Tracking Information Epidemics in Blogspace

Eytan Adar and Lada A. Adamic
HP Labs, Information Dynamics Lab
{eytan.adar,lada.adamic}@hp.com

Abstract

Beyond serving as online diaries, weblogs have evolved into a complex social structure, one which is in many ways ideal for the study of the propagation of information. As weblog authors discover and republish information, we are able to use the existing link structure of blogspace to track its flow. Where the path by which it spreads is ambiguous, we utilize a novel inference scheme that takes advantage of data describing historical, repeating patterns of "infection." Our paper describes this technique as well as a visualization system that allows for the graphical tracking of information flow.

1. Introduction

The rapid adoption of new tools such as weblogs (or blogs for short) and wikis is transforming the WWW. These tools have further advanced the ease with which new content can be created and have led to adoption of the web as medium for information propagation that had previously occurred online primarily through e-mail and instant messaging. One significant use of blogs is as an online diary where individuals list their own thoughts and experiences and occasionally comment on those of others. While at times blogs describe real-world experience, they are as likely to be related to the individual's Web based experiences. Further, as blog networks form social networks, with bloggers reading and commenting on each others content, newly discovered information can propagate through these online communities. Frequently, blog entries (also called *posts*) are intended to relay the latest interesting, humorous, or thought-provoking information the user has run across. For example, in early May of 2003 a company called Giant Microbes began selling stuffed animals modeled after common infections (e.g. the flu). Dave Barry, the popular humorist, mentioned this information on his site. Other bloggers, on reading about the Giant Microbes, became *infected* and re-published this humorous meme[8] on their own sites. Bloggers list these finds with the full realization, or hope, that they will be read by others.

In this paper, we study the pattern and dynamics of information spreading among blogs. Specifically, we are interested in determining the path information takes through the blog network. This *infection inference* task is related to both *link inference* and *link classification* but makes use of non-traditional features unique to blog data. Our goal is to correctly label graph edges between blogs when one blog infects the other. The difficulty is that frequently blogs do not cite the source of their information and appear disconnected from all likely sources of that information (i.e. other infected blogs). Thus, we are required to apply link inference techniques to find graph links that are not explicit.

Understanding how memes are propagated in blog networks is useful for a number of reasons. First, it allows us to understand how information can flow through other social networks and how these structures dampen or amplify this spread. Further, as we show in [3] infection inference can be utilized in the construction of alternate ranking strategies for search engines where we would like to find early sources of information (the so-called "patient zero") rather than the most popular information sources.

In Section 2, we expand on the structure and attributes of Blogspace. Using these properties we define a set of features appropriate for infection inference in Section 3. These include a set of novel features based on repeated infections between blogs and the timings of those infections. We then describe a Support Vector Machine (SVM) and logistic regression based classifiers to find and label potential infection routes. Finally, in Section 5 we describe an algorithm that constructs an *infection tree* which we use to visually render infections (the application is available at: http://www.hpl.hp.com/research/idl/projects/blogs/).

1.1 Terminology and Data

It is important to note that we are not studying memes themselves as they can take many forms which are difficult to work with and can be difficult to disentangle from other information (e.g. audio, video). Rather, we concentrate on well defined links to web pages external to the blog network. Links are far simpler to detect and track and variants are infrequent (i.e. there are many ways to describe a web page, but there are far fewer ways to point at it). We will thus be tracking the appearance and propagation of links (i.e. *URLs*), and will say that any blogs containing this URL are *infected*. In the Giant Microbes example this means that we will be tracking the link to the Giant Microbes website, http://www.giantmicrobes.com.

For our purposes a blog is a single web page containing time ordered entries. Our blog data consists of daily Blogpulse (www.blogpulse.com) differential crawls for May, 2003 as well as a fulltext crawl from May 18th, 2003. This data set contained 37,153 blogs with 175,712 URLs that have infected more than two blogs. The distribution of the number of blogs citing a particular URL follows a power law with exponent 2.7.

While approximately half a million URLs were mentioned by only a single blog, a few were cited by hundreds. This kind of highly skewed distribution is observed widely on the web, for example in how traffic and links are distributed among webpages.

1.2 Related Work

Due to their relative newness, academic research on blogs has been limited. Various measurements have been made to determine the size and activity of the blogspace[24] as well as some exploration on the dynamics of blog communities[3][18]. These prior studies have not specifically addressed the spread of information in blogs, with the exception of the work of Gruhl et al. [13], who concurrently with our work developed a method to identify topics discussed by blogs and to induce the graph of information traffic between blogs. There are also a number of online trend tracking services, including Blogdex, Blogpulse and Daypop, that list topics and URLs that are currently being discussed in the blogosphere. Marlow[22] used Blogdex to identify communities of blogs citing the same URLs, but did not track specific routes the information was taking.

Despite the usefulness of epidemiological terminology, the tools and goals of describing true disease epidemics are frequently different than our own. The goal of epidemic research is frequently to determine how far and how quickly an infection will spread rather than to track the disease through the population. Particular attention has been paid to the influence of network structure on the existence of an epidemic threshold, which determines whether a disease will spread from a randomly chosen individual to a fraction of the network[25][31]. Kleinberg et al. [16] identified which nodes, when infected, will maximize the spread of information. Recent theoretical work [27] has pointed at a possible way to infer contacts based on infection timings but has not been tested on real data. Perhaps the most related work is an attempt to predict the infection tree of a foot-and-mouth outbreak[14]. This simple but effective technique combined the distance between animals and infection timings to predict the infection tree. We will make use of this idea in creating our own infection trees.

In the social networks literature some attention has been given to link inference and diffusion[16]. Such techniques primarily utilize properties of the graph structure [1][4][6][12][17][28] as well as properties of nodes[29], but we are not aware of the use of timing information for predictive purposes.

Similarly, relational data mining research has also led to various advances in link prediction including inductive logic programming [23], probabilistic and statistical relational models [8][10][11], and structural logistic regression[25]. While related to our task, these systems are targeted at complex relational data and are frequently used for data discovery applications. The added complexity of these approaches is not necessary given our data and do not specifically address the task of infection inference.

2. Spread of Individual URLs

Given any URL, we do not expect that all blogs who have mentioned it first saw it at the source. More likely, a small number of sites initially made reference to it, and their readers replicate it further. Our goal then is to answer the question: is it possible to determine the source of infection for any given blog that is infected with a URL?

The explicit structural information found on blogs is a good starting point. Blog creators frequently provide *blogrolls* (essentially links to other blogs) indicating readership and possibly virtual, or real, friendships. Additionally, blog software allows users to automatically list referrer links (links generated by log analysis) and *trackbacks* [30]. Trackbacks provide a number of features, but the one of interest to us is the scenario in which one blog user, say Joe, links to Alice's blog. Joe's blog software will automatically inform Alice's site of the new link and Alice's blog software will add a link back to Joe. This link structure of the blog network can inform us about the popularity and centrality of different participants. Mining this structure has already led to the creation of a number of services (Technorati, Blogshares, etc.).

Figure 1 illustrates a possible sub-graph of blogspace. In this example all 4 blogs mention a URL, and furthermore, $T_{infection}(C) < T_{infection}(B) < T_{infection}(A) < T_{infection}(D)$ where $T_{infection}(X)$ is the time when blog X mentioned the URL. The infection inference task for blog A is then to determine from which other blog A was infected. Clearly, an infection from blog D is impossible as D was infected after A. If an explicit link only exists between A and B (link 1) we may infer that that B was the source of infection. Note, however, that this is still not a guarantee as other blogs may exist, that A is connected to, but which are not in our sample. Additionally, given the huge number of crawled links, one might expect that explicit link structure would largely describe routes of infection. However, this is largely not the case.

For URLs appearing on at least 2 blogs, 77% of blogs do not have an explicit link to another blog mentioning the URL earlier. For those URLs infecting at least 10 blogs, 70% are not attributable to direct links. The bulk of URL mentions on blogs are free-floating and are considered "unexplained." Possible reasons for this include missing data (e.g. uncrawled

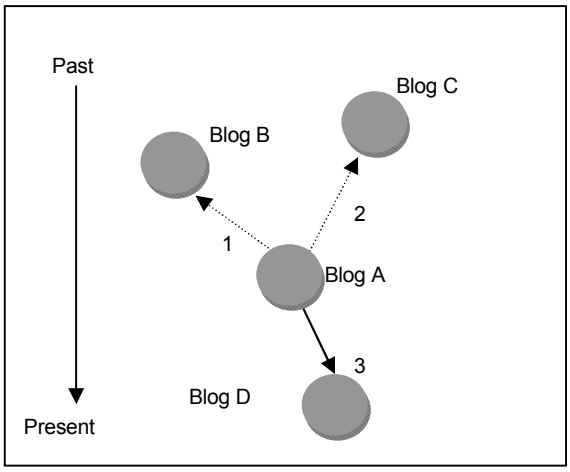


Figure 1: The Infection Inference problem.

or partially crawled blogs), external channels (e.g. infection through email, instant messaging, and mass media), different URLs for the same page (e.g. the URL to a mirror site or linking to a sub-page of the main website), and finally competitive dynamics of blog networks (e.g. authors trying to retain traffic). We therefore turn to other mechanisms for determining the source of the information. Whatever the reason, we frequently find that no explicit link exists to explain the infection (links 1 and 2 of Figure 1). This case requires the use of additional variables to infer which of the possible links is a plausible link. Finally, in the case where both links (1 and 2) are explicit and plausible we would like to determine which is more likely.

2.1 Via Links

While not often used, there is a mechanism of attribution in the blogosphere through "via" links. For example, the May 16th entry at *www.livejournal.com/users/bentleyw* reads:

"8:48a - GIANTmicrobes http://www.giantmicrobes.com/

'We make stuffed animals that look like tiny microbes—only a million times actual size! Now available: The Common Cold, The Flu, Sore Throat, and Stomach Ache.' (via Boing Boing)"

Unfortunately, via links are rare. Only 2,899 via links were found between two known blogs with an additional 2,306 links between a known blog and an uncrawled source (sometimes a blog, sometimes not). Additionally, blog posts containing via links may not contain any other trackable URLs. In exhausting explicit attribution as explanation of infection we are forced to rely on inference techniques.

3. Inferring Infection Routes

In order to accurately predict links on which an infection may travel we would like to take advantage of different kinds of features available to us about the blogs in our sample. These include structural features of the blog network, properties of the blogs themselves such as posted URLs and text, and finally the timing information of the infections. Our classifier uses a subset of 5 main features that relate one blog to another:

- The number of common blogs explicitly linked to by both blogs (indicating whether two blogs are in the same community)
- The number of non-blog links (i.e. URLs) shared by the two
- Text similarity
- Order and frequency of repeated infections. Specifically, the number of times one blog mentions a URL before the other and the number of times they both mention the URL on the same day.
- In-link and out-link counts for the two blogs

Blog features were obtained from two data sources provided to us by Blogpulse. A full text crawl of the blogs on May 18, 2003 served to provide blog URLs, non-blog URLs, and the full text. Infection timing was determined through analysis of differential text crawls conducted from May 2, 2003 to May 21, 2003, where only text differing from one day to the next was considered.

The sample of 1000 blogs was found by extracting links from the explicit network. This sample contained 1841 reciprocated and 2216 unreciprocated links. We randomly selected an additional 1000 pairs of unlinked blogs from the set to serve as negative examples. Because via links are rare for trackable URLs we do not make exclusive use of these for training and testing but consider them in the random sample.

3.1 Positive and Negative Examples

We briefly describe the examples generated for training our classifiers. Without the use of extensive human-driven surveys it would be impossible to determine the true path of infection (i.e. asking the creator of each blog where they first saw a given URL). We can, however, make reasonable approximations. First, we can classify whether blogs pairs are likely to be linked based on how similar the blogs are. Second, more pertinent to our goal of tracking infections, we classify explicit links as being likely or unlikely participate in the spread of infection. As we discuss below, both classifiers have positive and negative aspects that makes them appropriate for infection inference in different ways.

3.1.1 Link Existence

In order to determine if our features had any predictive power we applied them first to the task of identifying whether a link should or should not exist, essentially a standard link inference task. We define two positive classes: *reciprocated* (e.g. both blogs explicitly link to each other), and *unreciprocated* (one way links between blogs). The negative

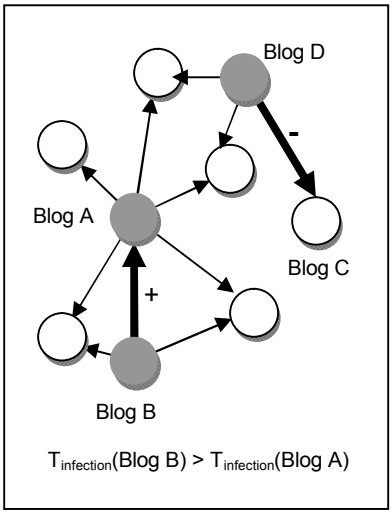


Figure 2: Positive (B → A) and negative examples (D→C) for the infection classifier. Existing edges that are part of an isolated, infected, blog pair or single blog are used.

class is *unlinked* pairs where both blogs are disconnected.

3.1.2 Infected Links

While it is difficult in the absence of via links to be absolutely certain that a link is responsible for an infection, we can nonetheless find plausible (positive) and implausible (negative) examples of infections. The positive examples are found by finding pairs of explicitly connected blogs that are infected with the same URL but are disconnected from any other infected blogs. Figure 2 graphically depicts the positive example (blog B → blog A) where blog B and blog A have been infected and are linked. We can infer that the infection of blog B from blog A is plausible given that no other neighbors of A or B have been infected. We further restrict the relationship so that blog B must be infected after blog A for the particular URL. Negative examples are defined as the link between an infected and an uninfected node as in blog D → blog C. Here, only blog D is infected and is disconnected from all other infected nodes. As blog C is uninfected we can surmise that the URL was not propagated over this connection.

While particularly restrictive, this approach isolates our examples and counteracts instance independence issues[15].

3.2 Blog Links, URLs, and Text Similarity

The first two features we examined were similarity metrics derived from blog-blog links and blog-non-blog (i.e. blog-URL) links. The similarity was computed in terms of a simple cosine similarity measure that ranges between 0 (no overlap in URLs) to 1 (all URLs are shared). If n_A and n_B are the numbers of URLs found in blogs A and B, and n_{AB} is the number of shared URLs, then the similarity is computed as:

$$s(A,B) = n_{AB} / \sqrt{n_a} / \sqrt{n_b}\,.$$

From Figure 3a, it is immediately apparent that blogs not linking to one another tend to point to different blogs and websites. On the other hand, blogs that do have links to one another tend to mention other blogs and URLs in common. This is not surprising, given the transitive nature of social networks. We find that if blog A links to B and B links to C, there is a 15% chance that A will link to C as well. We further found that the directionality of the link mattered, but not as much as whether a link was present at all. Blogs that reciprocated links tended to share slightly more links than ones where the link was not reciprocated. All similarity distributions for the three categories of pairs were found to be distinct by the Kruskal-Wallis test with $p < 10^{-5}$.

Textual similarity was determined through a basic cosine similarity metric on a term-frequency, inverse document frequency (TF-IDF [21]) weighted vector representing the textual content of the blog. The similarity metric followed a similar pattern to the other features, with unlinked pairs of blogs far less textually similar on average than linked ones. The contrast was lessened slightly by the abundance and ambiguity of words relative to URLs, which are unique and sparser.

While similarity features were useful in predicting the existence of a link, they did not provide as clear of a discriminator for links that were plausible and implausible infection routes. This is largely due to the fact that these links are sampled from already connected nodes that are likely to share URLs and hence be co-infected. As is visually evident from Figure 3b, for blog pairs that are already linked, the pairs that represent plausible routes of infection are not markedly more similar than ones that are not. For this reason, it is important to incorporate the additional information of infection timings when classifying plausible links.

3.3 Timing of Infection

A unique feature of blog analysis is that approximate timings of URL citations can be automatically collected, providing information on how frequently one blog is infected before another. Thus if blog A consistently cites the same URL before blog B we may be more convinced that A is an infection source for B.

As new links appeared on blog pages day to day, we looked at whether pairs of blogs mentioned the same blog and non-blog URLs, and whether they did so on the same day or

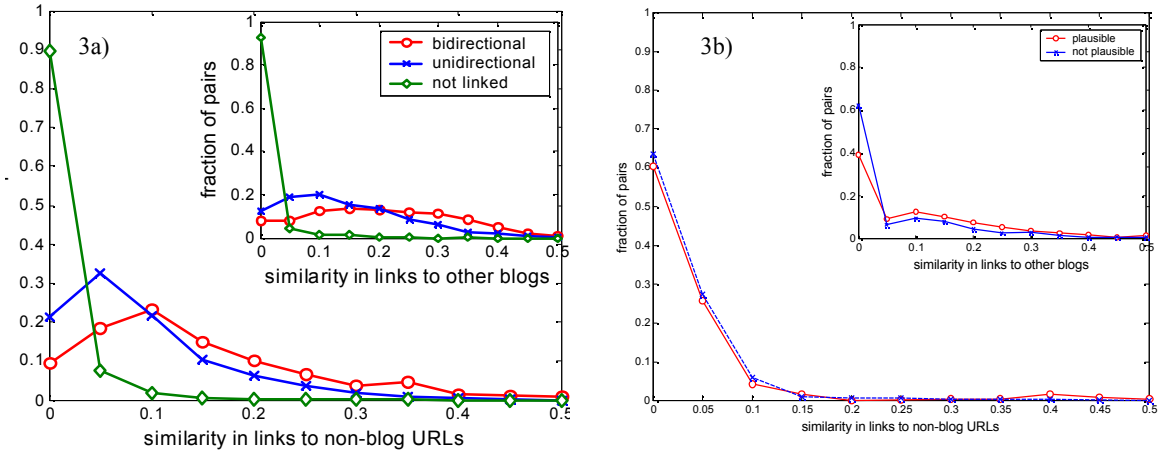


Figure 3a) Similarity for both non-blog URLs (main figure) and blog URLs (inset) for a) links between reciprocated, unreciprocated, and non-linked blog pairs, and b) plausible and implausible infections in existing links.

	% blog pairs that cite >= 1 URL in common			
blog-blog link type	same day	A after B	A before B	any order
A ←→ B	17.4	24.5	24.5	45
A → B	10.9	22.9	17.0	36
A,B (unlinked)	0.6	1.5	1.3	3

Table 1 Timing of URL citations between pairs of blogs, broken down by the existence and directionality of links between the blogs.

one day after the other. For some blogs we detected no new URLs added, and those we removed from the analysis. The results are summarized in the Table 1.

From Table 1 we can see that blogs that do not link to each other have a very small total probability (3%) of mentioning the same link during the period of approximately 3 weeks that the differential data was collected. Blogs linking to one anther have about a 45% chance of mentioning at least one common URL. Blogs with an unreciprocated link have 36% chance of mentioning the same URL, with a greater chance that A, the blog that has the unreciprocated link to B, mentions the URL on the same or later day than blog B. This makes sense, since we know that A knows of B and so might be getting content from it. To further check how strongly correlated URL mentions are with explicit links, we extracted three separate sets of blogs. The first had 100 pairs of blogs that shared the greatest number of URLs but mentioned them on different days. In this set, blog pairs had a 33% chance of having a link between each other. The set of pairs that had the greatest number of shared URLs on the same day had a somewhat lower chance (20%) of linking. This was due to noise introduced by duplicate sites (for example anythink.net and anythink.org are the same blog), or sites sharing generic sidebars duplicating the latest links from www.slashdot.org or freshmeat.net. In contrast, the 100 pairs that shared at least one link, but had the lowest overlap, had just a 4% chance of linking to one another.

Six independent features were extracted from the time ordered data: $A_{before}B/n_A$, $A_{after}B/n_A$, $A_{same\text{-}day}B/n_A$: $A_{before}B/n_B$, $A_{after}B/n_B$, $A_{same\text{-}day}B/n_B$. $A_{before}B$, $A_{after}B$, and $A_{same\text{-}day}B$ represent the number of links mentioned by A before, after, and on the same day as B respectively, and n_A and n_B represent the number of links referenced by A and B over the crawl period. Using our previous terminology $A_{before}B$ is the number of URLs for which $T_{infection}(A) < T_{infection}(B)$.

These features are even more interesting in the case of infection links. Specifically, in nearly 4000 sample instances where an infected blog A is connected to the infected blog B (infected before A), the average $A_{before}B/n_A$, $A_{after}B/n_A$, $A_{same\text{-}day}B/n_A$ scores are 0.014, 0.055, and 0.052 respectively. These scores are significantly higher than the average scores for the 4000 instances in which A is infected and B is not (2x, 6x, and 6x respectively). This indicates that infection is 6 times more likely when it has happened previously.

3.4 Classifiers

Using the features and training classes defined above it is possible to build classifiers to automatically infer links and detect likely routes of infection. Our experiments make use of both SVM[5][20] models and a simpler logistic regression classifier.

3.4.1 Link Inference

The first classifier (SVM) predicted three different classes (reciprocated links, one way links, and unlinked pairs). A second classifier (both implemented in SVM and as a logistic regression) distinguished simply between linked (undirected) and unlinked pairs. All SVM classifiers were trained with 10-fold cross validation with a radial basis function using all features.

An initial experiment using the SVM three-way classifier on 300 training samples did not perform well (57% accuracy, C = 32, γ=2.0, 1964 test samples). This result is primarily due to difficulty in differentiating between one-way and reciprocated links due to similarity in their features. However, this may be possible to resolve in the future through the use of additional trace data (more a-before-b type counts).

The two-way SVM classifier trained on 3572 examples and tested on 1485 cases. With an optimal C=.03125, γ=2.0, the SVM classifier yields an accuracy rate of 91.2%. The logistic regression classifier trained on the same data performs similarly (91.9% accuracy) but only makes use of the blog, link, and text similarity features. The regression assigned highest importance to the common blog links feature. Textual similarity is the least important feature for classification and given the expense of calculation is probably not worth the additional overhead. It may be that restricting textual similarity to words representing a new or "bursty" topic, as in [13] [18], would enhance the predictive ability of textual data.

3.4.2 Infection Inference

An additional set of classifiers was designed to handle the plausible/implausible infection route labels, illustrated in Figure 2. Both the SVM and linear regression classifiers were trained on 1108 examples and tested on 456.

The most successful logistic regression classifier used the features: $A_{before}B/n_A$, $A_{after}B/n_A$, $A_{same\text{-}day}B/n_A$, in-link and out-link counts to A and B resulting in 77% accuracy. However, it is notable that with only the first three features 75% accuracy is achieved. As might be expected, the most important feature in determining whether it is plausible that A obtained information from B is whether and how often A mentioned links after B, To a lesser extent, A mentioning links on the same day or before B helps classify routes as plausible. Various SVM classifiers were tested, ranging in accuracy from 61-71.5%. The best performing one used link similarity, blog similarity, and all six timing features with a C = 32, γ=32.0. These classifiers can then be used to evaluate how likely transmission of information is between any two blogs.

4. Visualization

Using the classifiers above as well as a set of heuristics we are now able to construct an infection tree, with each node

representing a blog. Visually we place each node at the vertical position corresponding to the date on which the blog posted an entry containing a specific URL. Links in the tree between the nodes show how infection may have spread for a specific URL. The constructed trees have been made available as a web service. Through the service users can enter a URL from the May crawl data. The system will automatically produce figures representing the explicit link structure. Note that the current public system only contains edges inferred through the two-class link inference SVM.

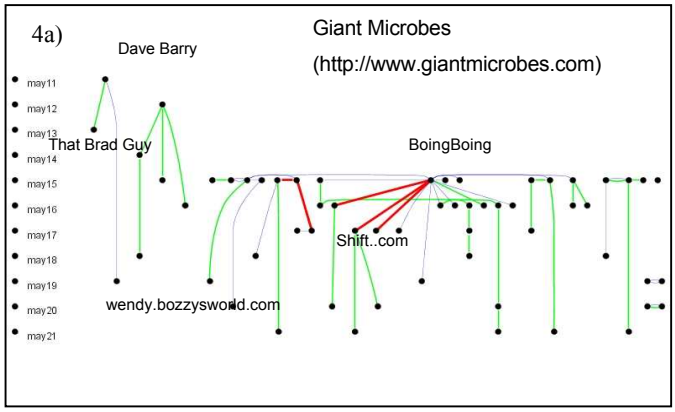
Two infection graphs are built for each URL. The first, a directed-acyclic graph (DAG), contains all possible links A → B that are either explicitly defined in the blog network or identified as being plausible infection routes by the classifier (as long as $T_{infection}(B) <= T_{infection}(A)$). Our system has also extracted the limited "via" information available in posts and those explicit links are labeled with a different color. We found that these trees contain a great number of edges, making them difficult to interpret. To address this issue we label each link with an inference score and allow users to dynamically control the threshold for display. Additionally, we build a simpler sparse tree that will attempt to "anchor" each blog by only the most likely edge (generating a true tree). This is achieved through the pseudo-algorithm described below. We use the following definitions:

- Let *via(A, URL_x)* be the set of blogs indicated by blog *A* as the explicit sources of *URL_x*. Each blog B in this set must further conform to $T_{infection}(B) <= T_{infection}(A)$.
- Let *explicit_directed(A,URL_x)* be the set of blogs that blog *A* explicitly links to and which are also infected by *URL_x*. Each blog B in this set must further conform to $T_{infection}(B) <= T_{infection}(A)$.
- Let *explicit_reverse(A,URL_x)* be the set of blogs that have an unreciprocated link to *A* that were infected by *URL_x* prior to A.
- Let *inferred (A,URL_x)* be the set of blogs that have been inferred by the classifier with timing restrictions

For each blog, *A*, infected by URL_x or algorithm is as follows. For the first non-empty set in (*via(A, URL_x), explicit_directed(A, URL_x), explicit_reverse(A, URL_x), inferred(A, URL_x)*), draw a link to each blog B in that set (e.g. if *via(A, URL_x)* is non-empty, draw link to each blog B in that set else check the next set). If more than one link exists between A and a previously infected blog use the classifier score, to remove all but the highest scoring link. Note that the algorithm does not guarantee that an "upward" link will be generated for each blog. For example, the earliest infected blog clearly can not be anchored to any source.

A further refinement allows us to incorporate blogs that we did not know were infected or that are outside our sample. This is done by using the via data. For example, boingboing.net is a popular blog that had mentioned the Giant Microbes site on May 14th, but this particular post was missed by the crawler . Our system, however, notes that a via link exists from an infected blog to boingboing.net in the Giant Microbes post. Adding boingboing.net into the network "explains" three additional sites that attribute the infection to it. A node for boingboing.net is generated and the $T_{infection}$ is set to one day earlier than the earliest node claiming infection by the boingboing site. If the via information points at a node not in our blog database a virtual "Other" node is generated and placed in the tree.

Both the complete DAG and "most likely" tree are visually laid out in layers for each day in a timeline style. The layout is done using the Graphviz tool [9]. This data is then imported into our exploratory graph analysis tool, Zoomgraph [2]. Zoomgraph functions as an applet allowing for the exploration of large graphs in a zoomable UI (infinite plane, infinite zoom). Using the Zoomgraph language we also allow



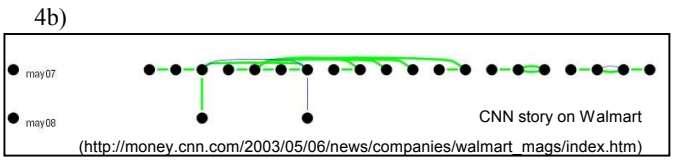


Figure 4a-b. Two depictions of the "most likely" infection trees determined by the algorithm described in Section 4. Thick red links (e.g. shift.com → boingboing) are "via" links. Thin blue links (e.g. wendy.bozzysworld.com → Dave Barry) are explicit links and the slightly thicker green links (e.g. thatbradguy.com → Dave Barry) are inferred links. Figure 4a represents a longer running infection for the Giant Microbes site and 4b a short topical infection (a CNN news article).

users to control which layers they are interested in as well as a classifier threshold.

The applet-based visualization allows for a quick analysis of the spreading patterns of URLs. For example, Figure 4 is the visual output for two URLs: the Giant Microbes site and a CNN.com news story. We can very easily see differences between these different epidemics. The Giant Microbes visualization shows many explicitly connected nodes that indicate a highly connected community is interested in the site. Additionally, we note that the infected nodes appear over a 10 day period (and potentially longer). On the other hand the CNN story received attention from nodes that do not display much explicit linkage illustrating that it is broadly interesting to individuals. However, we also see that the epidemic dies quickly as the story is most likely not interesting to connected groups and is no longer on the CNN front page.

In addition to the applet mode Zoomgraph also functions in an interactive mode that provides a graph analysis language that we can use to further explore the graph (e.g. finding disconnected clusters, exploring graph statistics, etc.). The full tool is also available for download and use.

5. Discussion

The results of all classifiers were encouraging. When applying them to practical applications such as constructing infection routes, we made use of additional heuristics to select only the most likely routes. One source of possible error in inference is the incompleteness of the crawl. For example, a link may be inferred between blog A and B since B is infected before A. However, there may exist an un-crawled blog (X) that is the true source of infection for both blogs A and B. The classifier may therefore detect a likely route of infection, but not the most likely. A further potential problem was that our timing data extended over only a few weeks, and so provided only sparse examples of likely infection routes. Some further work is necessary to determine how robust the classifiers are with respect to sample size. Both the issue of uncrawled blogs and sparse timing data can be helped by using larger crawls over an extended period.

Another aspect of meme propagation in the blogosphere that we did not incorporate into our current visualization system, but which could potentially assist in finding likely infection routes, is the fact that a meme can sometimes reside at multiple URLs. It may be possible to allow users to either manually, or with the assistance of text similarity type mechanisms, express to the system which URLs should be considered part of the same infection. While in most cases content is hosted at a single website (with one URL), URL variants may exist due to occasional mirroring (especially for high-bandwidth data such as video) or the original link may be subsumed by another site with "value added" information (collected links, such as news articles, about a specific meme).

For example, a popular Honda advertisement[1] in 2003 appeared as 14 URL variants on 359 blogs. The blog pairs that were linked were twice as likely (71%) to have referenced the same URL within 2 days of each other as those that were not (33%). The fact that the different 'strains' of the same meme exist makes it easier to discern who got the information from whom, but further work is required to make the inclusion of such information practical in large scale analysis.

6. Conclusions & Future Work

In this paper we have demonstrated a technique for inferring information propagation through a blog network. By analyzing the routes of individual URLs, we built a tool that is able to visualize and explain how information travels. These explanations are generated by utilizing explicit blog link structure, attribution links, node properties, and historical data. We also demonstrate a visualization tool that allows users to explore the propagation of specific URLs in the blog network.

We are continuing to refine the classification mechanisms. This includes better sampling and additional features. We are also interested in exploring ways of combining the link and infection predictors into one decision system to generate better predictions. Using the inferred graphs we are beginning to explore how graph structure can influence or cause different types of epidemics. Finally, we have begun using this and related inference techniques to produce novel ranking algorithms for blog search tools [3] and in the study of meme "mutations."

7. ACKNOWLEDGMENTS

We would like to thank Natalie Glance and Intelliseek / BlogPulse for giving us access to their data without which this work would not have been possible. Also thanks to Li Zhang and Rajan Lukose for useful discussions, Bernardo Huberman for his support of our work, and Sara Dubowsky for her invaluable advice.

8. REFERENCES

[1] Adamic, L.A., O. Buyukkokten, and E. Adar, A social network caught in the web, *First Monday*, 8(6).

[2] Adar, E., and J.R. Tyler, Zoomgraph, working paper. Available at: http://www.hpl.hp.com/research/idl/papers/zoomgraph1/index.html

[3] Adar, E., L. Zhang, L.A. Adamic, R.M. Lukose, Implicit Structure and the Dynamics of Blogspace, Workshop on the Weblogging Ecosystem, 13th International World Wide Web Conference, New York, May 2004.

[4] Butts, C. Network Inference, Error, and Information (In)Accuracy: A Bayesian Approach, *Social Networks*, 25(2):103-140.

[1] http://multimedia.honda-eu.com/multimedia/video/clips/cars/thecog.zip

[5] Cristianini, N., and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[6] Dombroski, M., P. Fischbeck, and K. Carley, An Empirically-Based Model for Network Estimation and Prediction,. *NAACSOS conference proceedings*, Pittsburgh, 2003.

[7] Domingos, P., and M. Richardson, Mining the Network Value of Customers, KDD'01: Knowledge and Data Discovery, San Francisco, CA, 2001.

[8] Dawkins, R, *The Selfish Gene*, Oxford University Press, 1976.

[9] Gansner, E. R., and S.C. North, An open graph visualization system and its applications to software engineering, *Software – Practice and Experience* 00(S1):1-5 (1999).

[10] Getoor, L., N. Friedman, D. Koller, Learning Structured Staistical Models from Relational Data, *Linköping Electronic Articles in Computer and Information Science*, Vol. 7(2002).

[11] Getoor, L., N. Friedman, D. Koller, and B. Taskar,, Learning Probabilistic Models of Link Structure, *Journal of Machine Learning Research*, vol. 3(2002), pp. 690-707.

[12] Ghani, A.C., C.A. Donnelly, and G.P. Garnett, Sampling Biases and Missing Data I Exploration of Sexual Partner Networks for the Spread of Sexually Transmitted Diseases, *Statistics in Medicine* 17:2079-2097.

[13] Gruhl, D., R. Guha, D. Liben-Nowell and A. Tomkins, Information Diffusion Through Blogspace, WWW'04: 13th Annual World Wide Web Conference, New York, May 2004.

[14] Haydon, D.T., M. Chase-Topping, D.J. Shaw, L. Matthews, J.K. Friar, J. Wilesmith, and M.E.J. Woolhouse, The construction and analysis of epidemic trees with reference to the 2001 UK foot-and-mouth outbreak, *Proceedings Royal Society B*, 270:121-127.

[15] Jensen, D., Statistical Challenges to Inductive Inference in Linked Data, in *Artificial Intelligence & Link Analysis*, D. Jensen and H. Goldberg, eds., AAAI Press, Menlo Park, CA, 1998, pp. 59 - 62.

[16] Kempe, D., J. Kleinberg, and E. Tardos, Maximizing the Spread of Influence through a Social Network, KDD'03: Knowledge and Data Discovery 2003, Washington DC, August 2003.

[17] Kleinberg, J., and D. Liben-Nowell, The Link Prediction Problem for Social Networks, CIKM'03: Conference on Information and Knowledge Management 2003, New Orleans, LA, Novermber 2003.

[18] Kleinberg, J., Bursty and Hierarchical Structure in Streams, KDD'02: Conference on Knowledge Discovery and Data Mining 2002, Alberta, Canada, July 2002.

[19] Kumar, R., J. Novak, P. Raghavan, and A. Tomkins, On the Burst Evolution of Blogspace, WWW'03: 12th World Wide Web Conference, Budapest, May 2003.

[20] LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[21] Manning, C. D. and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[22] Marlow, C., 'Classifying Emergent Communities through Diffusion' Presented at the Sunbelt Social Network Conference XXIII, Cancun, Mexico, February, 2003.

[23] Mooney, R. J., et. al., Relational Data Mining with Inductive Logic Programming for Link Discovery, in Data Mining: Next Generation Challenges and Future Directions, H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha, eds., AAAI Press, Menlo Park, CA, 2004.

[24] NILTE Blog Census, http://www.blogcensus.net/

[25] Pastor-Satorras, R. and A. Vespignani, Epidemic spreading in scale-free networks, *Physical Review Letters*, 86 (2001), pp. 3200-3203.

[26] Popescul, A. and L. H. Ungar,, Structural Logistic Regression for Link Analysis, Multi-Relational Data Mining Workshop at KDD'03, Washington DC, USA, August 27, 2003.

[27] Riolo, C.S., J.S. Koopman, and S.E. Chick, Methods and Measures for the Description of Epidemiologic Contact Networks, *Journal of Urban Health*, 78(3):446-457

[28] Skvoretz, J., T. J. Fararo, and F. Agneessens, Advances in Biased Net Theory: Defintions, Derivations, and Estimation, submitted for publication.

[29] Taskar, B., M.F. Wong, P. Abbeel, and D. Koller, Link Prediction in Relational Data, NIPS'03: Neural and Information Processing, Vancouver, Canada, December 2003.

[30] Trott, M., and B. Trott, A Beginner's Guide to TrackBack, http://www.movabletype.org/trackback/beginners/

[31] Wang, Y., D. Chakrabarti, C. Wang and C. Faloutsos. Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint. 22nd Symposium on Reliable Distributed Computing, Florence, Italy, Oct. 2003.