducing our data model and then describe our datasets in Section 4. In Section 5 we construct a simple topic model, the output of which we use to perform an initial correlation analysis in Section 6. As we will demonstrate, simple correlation statistics are not always the best mechanism for comparing behavioral datasets, and we develop an algorithm and the DTWRadar, a visualization primitive, to better support this in Section 7. By making use our tool, we conclude in Section 8 with a discussion of a number of general findings about behavioral properties of Internet users.

2. RELATED WORK
A critical aspect of analyzing time-series data is the ability to detect trends. We benefit here by the long-running interest in the data-mining community on trend detection in web and textual data (e.g. [5][10][24] and extensively reviewed in [11]). Such systems can extract a number of interesting events that can then be analyzed and compared in multiple datasets using our system.

Another requirement of our system is the ability to generate topics from query logs and text collections. As we are primarily motivated by query behavior in this work we have made use of algorithms similar to [26]. However, there is a large literature on mining topics from document collections (e.g. [2][12]). These may be useful to our work in the future as we begin to utilize

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

textual data source to automatically generate topics of interest and compare these using our techniques.

In predicting the effects of one data source on another there are a number of examples targeted at specific application areas including: the prediction of purchasing behavior [6], the effect of published text on stock prices [12], as well as interest levels in web pages based on the (dis)appearance of links [1]. These systems are most closely aligned with our goals, as they frequently utilize automatic alignment of documents to events.

Because we wish to support targeted data analysis (rather than simply automated mining) we are interested in providing users with visual representations of the data that can be quickly and easily interpreted at a glance. Though there are a number of systems for the visualization of time-series data (e.g. [7][13][23][25]), there are far fewer that support visual summaries ([13][23][25]), and none that we are aware of that provide a visual summary of the differences between datasets. Because of this, the DTWRadar may be of independent interest to other domains where time-series are compared.

3. QUERY AND TOPIC EVENT STREAMS

We use the term *query* in the usual sense, that of multiple tokens submitted to a search engine (e.g. "*american idol*"). We define a *topic* to more broadly represent a set of related queries (e.g. {*american idol*, *american idol finale*, *americanidol.com,* etc.…}). For simplicity, we will generally refer to a topic by the most frequent query for that topic.

Each dataset is composed of a number of events which we represent as tuples of the form <*text*, *weight, time, dataset*>. For the two query log datasets, *text* is simply the query string. For the BLOG, NEWS, and TV datasets *text* is the content of a blog, a news article, or webpage respectively. In general, the initial *weight* value assigned to each tuple is 1. However, because each dataset contains a large number of tuples we *bin* multiple tuples within a specific time range (1 hour units at minimum) that have equal queries and are from the same dataset. The *time* value in each tuple is an integer ranging from $hour_0$, representing the first hour of May, to $hour_{755}$, the last hour. For example, we represent the 10 queries for "*george bush*" that appear in the first hour in the MSN dataset as <"*george bush*", 10, $hour_0$, MSN>. Binning reduces the possible range of time (e.g. binning by 24 hour periods means the range of *time* is $day_0...day_{31}$). Note, that because different hours have varying activity, in general, we will normalize the weight by the amount of activity. Thus, if there were 1432 queries in the first hour, our previous tuple would actually be <"*george bush*", 10/1432 ≈ .007, $hour_0$, MSN>. Finally, for simplicity, we generate null-tuples (tuples with a weight of 0) for periods in which a given query/event is not witnessed. This will allow us to easily transform a set of tuples to a time-series.

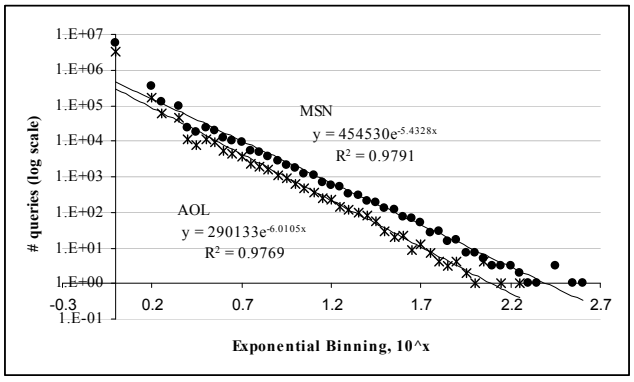


Figure 1: Query distribution of the MSN and AOL logs (log-log plot)

By filtering a dataset based on a query and source, and ordering by time, we have the basic unit of analysis: a time series which we call a *query-event-series* ($QES_{dataset\text{-}query}$). For example, $QES_{MSN\text{-}american\ idol}$ represents all tuples for which text = "*american idol*" and dataset = "MSN." Note that for the other datasets, $QES_{X\text{-}american\ idol}$ represents all blog posts, news articles, etc. that *contain* the phrase "*american idol*." Each QES is a time series with a characteristic curve where values on the *x*-axis represent time and *y* values are the weight. For ease of notation we use the semantics that for the QES *q*, *q(i)* will return the weight at time *i*. Finally, a *topic-event-series* ($TES_{dataset\text{-}\{queries\}}$) is a combination of all the QES instances representing a query in {queries} (the generation of these is explained in Section 5).

4. THE DATASETS

To support our research goal of understanding how different time-varying behaviors correlate, we leverage 6 datasets. Each dataset represents a historical record of the interest users have in different topics in different systems. Our datasets are composed of a log of queries and clickthroughs for MSN and AOL users, blog posts, news posts from CNN and the BBC, and posts about TV shows from TV.com.

4.1 MSN Query Logs (MSN)

The primary dataset used in this study is a query log sample from May 1 – 31, 2006 from the MSN search engine [15]. The log contains 7.47M sessions representing 15 million queries and clickthroughs. Although we do not have complete information about the sampling methodology, the logs represent a random sample of sessions (i.e. a continuous, cookie-tracked browser session). For consistency with the AOL dataset used below, we normalize the traces by "breaking" a session when a user has not been active for more than an hour. The result is 7.51M redefined sessions for the period. Rather than identifying true session barriers, our goal is to eliminate over-counting for users who are repeating the same queries in a short period or are surfing through multiple result pages (though clearly other definitions of sessions barriers such as [16] can be applied). To reduce the memory requirements, tuples were generated by binning at the hour level with the weight normalized by the total number of queries issued during that period.

4.2 AOL Query Logs (AOL)

The second dataset was extracted from the AOL query logs [19]. The trace represents the browsing behavior of 484k users generating 12.2M queries for the period of interest (May 1 – 31). As above, the AOL dataset was normalized into sessions by breaking long traces into a distinct session with an hour of inactivity (yielding 2.6M sessions for the period).

While breaking sessions in this way was an attempt at normalizing the two query logs, it is notable that the particular sampling methodology creates a different distribution of query popularity. Because users have a known preference towards re-querying information they have previously sought [21] there is a potential bias in the AOL dataset towards fewer unique queries. For example, despite the fact that the AOL session population is a

third the size of the MSN population it nonetheless generates a similar number of queries. Although the distributions (see Figure 1) for query recurrence are similar, there are nearly twice the number of unique session-query pairs in the MSN set (6.6M vs 3.7M). While this difference may be caused by some intrinsic feature of the MSN and AOL user populations, it may simply indicate that repeated sampling from the same set of users does not yield the full variety of queries found by random sampling. The effect of this is that the topics found in the AOL set may not represent a sampling of global interest in a topic.

Table 1: A summary of the datasets used

Dataset Name	Time range	Bins (hours)	Source records	Weight scheme
MSN	May 1-31	1	7.51 M search sessions	Searches per unit time
AOL	May 1-31	1	2.6M search sessions	Searches per unit time
BLOG	May 1-24	24	14 M posts, 3 M blogs	Posts per unit time
NEWS	May 1-31	1	13K news articles	Total in-links
BLOG-NEWS	May 1-24	24	13K news articles	Blog-source in-links per unit time
TV	May 1-31	24	2547 TV show records	Votes

Despite this, we note that both the AOL and MSN datasets share a common rank 1 query ("*google*") and have a 80% overlap for the first 10 queries. Beyond the 10th query, the overlap drops to ~66% and stabilizes for as high as the first 50k ranked queries. This high overlap provides some support for comparing the two datasets, at least for some queries. We will return to some of the consequences of this below.

One further issue with the AOL dataset was a small period of missing data of approximately 22 hours around May 16th. Clearly, we can not simply splice out the data from this period as it would represent a temporal shift that would be unmatched in the other datasets. We considered leaving the data at this range as 0 and also "patching" the data by replacing this period with the average values between the two end points as well as the linear extrapolation of the two ends. We found that all three options appear to generate roughly the same cross-correlation levels. We believe that the only situations for which we would fail to capture a correlation accurately is when there is a single spike in query behavior that greatly overlaps with the missing time period (assuming an equal distribution of spikes with short lifecycles this would mean a failed correlation on approximately 3% of topics).

4.3 The Blog Dataset (BLOG)

In addition to the two query logs we also made use of a database of 14 million posts from 3 million blogs from May 1 – 24, 2006 [18]. Though the posts were annotated with a date/time tag, these tags were not normalized to any time zone. By manually testing the feeds from a number of blogging sites (e.g. Blogspot, LiveJournal, Xanga, etc.) we were able to generate the correct time zones for a number of the posts. However, because many of the remaining posts came from popular blogs that were not readily validated, we opted to bin blog posts by day (24 hours).

4.4 The News Datasets (NEWS & BLOG-NEWS)

We hypothesized that the behavior of bloggers and search engine users was frequently influenced by items in the news. In order to validate this we generated a news dataset by crawling the CNN.com and BBC.co.uk websites in search of documents from the May 1– 31 timeframe. We selected these two sources as they allowed us to obtain news in an automated fashion and articles contained normalized (GMT-based) timestamps. We were able to download over 12k articles from BBC from the period of interest and roughly 950 articles from CNN (many CNN articles are derived from Reuters and are no longer available online).

Ideally, we would like as many news sources as possible in order to determine the popularity of a certain topic over time. However, because most sources do not provide access to their datasets after 7 days, it was difficult for us to find more articles from the period of interest. Since the CNN and BBC sources will likely publish only one story about a given topic, for most topics this does not give us much signal to work with (i.e. each QES contains only one tuple with a weight of 1). One modification to generate more realistic weights is to find the interest level in specific articles from external sources. For example, one non-binary approximation of popularity is the number of inlinks to a specific article. To find this value, we used the MSN search engine to find all links to the articles from external (i.e. non CNN or BBC) sources. Using our notation, we set each tuple's weight to the number of incoming links. We refer to this event stream as NEWS. Because all articles are time-stamped, the news dataset can be grouped at the hour level and normalized by the number of inlinks to all stories in that hour.

While re-weighting as above gives us a certain amount of information about eventual interest (useful in itself) it does not give us a sense of the changing interest level in a topic. By simply adding weight at the time of the article's publication based on interest, we force the mass of weight towards one point. In reality, interest in a topic, and hence an article's popularity changes over time as links are added. To simulate this we make further use of the BLOG dataset and count the number of daily inlinks to an article in the posts. Think of this as distributing the total number of (blog) inlinks so that the tuple from each day has a weight equal to the number of inlinks from that day. We call this dataset the BLOG-NEWS set as it gives us a sense of user interest in a topic based on the interest levels of bloggers. Because of the binning (24 hour binning) and time-span limits (only 24 days of data) of the BLOG dataset, the binning for BLOG-NEWS is at a daily level with data only up until the 24th of May.

4.5 The TV Dataset (TV)

A number of the most popular queries in the MSN dataset appear to be targeted at finding information about TV shows, actors, and actresses. It is likely then that the broadcast of a specific TV shows precedes or succeeds a burst of querying activity as users anticipate or react to the show. Similarly, interest in actors and actresses may vary based on the popularity of a show. To test this, we crawled for all episodes of shows broadcast during May of 2006 on the TV.com website. The site was chosen as it is a popular site for the discussion of TV shows and contains detailed information about the cast, summaries of the episodes, and user ratings. The crawl produced 2457 different shows for the period with 2 pages per show (a short summary and an extended recap). Because we would like the event "weight" to represent viewership, we estimate this value by the number of votes an

episode received on the website (all sites were given a minimum of 1 vote for normalization purposes). Because a particular TV show may be broadcast at different times on multiple occasions, pages from this dataset were tagged by day of the first broadcast and we bin in 24 hour intervals.

The BLOG, NEWS, and TV datasets were indexed in a Lucene (http://lucene.apache.org) index for easy retrieval. Table 1 summarizes the datasets.

5. FROM QUERIES TO TOPICS

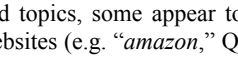
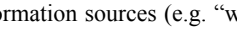
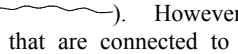
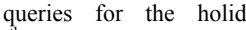
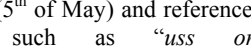
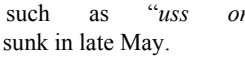
One issue we have ignored thus far is how topics are actually generated. Because we are primarily interested in studying how query behavior relates to other behaviors, we would like to group sets of queries into related buckets. There is a great deal of literature on topic detection both in data streams and static text collections [2]. While these techniques generate good results we found that a very simple scheme related to [26] yielded a useful set of queries grouped as "topics." Abstractly, the algorithm works by hierarchically clustering queries based on overlapping clickthrough and search engine result sets.

Our starting dataset includes the 963,320 unique queries that appear two or more times in the MSN and AOL query logs (lowercase normalized). These represent a pool of potential queries that we can lump together into topics. Each query was submitted to the MSN search engine and up to 50 results were returned. For each of the queries we also took into consideration the most popular clickthroughs. In the case of the AOL logs, where only the domain of the clickthrough was logged, we found the most likely full URL by comparing the domain of the click to the returned MSN results. Without much effort this allowed us to determine the full URL for ~40% of the redacted clickthroughs. Each query was then mapped to a set of URLs composed of the clickthrough URLs and up to 10 of the top hits from search engine. This set was transformed into a weighted vector using the standard TF-IDF scheme [3] where each of the 6,908,995 URLs (the "terms") was weighted by the number of times they are returned for the queries (the "documents"). Note that a side-effect of using clickthroughs was that results were term-order sensitive.

A pairwise comparison of the 963,320 queries—using a standard cosine distance metric [3]—resulted in 1,332,470 non-zero edges. An edge represents the similarity of one query to another and is used to cluster our queries for further analysis.

To construct an initial sample of potential topics to study we start by finding all queries that appear 100 or more times in the MSN logs (we believe these to be slightly more representative of broader search behavior than the AOL logs due to the sampling methodology as explained in Section 4.2). The resulting 5,733 queries are sorted by their frequency. Starting from the most popular query we generate a list of all related queries by traveling the edges described above (1 step). These neighboring queries are "assigned" to the initial query as alternative variants of that query. As we travel down the list of 5,733, those less-frequent queries that have already been assigned are ignored. The end result is a list of 3,771 main queries (i.e. topics) with an average of 16.2 variant queries. Note that the queries we use appear in multiple datasets 96% of the time and do not uniquely identify any user.

In analyzing the dataset we found that many variants were misspellings, changes in word ordering, and spacing. Although we have not done so in our experiments, our approach also lends itself to weighting QES's differently when generating a combined TES for some original query. That is, the weight contributed by a QES of a query can be made proportional to the similarity of the query to some "seed" query.

Of the generated topics, some appear to be navigational queries for corporate websites (e.g. "*amazon*," $QES_{MSN/AOL}$ =  [1] or "*bank of america*" ) while other are for search engines and information sources (e.g. "white pages"  or "*weather*" ). However, this set also contains various queries that are connected to external events. These include searches for the ever popular "*american idol*"  , queries for the holiday, "*cinco de mayo*"  (5th of May) and references to people and events in the news such as "*uss oriskany*,"  a US battleship sunk in late May.

The broad distribution of topics was encouraging as it represented both queries from which we would expect to see no interesting correlations, as well as highly correlated events.

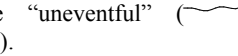
6. COMPARING DATASETS

There are a number of different parameters to consider in comparing the datasets. These parameters impact the performance of the correlations and require tuning and testing with different variations. There are 756 hourly bins in the period of interest and we can imagine comparing at different levels of granularity. This is supported by the binning parameter, b (recall that a number of the datasets are pre-binned at 24 hours, so for those b must be ≥ 24). As described earlier, binning simply reduces the number of event tuples in each QES by creating a new tuple with combined weights of the tuples in the period. Although binning smoothes the data somewhat there is still some noise in variations between bins. In order to diminish this noise in the time series we apply Gaussian smoothing as characterized by the function:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/2\sigma^2}$$

By convolving the Gaussian kernel with each QES or TES we are able to eliminate most noise. Our experiments and tools support arbitrary levels of σ. In the future, it may be worth considering smoothing techniques that are known to more accurately represent the visual characteristics of the original curve [27].

We made use of the experimental topics defined in Section 5 to create a set of TES's for the different topic/dataset pairs. For the log datasets this simply meant finding all queries that matched one of the queries defined by the topic. To generate the TES's for the remaining datasets we ran each query against the Lucene index. Although initially we allowed queries to be Boolean (simulating the most Web search engine behavior), a number of queries generated excessive low-relevance results. Rather than using an arbitrary heuristic threshold for discarding results, we opted to transform the Boolean queries into phrase queries. Thus, the query "*britney spears*" meant that both words had to appear

[1] We make use of sparklines [22] to represent the behavior of different QES's. These are simply a plot of the time series with tuple time as the x-coordinate and weight as the y-coordinate. Compare the "uneventful" () to "eventful" ().

together. Queries that behaved like stopwords (e.g. "*blog*" or "*www*") and matched a large number of documents were discarded (200 or more for the NEWS, BLOG-NEWS and TV datasets, or 25,000 in the case of the BLOG set).

Because not all topics exist in all data sources of the original 3781 topics we were able to generate 3638 (96% overlap), 3627 (96%), 1975 (52%), 1704 (45%), and 1602 (42%) TES's for the AOL, BLOG, NEWS, BLOG-NEWS, and TV datasets respectively.

To see how many of the TES's displayed random internal behavior we took all TES's generated above with a total weight greater than 10 (i.e. more than 10 searches for a topic, 10 blog posts, etc.). A test for partial (circular) autocorrelation at .05 significance level finds that 983 (~8%) of the final 12,324 are considered random (b = 24, σ = 0.5). Though we did not extensively analyze the content of random TES's, a brief scan of the topics indicates they are composed of company names, websites, and adult queries

6.1 Correlating the Datasets

To find the correlation between the two time series (x and y) we use the cross-correlation function:

$$r(d) = \frac{\sum (x(i) - \bar{x}) * (y(i-d) - \bar{y})}{\sqrt{\sum (x(i) - \bar{x})^2} \sqrt{\sum (y(i-d) - \bar{y})^2}}$$

the variable d is the delay and is varied from the –length(x) to +length(x). This represents the possible shift of the one curve away from the other to each extreme (all the way before, with no overlap, to all the way after). While we did not want to set an arbitrary threshold for a match, with an appropriate model one could limit d to a smaller range. The maximum value of r is the "best" correlation of the two functions, the value of d at this point is the "best" fitting delay between the two function. We reject the null of no-correlation by applying a Monte Carlo simulation that reorders one of the time series randomly and repeatedly finds the cross-correlation. By repeatedly generating max cross-correlations values less than the non-randomized time series we can reject the non-correlated hypothesis.

Though we concentrate on cross-source correlations in this paper (i.e. $TES_{source\ 1\text{-}topic\ 1}$ versus $TES_{source\ 2\text{-}topic\ 1}$) there is no fundamental reason why this analysis could not be applied to different topics in the same source. For example, by shifting the two TES's (e.g. aligning two movies' release dates), we could compare the reaction to two different movies in the same source (e.g. $TES_{BLOG\text{-}x\ men}$ versus $TES_{BLOG\text{-}superman}$).

Our working hypothesis is that if an event causes an effect in a dataset, all affected topics will see a positive change in weight. Although responses may be offset, it is unlikely that a news event will cause decreased search with increased blogging (or vice versa), and so we do not expect that negatively correlated behaviors correspond to valid mappings. In fact, when manually evaluating our results, and taking the positive magnitude of each correlations, we find that if the maximum corresponds to a negative correlation the topic appears to not correspond to any real event or trend. Topics of this type appear to be generic searches for company names (e.g. "*home depot*" or "*expedia*") or websites (e.g. "*nick.com*" or "*www.myspace.com*") and not responses to events. Because of this, we restrict ourselves to only maximum positive correlations.

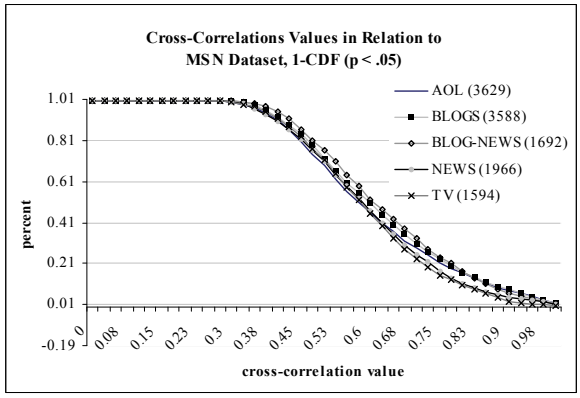


Figure 2: The cross-correlation values of all TES's against the equivalent TES on MSN, the *y* value at each value of *x* represents the percent of TES's for which the correlation is ≥ *x*.

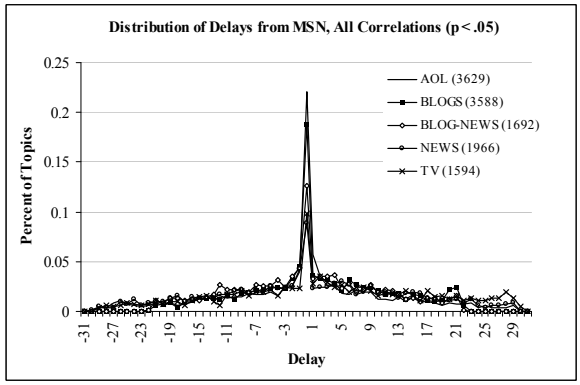


Figure 3: Distribution of delays (in days) of all TES's against the equivalent TES on MSN.

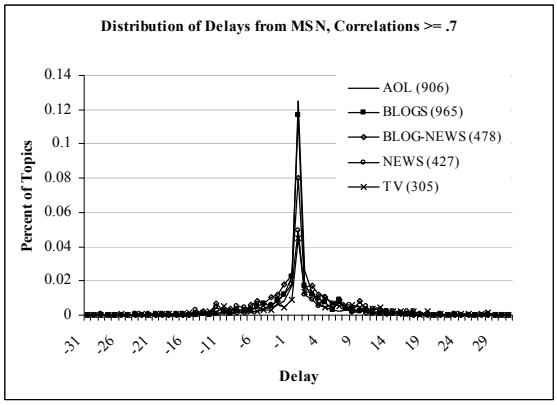


Figure 4: Distribution of delays (in days) of TES's with correlation ≥ .7 against the equivalent TES on MSN.

Figure 2 represents the distribution of all significant correlations found when comparing all TES's in the MSN test dataset against each of its counterparts in the other datasets (bins = 24 hours, σ = 2). The cutoff point for significance was determined by the formula: $\pm z_{1-\alpha/2} / N^{1/2}$, where z is the percent point function of the standard normal distribution, α is the significance level and N is

the number of "samples" (in most cases twice the length of the time series). The delay at which those correlations are found is depicted in Figure 3. Figure 4 is a depiction of the distribution for only those TES's for which the cross-correlation was ≥ .7 (well exceeding the $p < .05$ significance levels). On average, 38% of delays are centered at 0 days. What is interesting are the remaining, highly-correlated TES's that are shifted from 0. If these shifts are consistent, topics of this class are potentially useful for prediction purposes. However, there are many correlated TES's that are not simple spikes but are repeating patterns potentially representing a response to multiple events (e.g. a weekly TV show). While there might be an optimal delay as determined by the maximum correlation to align two TES's, each peak in periodic behaviors may not always lead or lag in the same way.

7. DYNAMIC TIME WARPING (DTW)

One of the limitations of simply shifting two correlated QES's is that it is still difficult to get a sense of how one stream relates to the other in term of magnitude. Imagine, for example, two correlated curves as in Figure 5 where one "encapsulates" the other. The difference in magnitude is drawn as a number of vertical lines in the top figure. While this represents a plausible mapping between the two time series, it fails to capture a different interpretation. The bottom series is essentially a smaller version of the top one, condensed both in time and magnitude. Thus a different mapping is one that captures these behavioral properties by mapping inflection points and behaviors such as the rise in one curve to the rise in the second, peak to peak, run to run, etc., as illustrated in the bottom of Figure 5.

A way to achieve this mapping is by making use of Dynamic Time Warping (DTW) [16]. Though there are several versions of this algorithm, a simple scheme using a dynamic programming approach follows:

```
DTW[0,0] = 0
for i = 1 .. length(x)
    DTW[0,i], DTW[i,0] = ∞
for i = 1 .. length(x)
   for j = 1 .. length(y)
      cost = sqrt((x(i)-y(j))^2)
      DTW[i,j] = min(DTW[i-1,j]+cost,
                     DTW[i,j-1]+cost,
                     DTW[i-1,j-1]+cost)
```

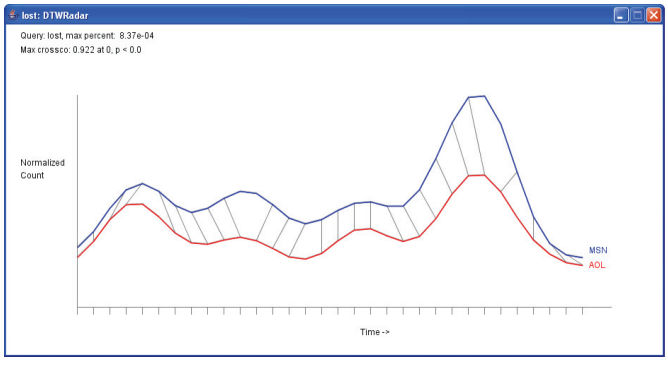


Figure 6: The DTWExplorer (time scale in day units)

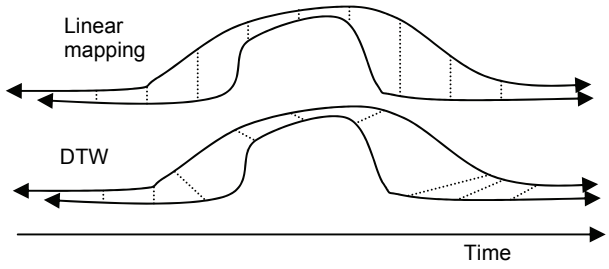


Figure 5: Demonstration of DTW

At the end of this run the two dimensional DTW array contains a mapping between the two time series. By starting at the extreme corner of the array and crawling backwards along the minimum gradient we find the best warp path. In experimenting with the algorithm we discovered that alternate versions of the cost function that include time as well as magnitude, do not appear to work as consistently due to the sensitivity of the cost function to the scaling. We also found that in general the results appear more accurate when we used a Sakoe-Chuba band [20], which restricts the warp path to a fixed distance from the diagonal of the DTW array. This technique reduces unrealistic warp paths that map distant events by limiting the max delay.

Another modification that we found useful was the use of the delay from the cross-correlation as a shift before calculating the DTW. That is, if the cross-correlation was at a maximum at d_{max}, we shifted one curve by that amount prior to calculating the warp path and then shifted the warp back by that amount.

Finally, we note that the DTW algorithm, as presented, generates an "optimal" mapping from sequence x to sequence y because of the order in which it generates the array. However, this mapping may not be optimal from y to x. To address this issue we calculate the DTW twice, switching the order of the input, and find those warp mappings that are common to both directions.

To allow users to explore the relationships between QES's and TES's from different sources we constructed an application called *DTWExplorer*. A sample screenshot is captured in Figure 6 for the seed query "*lost*" (a popular TV show, binned at 24 hour periods with $\sigma = 2$). Note the weekly oscillations around the time of the broadcast of the show with a significant peak around the season finale.

To use the DTWExplorer, a user submits a seed query to the system which finds a list of similar queries with which to construct a topic. These are listed in the table view similar to the

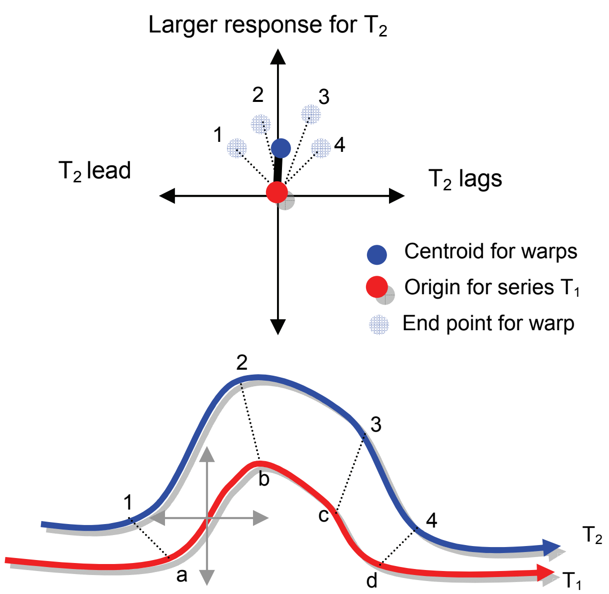


Figure 7: A graphical representation of the construction of a DTWRadar.

one on the right. The first column indicates whether the events matching the particular query should be included in the analysis. The second column is the query followed by the similarity based on the cosine metric described previously. The fourth column is the number of events of a certain type over the period of interest and the final column contains sparklines [22] that show the cumulative behavior of the QES for that query in both the MSN and AOL logs. Sorting is possible on any column. In the example we have sorted by the time behavior (the "distribution" column) which sorts by finding the cross-correlation of each series to the seed query. When sorted in this way, the top matching queries have a very similar distribution in terms of time. This is inspired by the work of [4] where semantic similarity was determined by time series correlations. The left hand display shows the output of the DTW on the selected query variants. The thin lines connecting the two curves represent the best DTW mapping. Note that in our implementation the applications supports searches on all indexed queries, not just the smaller, high frequency, sample.

The DTWExplorer application also supports the comparison of specific streams to generated time-series corresponding to different periodicities (e.g. once only, weekly, bi-weekly, etc.)

7.1 Warp Summaries: DTWRadar

While the warp paths generated by the DTW algorithm are informative for understanding the changing behavior over time, they do not necessarily give us a sense of the overall differences between the two series. Ideally, we would like a summary statistic or visualization that is descriptive of the total behavior of two series in relation to each other.

To achieve this we introduce the DTWRadar. The DTWRadar is a visual construct that takes each warp vector determined by the DTW algorithm and places it on a normalized axis (see Figure 7). Assume two time series T_1 and T_2 and a warp mapping between them (point *a* to point 1, *b* to 2, and so on). We construct the radar view by sliding a two dimensional axis along one of the time series (say T_1) from left to right. As the gray axes hits a mapped point on T_1 we draw a new point on the radar corresponding to the location of the end point of the warp on T_2. The result of this motion is illustrated at the top of Figure 7. Note the 4, slightly transparent, points on the radar, with a 5^{th}, darker, point showing the centroid of the original 4 points.

Understanding this radar simply means looking at the location of the centroid relative to the origin. From this specific example, we see that T_2 has a larger response (high on the y-axis). We also see a very small lag as the centroid is slightly to the right on the x-axis. If the two time series represented the number of queries for some topic X in the two search engines (QES_{1-X} vs. QES_{2-X}) we could say that, "more users query for X on search engine T_2 but with a slight lag." Note that the coordinate system of the DTWRadar maps directly to that of the time series. Thus the view allows us to know the average lead or lag time by looking at the position of the centroid along the x axis, and the magnitude difference by the position on the y. Keeping the transparent points in the radar allows a user to understand the distribution of the warp (i.e. the magnitude of T_2 is consistently higher than T_1, but varies in terms of lead/lag). Note also that the output of algorithm is a pair of points, one representing each time series. As we will show below, these points are useful for clustering behaviorally-related queries and topics.

We have found that once this representation is explained to a user they are able to easily interpret this view. This one view combines a great deal of data in a compact space and lends itself to the generation of many small multiples and facilitating comparison of multiple relationships simultaneously. We have extended the DTWExplorer so that it automatically generates a scaled version of the DTWRadar for each pair of series that are being evaluated. Figure 8 illustrates this for the $QES_{BLOG\text{-}prison\ break}$ vs. $QES_{MSN\text{-}prison\ break}$ (b = 24 hours, $\sigma = 2$).

7.1.1 Radar Variants

Clearly, there are a great number of possible variants to the DTWRadar that can enhance understanding of

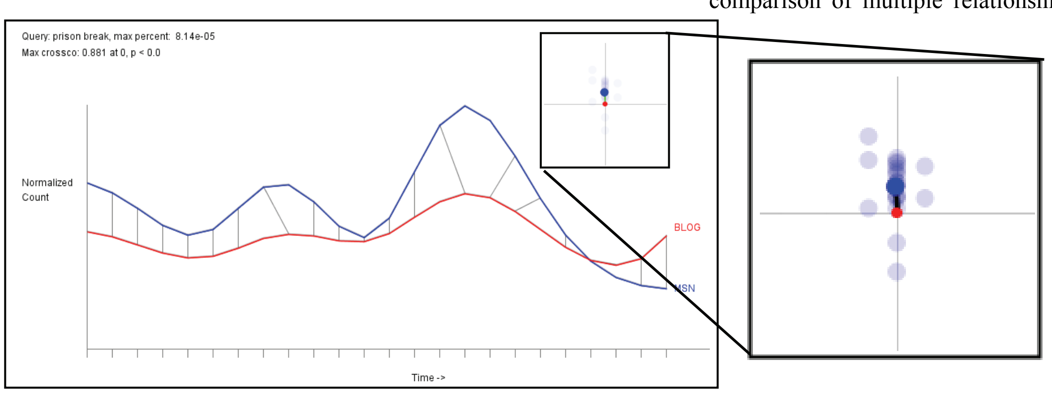


Figure 8: DTWExplorer output and overlaid DTWRadar for the query "*prison break*" (daily scale, radar view magnified on right).

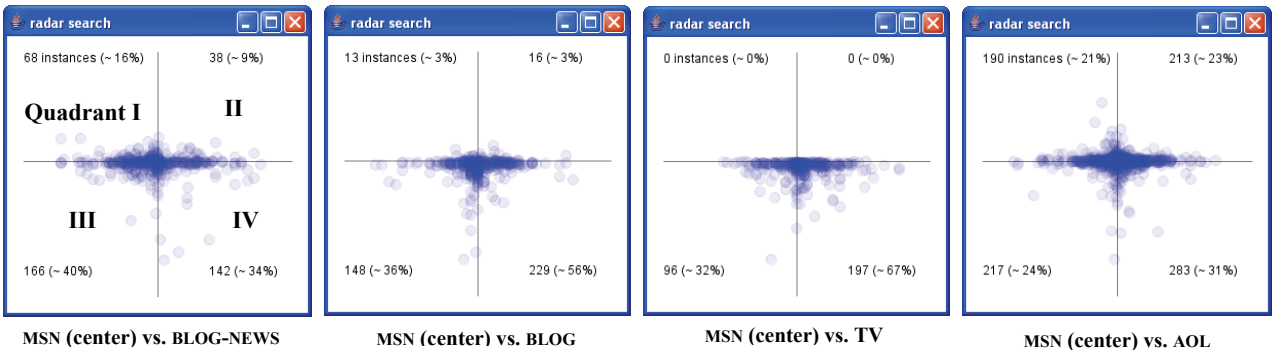


Figure 9: The cumulative radar positions for all queries against the MSN dataset. Only queries with cross-correlation ≥ .7 were considered. A point in Quadrant I indicates that the behavior leads searches in popularity and time. Quadrant II holds events that lag but have a higher overall popularity. Quadrants III and IV represent less overall interest than search with leading and lagging behavior respectively.

different properties. For example, though we have omitted tick marks from the view as they generally present a lot of noise in the scaled versions, these can be added.

We have also experimented with different centroid/mediod configurations for the "average" point in the radar. From our experience, it appears that the most useful selection of this point depends on the distribution of the individual warp points. One extreme warp point, correct or otherwise, can greatly influence the location of a centroid. At the moment, the explorer interface supports both forms. Another possible radar view is one that weights later mappings more than earlier ones. This can be graphically depicted with decreasing transparency on the mapped points and a weighted average to the "centroid."

We feel that the DTWRadar is both flexible enough to support the visualization of different properties while still being easy to quickly interpret. We believe that even if different alignment techniques [8][9] are used, this visualization is still informative.

Because we constructed a DTWRadar for each pair of QES's, we have a pair of points that describes the relationship between the series. One could easily envision support for searching this space of points. We have created an application that allows users to choose one source as the origin, and scaling those points representing the relative location of other QES's or TES's. Put another way, we have overlaid all the DTWRadars with one chosen source as the origin. By clicking and dragging a region of this meta-radar, the system finds all queries or topics that display a certain relationship and returns them to for further analysis.

7.1.2 Clustered results

A natural question is how different the data streams are from each other for all QES's. For example, we can ask: on average, does a QES from a blog source (i.e. posts mentioning a string) lead the QES for AOL? Figure 9 represents a set of overlaid DTWRadar centroids with MSN as the origin. We selected the set as those QES's determined to have a high correlation from the experimental list created in the correlation analysis. MSN was chosen because we are guaranteed to have data for MSN for any given query, though we can create these figures for any relationship. The centroid of centroids is not rendered as it sits fairy close to the origin in all cases. The display illustrates the spread of all TES's relative to the MSN query logs and indicates the distribution of lag times and magnitude differences.

One effect of our chosen weighting scheme for the TV dataset is that all centroids appear in quadrants III and IV (lower magnitude queries). Despite the fact that TV.com is popular, the reality is that limited votes are distributed among many shows (i.e. few votes per show). Similarly, we also discovered through this analysis that because of the tremendous number of blog posts per day, we had a significant reduction in the magnitude of each individual blog QES. Many blog posts are not in English or do not have anything to do with external events (i.e. the personal diary blog). In the future it is likely worth eliminating blogs that never match against any topic of general interest.

8. TRENDS AND DISCUSSIONS

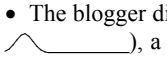
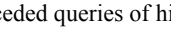
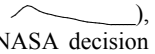
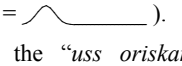
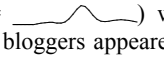
We are fundamentally interested in being able to use events in one information source to predict those in another. The radar views provide us with a global sense of the relationship of the responses to an event in one stream versus another. Though, as we saw in Figure 9, many of the centroids cluster in the middle, those that do not may provide insight into which sources, topics, and behaviors are predictive or predictable. For example, one might expect a ramp-up much earlier for expected events in the query-logs followed by a burst of posting and news activity in the blog and NEWS datasets around the time of the actual event. To understand which behavioral patterns are common, we are interested in classifying and characterizing the relationships between sources and topics.

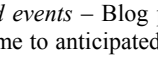
We randomly selected 244 queries from the set of highly-correlated queries. We manually annotated 216 of those with various categories (corporate websites, financial, TV shows, in the news, etc.). We discarded those queries for which the objective of the search appeared ambiguous, and no specific category could be assigned. Queries were allowed multiple categories. For example "*nicole kidman*" was tagged as both a *celebrity* and for being *in-the-news* (her engagement to Keith Urban was announced in mid-May). Similarly, "*world cup*," was tagged as both *sports* and in *in-the-news* as the lead-up and preparation for the event generated news stories.

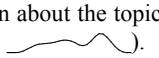
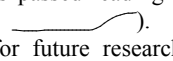
Below we describe a number of general trends and issues, some hypothetical, and motivate them by a few characteristic examples.

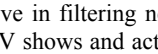
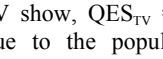
News of the weird – Certain events have the property that they are obscure and yet have the potential to become viral and capture the imagination of many individuals. We call this type of topic

"news of the weird," (in fairness, not all obscure topics are actually weird, but they may not be front page news). We find that bloggers tended to be much better at anticipating the eventual popularity of such stories. For example:

- The blogger discussion of "*igor vovkovinskiy*," (QES_{BLOG} = ), a 7'8" man who received a custom set of size 26 shoes and was featured in an Associated Press article, preceded queries of his name (QES_{MSN} = ).
- Similarly, the posting behavior around "*burt rutan*" (QES_{BLOG} = ), an aerospace engineer who criticized a NASA decision in early May was discussed in blogs earlier and queried later (QES_{MSN} = ).
- The response to the sinking of the "*uss oriskany*" (QES_{BLOG}= ) was nearly identical in shape to MSN, but bloggers appeared to go through the cycle 3 days earlier.

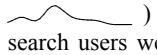
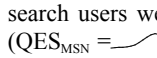
Anticipated events – Blog postings were not nearly as predictive when it came to anticipated events. For example, in the approach to the "*preakness*" horse race (QES_{MSN} = vs QES_{BLOG}= ), many users queried for information while few bloggers produced posts. We believe that this type of reaction is due to the social nature of blogs, and the reward for finding "new" information, making bloggers unlikely to discuss widely known events. As the time of the event approaches, and new information becomes available, bloggers will begin to react.

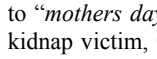
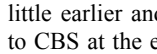
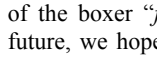
Familiarity breeds contempt – We notice a related effect when comparing news to search behavior. Though a topic is still newsworthy it may be so familiar to the average user that they will not actively seek new information about the topic. One such event is the "*enron trial*" (QES_{NEWS} = ). Though the trial was still a topic in the news, searchers ignored it until a verdict was passed leading to a sudden rise in search behavior (QES_{MSN} = ). Such behaviors form an interesting direction for future research as we may want to differentiate "significant" events from "standard" news. With additional news sources the difference is readily identifiable by the number of sources covering the story. This may be detectable with only a few news sources by noticing the time difference between the actual event and the time the story was published as well as the way in which the source conveys the story (e.g. a "breaking news" headline might be a good signal).

Filtering behaviors – One unanticipated consequence of choosing a highly specialized source such as the TV.com dataset was that it was effective in filtering news about certain classes of topics, in this case TV shows and actors and actresses contextualized to the shows they star in. For example, "*mischa barton*" (an actress on a popular TV show, QES_{TV} = ) saw a lot of overall activity due to the popularity of the show, with a sudden overwhelming burst of activity late in the month on MSN (QES_{MSN} = ) anticipating the death of her character in the season finale (the rightmost bump on the TV curve). By creating more specific labels of "source" (for example, by partitioning the blog dataset into communities), it may be possible to filter and classify certain topics by their behavioral correlation to these specific sources.

Elimination of noise – While we may want to break apart a source into highly specific sub-sources, it may also be useful to consider the combination of sources to reduce noise. Because news stories in the BLOG-NEWS dataset are weighted by blogging behavior, and because bloggers frequently reference news stories that they find worth blogging about, the events that are predictive in time and magnitude track the blogging behavior very closely. For example the behavior for "*uss oriskany*" is nearly identical in both datasets (cross correlation of .988 at 0 days delay). Because blogs are highly variable in quality and quantity of information, a blogger may simply make a reference to an article without actually discussing the content. Using the content of the pointer we can identify those bloggers which accurately capture the topic or automatically augment posts to include such information.

Fifty-two of the 216 queries were tagged as *in-the-news*. Because the 216 were suggested only when there was a high correlation to at least one other data stream this may provide a mechanism for finding interesting events. That is, if a given topic has correlated appearance in a number of data sources it has a high likelihood to be a response to a "news" event.

Correlation versus causation – One of the fundamental problems with trying to understand cause and effect is distinguishing between correlation and causation. While two QES's may both react to an event over time, the root cause of their reactions may be different. For example, a number of TV shows featured cast members, and spoofs, of the movie "*poseidon*" (QES_{TV} = ) at the time of the movie's opening. However, search users were frequently querying for the movie beforehand (QES_{MSN} = ). While there is an obvious correlation in the two streams due to the upcoming release of the movie, search users reacted to the marketing of the release in addition to the actual release. Thus, an "explanation" of the behavior of search users requires taking into account a combination of events (e.g. a marketing/advertising event source *and* a movie release database) and may require more complex models.

Portal differences – Today's search engine front pages are not simply an empty form box. Search engines are portals with news, entertainment, many other features. The information available on the front page may naturally lead to changes in behavior. While it may be impossible to dismiss these differences as being caused by demographic variations, we did notice a few that are likely caused by what information a portal chooses to present to its user's. For example, AOL users led in magnitude and time in their response to "*mothers day*" (QES_{AOL} = ) and "*ashley flores*" (a kidnap victim, QES_{AOL} =). On the other hand, MSN users led in queries for "*flight 93 memorial*" (QES_{MSN} = ), and in general appeared to respond to news a little earlier and with more queries (e.g. "*katie couric*['s]" move to CBS at the end of may, QES_{MSN} =  , and the death of the boxer "*floyd patterson*," QES_{MSN} =). In the future, we hope to capture the front pages of different portals to more accurately identify these differences.

9. CONCLUSIONS

In this work we have described the first large scale comparison and correlation study of multiple Internet behavioral datasets. We created a model for these events that allows us to automatically compare the reaction of a user population on one medium—be it search engines, blogs, or community sites—to the reactions on another. We implemented a visual tool, the DTWRadar, which allows users to view a summary of the differences between multiple time series and search for specific patterns. Using this tool, we mined the data for recurring relationships and described a number of key behavioral properties of different Internet systems and populations. We believe that the DTWRadar is of

independent interest and can be applied to any domain where time series are compared.

We are now taking the lessons we have learned about general behavioral patterns from this study and applying them to specific models that could automate prediction. Though our early analysis has generated some insight, our effort still required manual effort. We are now concentrating on the use of the techniques described here and other measures of behavior in conjunction with machine learning and time-series analysis techniques to determine which topics and behaviors are predictable.

We believe that this work is fundamentally important as the aggregate behaviors of crowds—wise or otherwise—represents a tremendous opportunity. Applications of such models including anything from understanding sociological phenomena to immediately practical systems ranging from marketing analysis tools to search engines that react to ever-changing user needs, whether those needs are for the latest on Iraq, or the latest news of the weird.

10. ACKNOWLEDGMENTS

Eytan Adar is supported by an NSF Fellowship, and an ARCS fellowship. This research is supported by a generous gift from Microsoft. The authors would like to thank Nielsen/BuzzMetrics for the blog dataset. Thanks to Lada Adamic and Brian Davison for crawling help, and Tanya Bragin and Ivan Beschastnikh for their comments.

11. REFERENCES

[1] Aizen, J., D. Huttenlocher, J. Kleinberg, and A. Novak, "Traffic-Based Feedback on the Web," *PNAS*, Suppl. 1: 5254-5260, Apr. 6, 2004.

[2] Allan, J., J. Carbonell, G. Doddington, J. Yamron, Y. Yang, "Topic Detection and Tracking Pilot Study Final Report," *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, Feb., 1998.

[3] Baeza-Yates, R., and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.

[4] Chien, S., and N. Immorlica, "Semantic Similarity Between Search Engine Queries Using Temporal Correlation," WWW '05, Chiba, Japan, May 10-14, 2005.

[5] Gabrilovich, E., S. Dumais, and Eric Horvitz, "Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty," WWW '04, New York, NY, May 17-12, 2004.

[6] Gruhl, D., R. Guha, R. Kumar, J. Novak, and A. Tomkins, "The Predictive Power of Online Chatter," KDD '05, Chicago, IL, Aug. 21-24, 2005.

[7] Havre, S., E. Hezler, P. Whitney, and L. Nowell, "ThemeRiver: Visualizing Thematic Changes in Large Document Collections," *IEEE Transaction on Visualization and Computer Graphics*, 8(1):9-20, 2002

[8] Keogh, E.J., J. Lin, and A. Fu, "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence," ICDM '05, Houston, TX, Nov. 27-30, 2005.

[9] Keogh, E.J., and M.J. Pazzani, "Derivative Dynamic Time Warping," SDM '01, Chicago, Apr. 5-7, 2001.

[10] Kleinberg, J., "Bursty and Hierarchical Structure in Streams," KDD '02, Alberta, Canada, Jul. 23-26, 2002.

[11] Kleinberg, J., "Temporal Dynamics of On-Line Information Streams," In *Data Stream Management: Processing High-Speed Data Streams*, M. Garofalakis, J. Gehrke, R. Rastogi, eds., Springer, 2006.

[12] Lavrenko, V., M. Schmill, D. Lawrie, and P. Ogilvie, D. Jensen and J. Allen, "Mining of Concurrent Text and Time Series," Workshop on Text Mining, KDD '00, Boston, MA. Aug. 20, 2000.

[13] Lin, J., E. Keogh, and S. Lonard, "Visualizing and discovering non-trivial patterns in large time series databases," *Information Visualization*, 4(2):61-82, July, 2005.

[14] Martzoukou, K., "A review of Web information seeking research: considerations of method and foci of interest," *Information Research*, 10(2), paper 215, 2004.

[15] Microsoft Live Labs, "Accelerating Search in Academic Research," 2006.

[16] Murray, G. C., J. Lin, and A. Chowdhury, "Identification of User Sessions with Hierarchical Agglomerative Clustering," ASIS&T'06, Austin, TX, Nov. 3-8, 2006.

[17] Myers, C.S., and L.R. Rabiner, "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected Word Recognition," *The Bell System Tech. J.*, 60(7):1389-1408, September, 191.

[18] Nielsen BuzzMetrics, ICWSM Conference dataset, http://www.icwsm.org/data.html

[19] Pass, G., A. Chowdhury, C. Torgeson, "A Picture of Search" Infoscale '06, Hong Kong, June, 2006.

[20] Sakoe, H., and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-26(1):43-49, 1978.

[21] Teevan, J., E. Adar, R. Jones, and M. Potts, "History repeats itself: repeat queries in Yahoo's logs," SIGIR'06, Seattle, WA, Aug., 6-11, 2006.

[22] Tufte, E., *Beautiful Evidence*, Graphics Press, 2006.

[23] Van Wijk, J.J. and van Selow, E.R., "Cluster and Calendar Based Visualization of Time Series Data," Infovis '99, San Francisco, CA, Oct. 24-29, 1999.

[24] Vlachos, M., C. Meek, Z. Vagena, and D. Gunopulos, "Identifying Similarities, Periodicities, and Bursts for Online Search Queries," SIGMOD '04, Paris, France, June 13-18, 2004.

[25] Weber, M., M. Alexa, and W. Muller, "Visualizing Time Series on Spirals," Infovis '01, San Diego, CA, Oct. 22-23, 2001.

[26] Wen, J., J. Nie, H. Zhang, "Query Clustering Using User Logs," *ACM Trans. on Info. Sys.*, 20(1):59-81, Jan. 2002.

[27] Witkin, A. P. "Scale-space filtering", IJCAI '83, Karlsruche, Germany, Aug. 8-12, 1983.