

Searching the Blogosphere

Nilesh Bansal
Department of Computer Science
University of Toronto
nilesh@cs.toronto.edu

Nick Koudas
Department of Computer Science
University of Toronto
koudas@cs.toronto.edu

ABSTRACT

With the massive adoption of internet technologies, social media and blogs have become primary means of expressing opinions online. The Blogosphere contains a wealth of information about a variety of topics that can be used for extracting useful and actionable insights regarding the ‘public opinion’. Information in the Blogosphere differs significantly from the traditional web content, hence requiring specialized technology.

We present BlogScope (www.blogscope.net), a system for online analysis of high volume temporally ordered text sources, currently applied to the analysis of the Blogosphere. We describe the system, its main concepts, and techniques developed for searching blogs.

1. INTRODUCTION

The explosive growth of the internet and the massive adoption of social media has created new ways for individuals to express their opinions online. Over 50 million blogs are reported to exist, and around a hundred thousand new blogs are created everyday [17]. According to the same estimates, blogging activity is doubling in size every two hundred days or about once every six and a half months. Bloggers blog about diverse topics including their personal lives, product reviews, political opinions, technology trends, tourism experiences, sports events, and the entertainment industry. Without a doubt, blogging is a social phenomenon. This trend will persist and grow as our lives become more heavily dependent on internet technologies. Given such trends there is pressing need to monitor such online forums continuously, and extract useful and actionable information regarding the ‘public opinion’ on a variety of topics.

Traditional web search technology can be readily applied on the Blogosphere. Indeed numerous search sites exist, specializing on the Blogosphere. Information in blogs however, has a well defined temporal dimension that is not present in more traditional web content. Blog posts can be easily associated with a geographical location which is the same as the location of the author¹. Moreover,

¹Traditional websites, e.g., en.wikipedia.org or www.yahoo.com, do not have a well defined geographical location.

blog posts may trigger additional posts by the same or other bloggers leading to a discussion in the Blogosphere. These factors make information in blogs and its dynamics differ significantly from the traditional web content, and hence there is a need for specialized search and analysis technology. We introduce BlogScope, a system with enhanced analysis capabilities (well beyond keyword search) for blogs. Note that, while we confine our discussion to blogs for simplicity of argument, much of our discussion is pertinent to all temporally ordered streaming text sources, e.g., news sources, mailing lists, forums, newsgroups, and other social media.

Consider a search for information related to the actor ‘Phillip Seymour Hoffman’ on the Blogosphere. The functionality that a traditional search engine will offer would be a list of all blogs posts, ranked in some order, containing the search string. Although this is informative, we argue that in terms of information discovery, there is a lot more functionality one can offer in the case of blogs. One for example could observe a graph displaying the relative popularity of the keywords ‘Philip Seymour Hoffman’ in the Blogosphere as a function of time and automatically tag regions of time that the search string shows unusual (unexpected) popularity. These can be temporal regions that one may wish to focus, refining the search. For this particular query it turns out that the keywords ‘Philip Seymour Hoffman’ displayed unexpected popularity over the year of 2006 in the Blogosphere when the actor was nominated for Oscar, when he received the Oscar award and when a subsequent movie that he was acting (not the one he won the Oscar for) was released (MI3).

From an information discovery perspective, details explaining the ‘unusual’ popularity of the keywords ‘Philip Seymour Hoffman’ in the corresponding temporal intervals should be automatically provided. We argue that keywords that are highly correlated (in blog posts) with the search string in a temporal interval of choice are good candidates for explaining ‘unusual’ popularity. For the case of the first temporal interval in which ‘Philip Seymour Hoffman’ shows ‘unusual’ popularity, the query is closely correlated with the keywords ‘Capote’ (the film he acted and was nominated for an Oscar) and ‘Oscar’. For the second temporal interval with the keywords, ‘Oscar’, ‘Actor’, ‘Capote’ and ‘Crash’ (another movie winning an Oscar) and for the third the correlated keywords were ‘Tom Cruise’, ‘MI3’ (a co-star in the movie MI3). It is evident that such keywords provide information as to why the query might show relatively ‘unusual’ popularity in the corresponding time interval. Notice, that such ‘correlations’ between keywords can be repeatedly discovered, possibly triggering additional information discovery. For example one might choose to identify the keywords correlated with both ‘Philip Seymour Hoffman’ and ‘Capote’ in the first temporal window. Such a functionality would enable a finer exploration of the posts in the temporal dimension. Essentially it

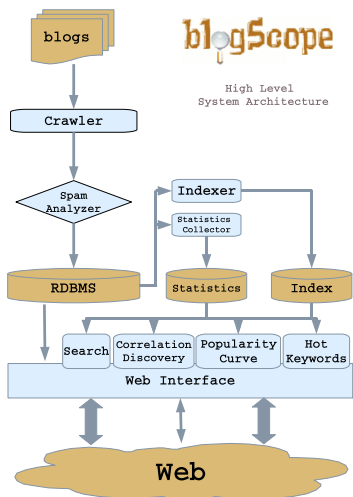


Figure 1: High level system architecture for BlogScope.

will enable a more focused ‘drill down’ in the temporal dimension.

We also argue, that information discovery in the Blogosphere is not necessarily query driven. One should be able to monitor posts and automatically suggest ‘interesting’ keywords to explore further. This paradigm is vastly different from the features offered by popular blog search services that solely monitor queries users pose or blog post tags and rank them based on relative popularity. There is a wealth of related information one can extract from blogs in order to aid information discovery. The list includes, adding a spatial component to queries and correlations, identifying temporal dynamics in the list of keywords correlated to a specific keyword, mapping correlated keywords to topics, to name a few.

In this paper, we present BlogScope (www.blogscope.net), a system for the analysis of the Blogosphere currently under development at the University of Toronto. BlogScope warehouses and indexes the Blogosphere extracting enough information in order to aid interactive analysis and information discovery. At the time of this writing, BlogScope was tracking around nine million blogs, indexing over 65 million posts in its database. Its features include (a) online, temporal *burst* detection for keywords, (b) efficient identification of correlated sets of keywords, (c) support for online OLAP style of analysis of the Blogosphere based on correlations and bursts (d) spatial blog post navigation (e) extraction of summaries for effective browsing in the form of hot keywords in the Blogosphere (f) authoritative blog post ranking and (g) support for effective ad hoc temporal reasoning for hot keywords.

We describe the system, focusing on techniques employed by BlogScope for analyzing the Blogosphere that scale to such massive level. Due to space constraints we discuss only a subset of features of the system, more details are available on the project website and in [1]. An overview of data aggregation and preparation techniques is presented in the next section. Section 3 describes various techniques used for analysis. Section 4 concludes the paper.

2. DATA AGGREGATION

In this section we briefly describe data aggregation and preparation techniques employed by BlogScope. At an average, around 300 thousand new posts are retrieved everyday by the crawler, which are then prepared for analysis. Figure 1 provides an high level overview of the system architecture.

2.1 The BlogScope Crawler

Crawling the Blogosphere is different from web crawling. An RSS feed is available for most blogs, and the crawler can fetch and parse the RSS XML instead of HTML. There is no need to follow outlinks as services like blog.g and weblogs.com maintain a list of recently updated blogs.

The BlogScope crawler receives a list of blogs updated in the last one hour from weblogs.com. We check this list against the list of spam blogs in our database (see next subsection for spam removal techniques employed), and schedule the rest to be fetched. Our current implementation only fetches blogs from blogspot.com but we plan to include more hosting services in the future. Once scheduled, we fetch the RSS feed after 12 hours and store all new articles. Addition of some delay in the fetch process helps reduce network accesses significantly as we fetch only once even when more than one article is posted on a blog in 12 hours (this is true for many machine created spam blogs).

2.2 Spam Removal

Spam is a big problem in the Blogosphere. Our experience² with blogspot.com data shows that half of the blogs are spam. These pages exist to boost page rank of some commercial sites. It is cheap to create many spam entries as hosting is provided for free by blogspot.com. Software is available in the market capable to automatically create thousands of spam blogs in an hour [16].

Spamming techniques are becoming quite intricate which makes the task of spam detection difficult. Language modeling techniques are used to generate sentences that are not just random strings but make some sense. Techniques used by spammers are sophisticated enough to confuse even a human observer at first. Many times, spammers just copy a news article from other sources and insert a few outgoing links to create the spam entry. Main characteristics of a spam blog are, (1) they are created by machine, and (2) they contain outgoing links to commercial sites.

Researchers have studied the problem of web spam in the past. A comprehensive taxonomy of current web spamming techniques is presented by Gyongyi and Garcia-Molina [4]. TrustRank [5] proposes a spam detection framework based on link structure. Spam detection in weblogs is an active research area. Kolar et al. [9] have proposed an SVM based approach to spam removal from the Blogosphere.

BlogScope’s spam analyzer builds upon previous work, utilizing a Bayesian classifier [14] in conjunction with many simple (but highly effective) heuristics. For example, spam pages contain a large number of specific characters (e.g., “-” and numerals) and contain certain keywords like *free*, *online* and *poker* both in their urls as well as in the urls of outgoing links. Capitalization of the first word of a sentence is often wrong in spam pages. Images are almost never present on spam blogs. These heuristics are based on manual analysis of blogs data.

Dealing with spam is an ongoing struggle. We plan to enhance our set of techniques deployed as we gain more insight on how spam is created in blogs [18].

2.3 Searching and Indexing

The crawler stores all its data in a relational database. At a certain time, this data is indexed to generate inverted lists and other statistics. We maintain two types of indices on all posts: standard and stemmed. Standard index maintains inverted lists for all tokens seen in the database while the stemmed index first converts all words to their roots, and maintains lists for all stemmed tokens.

²We have manually looked at a random sample of few hundred blogs



Figure 2: GeoSearch for the query *iphone*. Dots on the map represent regions where bloggers are writing about the searched query.

We also maintain a list of posts for each day. These indices form the core of our analysis engine. In a separate data structure, for efficiency, we also maintain *term frequencies* for each day and *inverse document frequency* over a *one year* sliding window for all stemmed tokens.

2.4 Spatial Component

Along with each blog post, while crawling, BlogScope attaches a city, state and country field and when possible geographical coordinates. There are several ways to infer a definite geographical coordinate given a blog post. These include:

- Utilizing meta data about location in the head of the blog. Several html tags and plug-ins exist to associate geographical information in blog posts. BlogScope automatically identifies such tags by parsing them and attaches a geographical set of coordinates to the post.
- Utilizing information related to the address of the blogger from its profile. The profile of a blogger may contain address information. In that case BlogScope extracts this information and maps it to a geographic set of coordinates. Approximate match information offered by tools like Spider [10, 15] enables effective matching of addresses.
- Looking up blog content against a set of standardized zip codes and city names also allows to extract geographic information from blog posts.

With the aid of such coordinates one has the option to identify the posts as a result of a query into a map and restrict the search using the map based on geography. Figure 2 provides an example screenshot. This enables BlogScope to conduct spatio-temporal navigation for blog posts and correlated keywords. BlogScope maintains inverted lists for city, state, country for blog posts. When the search is restricted using a spatial restriction, such lists are manipulated to suitably restrict the scope of the search.

3. ANALYSIS ENGINE

In this section we describe the analysis technology employed by BlogScope. The algorithms presented are efficient, scalable to millions of blogs, and amenable to online computation on streaming data sources. It must be noted that all the analysis is performed on the actual textual content of blog posts, and not on *tags* because: (1) tagging requires manual effort, (2) most blog posts are not tagged, and (3) a few tags can not accurately capture complete information present in a post.

3.1 Popularity Curves

The popularity curve for a keyword (or set of keywords) displays how often the query terms are mentioned in the Blogosphere

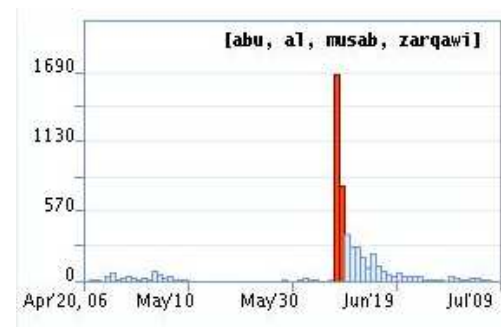


Figure 3: Popularity curve for *Abu Musab al-Zarqawi*, a member of Al-Qaeda in Iraq, was killed in an US air strike on 7th June 2006. Regions marked in red indicate bursts. Dark red bars in the curve (for June 7 and 8) mark bursts.

(in blog posts) as a function to time. Such a curve and its fluctuation can provide invaluable insight regarding the keyword popularity evolution over time. Figure 3 provides an example of the popularity curve for the query *Abu Musab al-Zarqawi* during April-June 2006. Popularity curves can also be used for comparing popularity of various keywords. We expect that closely related keywords will have very similar popularity curves at least for the temporal interval that the keywords are related. Hence, comparison of such curves provides an alternative approach to the analysis of the temporal relationship between keywords. Figure 4 displays the popularity of keywords *zidane* and *soccer*. Notice that the keywords exhibit strong similarity in their popularity for a short temporal window. This window spans a few days before the world cup final game (played on July 9) with a peak the day of the game, given the incidents in the game related to *zidane*.

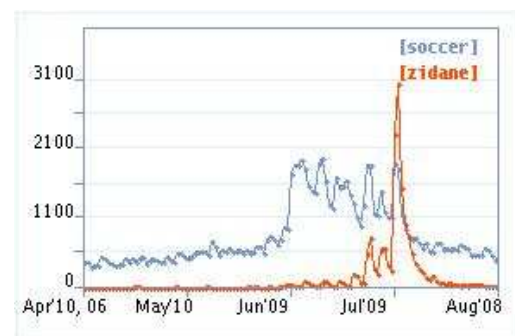


Figure 4: Popularity comparison curves for keywords *soccer* vs *zidane*.

BlogScope tracks the popularity of the keywords used in a query for a day by tokenizing the query and merging the inverted lists for each of the tokens (keywords) with the list of posts for that day. The popularity curve is generated by repeating this process for each day. The main cost of this algorithm is fetching the lists from disk; once the lists are read in memory, merging can be conducted very efficiently.

3.2 Information Bursts

Although blogging activity is uncoordinated, whenever something of interest to a fraction of bloggers takes place (e.g., a natural phenomenon like an earthquake, a new product launch, etc), bloggers write about it; and as a result, the popularity of certain

keywords increases. This fact allows BlogScope to intelligently identify and mark such interesting events on a popularity curve. We refer to these events as *bursts*. The notion of burst adopted by BlogScope is related to the notion of unexpected popularity for a keyword within a temporal window. Bursts play a central role in analysis and blog navigation using BlogScope, as they identify temporal ranges to focus and drill down, refining the search. Figure 3 shows an example of bursts.

Bursts can be categorized in two main types: *anticipated* and *surprising*. Popularity for anticipated bursts increases steadily, reaches a maximum and then recedes in the same manner. Release of a movie or the soccer world cup falls under this category. Unlike anticipated bursts, popularity for surprising bursts increases unexpectedly. Hurricane Katrina or the death of Abu Musab al-Zarqawi fall under this category (Figure 3).

Kleinberg [7, 8] has discussed burst detection in the context of text streams. Their approach is based on modeling the stream using an infinite state automaton. While interesting, this approach is computationally expensive, as it requires computing the minimum-cost state sequence that involves solution to a forward dynamic programming algorithm for hidden Markov models. It is therefore not possible to use this approach in our system where bursts need to be computed on the fly. Adapting this technique for on the fly identification of bursts would be prohibitively expensive. Fung et al. [3] have addressed the problem of bursty event detection, and have proposed techniques to identify sets of bursty features from a text stream. Inspired by the work of Fung et al. [3], the following algorithm is employed by BlogScope to detect bursts.

We model the popularity x of a query as the sum of a base popularity μ and a zero mean Gaussian random variable with variance σ^2 .

$$x \sim \mu + N(0, \sigma^2)$$

We can compute the exact popularity values x_1, x_2, \dots, x_w for the last w days by using our materialized statistics. We then estimate the value of μ and σ from this data using the maximum likelihood.

$$\mu = \frac{1}{w} \sum_{i=1}^w x_i \text{ and } \sigma^2 = \frac{1}{w} \sum_{i=1}^w (x_i - \mu)^2$$

From the standard normal curve, the probability of the popularity for some day being greater than $\mu + 2\sigma$ is less than 5%. We consider such cases as outliers and label them as bursts. Therefore, the i^{th} day will be identified as a burst if the popularity value for the i^{th} day is greater than $\mu + 2\sigma$. In our current implementation of BlogScope we use $w = 90$ to compute μ and σ .

3.3 Keyword Correlations

Information in the Blogosphere is highly dynamic in nature. As topics evolve, keywords align together to form stories; and as topics recede, these keyword clusters dissolve. This formation and dissolution of clusters of keywords is captured by BlogScope in the form of correlations.

With every search, a list of keywords in blog posts most closely related to the search query keywords is displayed. Such keywords can be seen as representative tokens for chatter in the Blogosphere, and can be used to obtain insight regarding the posts relevant to a query. Correlations are not static, as they may change according to the temporal interval specified in the query. Users can specify a temporal range for which a list of keywords correlated to query keywords is produced. Provided that users navigate, drilling down to posts related to a burst, such correlations can be used to argue why a burst occurred. Figure 5 shows a screenshot of correlations for *Philip Seymour Hoffman* for two different time periods, 1st-20th

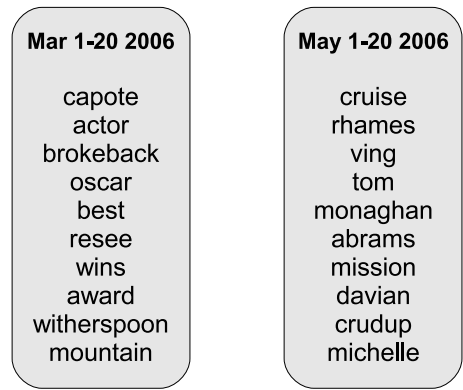


Figure 5: Correlations for keywords *Philip Seymour Hoffman* for two different time periods. Hoffman won the Oscar best actor award for the movie *Capote* on March 5th. Actress *Resee Witherspoon* and the movie *Brokeback Mountain* won several Oscar awards the same year. *Mission Impossible III* starring Hoffman (as Owen Davian), Tom Cruise and Ving Rhames, and directed by J.J. Abrams was released on May 5th.

March and 1st-20th May 2006. As it can be seen, correlations are different for different time intervals, and they reflect the events that occurred then. Choosing one of these keywords (say ‘capote’ in Figure 5) a list of keywords correlated to ‘Philip Seymour Hoffman’ and ‘capote’ in the temporal range will be produced, along with the associated popularity curve for the pair. Correlations are also employed by BlogScope to provide an exploratory navigation system. A user can easily jump from a keyword to related keywords and explore by following correlation links.

The notion of correlation of two random variables is a well studied topic in statistics [13]. Quantifying the correlation $c(a, b)$ between two tokens a and b can have many different semantics [12]. One semantics, for example, can be

$$c(a, b) = \frac{P(a \in D | b \in D)}{P(a \in D)} = \frac{P(b \in D | a \in D)}{P(b \in D)} = \frac{P(a \in D \text{ and } b \in D)}{P(a \in D)P(b \in D)}$$

where $P(t \in D)$ denotes the probability of token t appearing in a random document D in the collection \mathcal{D} ³. In words, correlation between a and b is the amplification in probability of finding the token a in a document given that the document contains the token b . The natural logarithm of the above mentioned correlation measure $c(a, b)$ is actually the pointwise mutual information [2] between a and b . Calculation of correlations using such semantics requires checking each pair of tokens. With tokens in the order of millions, calculating $c(a, b)$ using the above formula for every possible pair across several temporal granularities would amount to a large computational effort. This is complicated by the fact that such correlations have to be incrementally maintained as new data arrive. Increasing the number of keywords one wishes to maintain correlations for, from two to a higher number, gives rise to a problem of prohibitive complexity.

We describe a fast technique to find correlations which is currently adopted by BlogScope. Consider a query q and the collection of all documents \mathcal{D} . Let $\mathcal{D}_q \subseteq \mathcal{D}$ denote the set of documents

³Queries in BlogScope can be a set of tokens, in which case the semantics for $P(q \in D)$ can be extended to denote the probability of query q being relevant to a random document D in the collection.

containing all of query terms. For a token t we define its score $s(t, q)$ with respect to q as

$$s(t, q) = |\{D | D \in \mathcal{D}_q \text{ and } t \in D\}| * idf(t) \quad (1)$$

where $idf(t)$ is the *inverse document frequency* of t in all documents \mathcal{D} . The first term in Equation 1 is the number of documents containing t among those relevant to the query q . We multiply this frequency with $idf(t)$ which represents the inverse of overall popularity of the token in the text corpus. Commonly occurring tokens like “here”, “after”, “when” have high overall popularity and therefore low idf. Hence the proposed scoring function favors tokens which have low overall popularity but high number of occurrences in documents relevant to the query q . This represents keywords that are closely related to q as these tokens appear frequently only in the documents containing q . The list of top- k tokens having highest score with respect to q forms a representative of \mathcal{D}_q . We display this list as correlations for query q .

This technique requires a single scan over \mathcal{D}_q . As we scan the documents in \mathcal{D}_q , we maintain a count for each token that appears in \mathcal{D}_q in a separate hash table. After the scan is complete, we multiply these count values with precomputed idf values to find the scores, which can then be sorted to get the top- k . But even this could be prohibitively time consuming if the set \mathcal{D}_q is large. To circumvent this problem we bound the size of set \mathcal{D}_q by a number m ; if there are more than m documents containing query terms, we consider randomly selected m documents from \mathcal{D}_q . We denote the random sample of size m of \mathcal{D}_q by \mathcal{D}_q^m .

Notice that the proposed technique for finding correlated terms is in the same spirit as the one mentioned above based on amplification in probability. The number of documents containing both q and t , $|\{D | D \in \mathcal{D}_q \text{ and } t \in D\}|$, follows a binomial distribution, the characteristic probability of which could be approximated by scanning m random documents from \mathcal{D}_q . If we interpret the idf of a token t as $\frac{|\mathcal{D}|}{|\mathcal{D}_t|}$, where \mathcal{D}_t is the set of documents containing t , then the score of t with respect to q is

$$\begin{aligned} s(t, q) &= \frac{|\{D | D \in \mathcal{D}_q^m \text{ and } t \in D\}| \cdot |\mathcal{D}|}{|\mathcal{D}_t|} \\ &\propto \frac{\hat{P}(q \in D \text{ and } t \in D)}{P(t \in D)P(q \in D)} \end{aligned}$$

since $|\mathcal{D}|$ and $|\mathcal{D}_q|$ are constants for a given q . $\hat{P}(q \in D \text{ and } t \in D)$ denotes the estimate of $P(q \in D \text{ and } t \in D)$ based on m documents. Observe that $s(t, q)$ is proportional to $c(t, q)$ in expectation. Empirical evaluation however shows that the conventional interpretation of idf, i.e.,

$$idf(t) = \log \left(\frac{|\mathcal{D}|}{|\mathcal{D}_t|} \right),$$

works better. This is because interpreting idf as $\frac{|\mathcal{D}|}{|\mathcal{D}_t|}$ gives too much importance to very rare tokens; and this is further amplified by the fact that we are scanning only m documents.

The proposed technique requires a single scan over m documents among the search results for q . BlogScope uses $m = 30$, thus, considering just 30 text articles to find correlated terms for a query. Assuming that we have assessed that keywords q, t above are correlated in a temporal window, repeating this process, using q and t as a query (expanding the query set) would yield keywords correlated with q and t (thus obtain a larger set of correlated keywords).

An alternative way to identify correlations between terms, would be to identify terms with similar popularity curves, within the same temporal window. The premise (which is supported by term popularity curves presented by BlogScope) is that correlated terms have

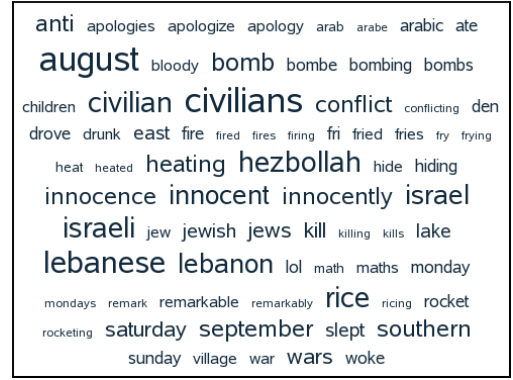


Figure 6: Example hot keywords cloud tag for 30th July 2006. The 2006 Israel-Lebanon conflict was at its peak during this time.

similar evolution in their popularity patterns. The similarity measure has to be robust with respect to scaling. Several approaches available in the literature for similarity queries on time series data are applicable [11]. Locality sensitive hashing [6] can also be employed for continuous nearest neighbor monitoring. We plan to explore this direction as part of our future work.

3.4 Hot Keywords

In its front page BlogScope displays a list of *hot keywords* for that day in the form of a cloud tag. BlogScope uses a measure of ‘interestingness’ for keywords (see Section 3.4) and ranks all keywords for a day according to this measure. The highest ranking keywords according to this measure, are displayed in the front page for that day with font size proportional to the measure of interestingness. Figure 6 shows an example screenshot taken on 30th July 2006.

Interestingness is naturally a subjective measure, as what is interesting varies according to the group of individuals it is intended for. Given the difficulty and the subjective nature of the task, BlogScope adopts a statistical approach to the identification of *hot keywords*. We employ a mix of scoring functions to identify top keywords for a day. In order to produce a final list we aggregate (using weighted summation) scores from all different scoring functions to find a ranked list of hot keywords.

Let x^t denote the popularity of some token t today, and $x_1^t, x_2^t, \dots, x_w^t$ be the popularity of the token in the last w days (except today). Let μ^t and σ^t be the mean and standard deviation respectively of these w numbers. We employ the following two scoring functions:

- *Burstiness* measures the deviation of popularity from the mean value and is defined as $\frac{x^t - \mu^t}{\sigma^t}$ for a token t . A large deviation (burstiness) of a token implies that its current popularity is much larger than normal. BlogScope, in its current implementation, uses a value $w = 90$ in this case. This value is set after conducting several experiments with BlogScope.
- *Surprise* measures the deviation of popularity from the expected value using a regression model. We conduct a regression of popularities for a keyword over the last w days to compute the expected popularity for today. Let $r(x^t)$ be this value. Then surprise is computed as $\frac{|r(x^t) - x^t|}{\mu^t}$. This measure gives preference to tokens demonstrating surprising burst, ranking anticipated bursts low. Our implementation uses a value of w as 15 for this case. The choice of w in this case is set after experimentation with BlogScope.

Both *burstiness* and *surprise* can be computed efficiently if term frequencies for each day are maintained precomputed. Final list of hot keywords for each day is computed by aggregating the scores from the two scoring functions.

3.5 Ranking and Burst Synopsis

We argue that there are important properties of the Blogosphere that cannot be easily captured by the ranking model in traditional web search. Documents on the web do not have a time-stamp associated with them, while blog posts have a definite time of creation. Simple relevance based ranking using $tf \cdot idf$ will mean ignoring the temporal dimension. Pure temporal recency based ranking will not be very good either. As a first attempt to address the ranking of search results in the Blogosphere, BlogScope employs a combination of both temporal recency and relevance to rank search results.

The semantics associated with the burst synopsis set for an initial query q is that it is the maximal set of keywords associated with q that exhibits a bursty behavior in the associated popularity curve for the set. Synopsis sets may have an arbitrary size (number of keywords) provided that all included keywords contribute to the burst. Authoritative blogs are blogs read by a large number of readers, and are usually first to report on news. These blogs play an important role in dissemination of opinions in Blogosphere.

Consider the query ‘italy’; blog posts may mention the keyword ‘italy’ in connection to both soccer and political events. All such posts contribute to the burst in the popularity of the keyword ‘italy’. The keywords ‘soccer’ and ‘politics’ are both correlated to keyword ‘italy’ in the associated temporal interval. However expanding the search and observing the popularity curves of ‘italy, soccer’ and ‘italy, politics’ turns out that only the curve for ‘italy, soccer’ has a burst in the temporal interval of the three summer months of 2006⁴. BlogScope can automatically generate such *synopsis* keyword sets for a burst. In this case, only the set ‘italy, soccer’ will be identified and suggested by BlogScope as a synopsis set, associated with the initial keyword query ‘italy’. Notice that the set ‘italy, politics’ will not be identified as a synopsis set, because ‘italy, politics’ does not have a burst in the corresponding popularity curve.

Based on such keyword sets, BlogScope automatically ranks blog posts related to the synopsis set based on *authority*. Authoritative blogs are the ones that gave rise to the burst on the synopsis keyword set. These are blogs that are relevant to the synopsis set, temporally close to the occurrence of the burst and most linked in the Blogosphere.

As an additional example, a search using query ‘cars’ on June 9th 2006 results in the synopsis set {cars, pixar, disney, movie} which disambiguate the burst resulted from the release of the movie Cars, from general discussion about automobiles in the Blogosphere. Such set is accompanied with authoritative blog posts that were the first to report the event and were most linked in the Blogosphere.

BlogScope computes the keyword synopsis set employing a greedy expansion technique using the original query keyword(s) as a seed set. We enumerate, keywords correlated to the searched query q , and then identify bursty intervals along the temporal dimension using the popularity curve of the correlated keyword in combination with q . We select the pair with maximum burstiness and iteratively repeat the same process till increase in burstiness is insignificant.

3.6 Interface

BlogScope adopts a simple and intuitive interface. Popularity curves provide OLAP style drill down and roll up functionality in the temporal dimension. Outlinks on correlations constitute a net-

⁴The Italian soccer team won their fourth World Cup, defeating France in Berlin, on July 9 2006

work of guided pathways to assist the user in a journey of Blogosphere exploration. Analysis using BlogScope can be summarized as a four step process.

1. Select keywords to analyze. BlogScope supports ad hoc keyword queries as well as suggests keywords in its hot keyword module.
2. Observe the search results (post snippets are displayed) ranked using BlogScope’s ranking function, the associated popularity curve of the keyword searched and its correlated keywords. Select a spatial region, if desired, to restrict the search in a specific geographic location.
3. Zoom in or out the popularity curve by selecting regions on it using the mouse. Select the time interval to analyze based on recommendations by bursts. Select the synopsis keyword set generation feature and observe blog posts ranked using authoritative ranking.
4. Use correlated keywords to reason about the burst. Outlinks on correlations can also be used to refine the query or explore further.

4. CONCLUSIONS

We have presented BlogScope, a text analysis system suitable for temporally ordered streaming text, currently applied to the analysis of the Blogosphere. BlogScope offers several unique features that when used in conjunction aid users to discover information but also navigate through information in a flexible way. We plan to continue enhancing BlogScope with several features to improve navigation, information discovery and performance and enhance the types of information sources that we index.

5. REFERENCES

- [1] N. Bansal and N. Koudas. BlogScope: Spatio-temporal Analysis of the Blogosphere. In *WWW Posters*. ACM, 2007.
- [2] K. W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *ACL*, 1989.
- [3] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.
- [4] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.
- [5] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, 2004.
- [6] Indyk, Motwani, Raghavan, and Vempala. Locality-preserving hashing in multidimensional spaces. In *STOC*, 1997.
- [7] J. Kleinberg. Temporal dynamics of on-line information streams. In *SIGKDD*. ACM, 2002.
- [8] J. M. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining Knowledge Discovery*, 7(4):373–397, 2003.
- [9] P. Kolari, A. Java, T. Finin, T. Oates, and A. Joshi. Detecting spam blogs: A machine learning approach. In *AAAI*. AAAI Press, 2006.
- [10] N. Koudas, A. Marathe, and D. Srivastava. SPIDER: flexible matching in databases. In *SIGMOD*. ACM, 2005.
- [11] J. Lin, M. Vlachos, E. J. Keogh, and D. Gunopulos. Iterative incremental clustering of time series. In *EDBT*, 2004.
- [12] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
- [13] R. L. Ott and M. T. Longnecker. *An Introduction to Statistical Methods and Data Analysis*. Duxbury Press, 1993.
- [14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 55–62, 1998.
- [15] SPIDER: a declarative data cleaning tool. <http://queens.db.toronto.edu/project/spider/>.
- [16] Splog software from hell. <http://ebiquity.umbc.edu/blogger/splog-software-from-hell/>.
- [17] State of the Blogosphere - aug 2006. <http://www.sifry.com/alerts/archives/000436.html>.
- [18] Y.-M. Wang, M. Ma, Y. Niu, and H. Chen. Spam double-funnel: Connecting web spammers with advertisers. In *WWW*, 2007.