Topic-Based Blog Article Search for Trend Detection

Peter Wortmann

19th March 2009

Abstract

Social Media and especially Blogs are a fascinating new way for publishing. It puts ordinary people into the position to share their thoughts with millions around the globe. As the contributions seldom stand alone but are in fact actively enriched with references from and to other's contributions, all this can be seen as building a greater whole.

There is obviously a lot of information that can be learned from this mixture. The practical problem is to get a good overview, as the structures emerge dynamically in a seemingly chaotic way. Especially where trends crystalize quickly, it is desirable to also detect them as fast as possible.

This work will therefore aim to help the user to locate the most active discussion threads as they are emerging. To do this, we will first try to collect reliable data about recent Blogosphere activity concerning a topic of interest. In the next step, we will analyze the data and provide the user with the information we think is most characteristic for the current state of the blog networks.

Contents

Chapter 1

Introduction

1.1 Social Media

The term "Social Media" was coined primarily to contrast it against the traditional concept of "mass media" represented by print, radio or television. In their days, these techniques revolutionized our idea of information flow because they allowed some people to broadcast their thoughts to thousands and millions of people around the globe – to whoever was willing to listen. This democratization on the information receiver end can be seen as one of the big factors that led us into the present information-driven world.

Social Media now aim to take this process one step further. They provide everyone with the means to broadcast his or her information – thereby making all people equal on the sender end, too. This breaks with the classic asymmetry of information "producers" and "consumers" and may thus truly deregulate information flows.

On the other hand, where everyone has something to share, the amount of available information will obviously grow enormously. The receiver might find it increasingly difficult to filter out the interesting bits from the presented wealth of input. This makes proper organization and aggregation vitally important for using Social Media effectively. This will not work without technical help, and Social Media indeed rely heavily on the existence of a large modern computer network as a carrier.

1.2 The Web

The Internet is a such a network – it spans the whole planet and can establish connections between all of the millions of people that use it. This means massive potential for decentralizing the information flows around the world. At the same time, the vast computational resources available in all connected computers make advanced organization techniques feasible.

Even though this was actually realized even before the Web was born [4], it took some time for the idea to get momentum. The first attempts date way back to the design of HTTP, where it was thought that users might be allowed to not only view, but also change web pages and build the Internet in an collaborative effort.

But in practice, this did not work. This was mainly because most content is to this day still hosted on central web servers that are quite expensive to operate. As a result, server operators generally did not see why they should advocate user involvement on their web sites. So in early ages days of the Internet, normal users could only contribute information to the web in some limited ways:

- Web forums
- Home pages

Even though both forms already theoretically allowed users to reach everyone using the Internet, organization was still lacking. Even with the help of search engines, the data was just too hard to discover: Forums mostly formed their own micro-cosmoses centered around the topic dictated by the owner, and home pages lacked stability and a useful way to link individual contributions together. To make matters worse, a lot of forums do not allow indexing by search engines, so forum posts becomes unavailable for people that are unaware of the forum in the first place (it becomes a part of the "Deep Web" [3])

1.3 The Web 2.0

Lately though, there have been quite a few technological and ideological trends that favor a better implementation of Social Media. These trends have been grouped together under the term of "Web 2.0" by O'Reilly [12].

One of the main paradigm shifts was the realization by businesses that they could actually not only greatly benefit from accepting user contributions, but that pushing the limits in terms of organization tools was in their best interest, too. This is mainly due to the fact that in presence of good data organization, the value of user contributions can uniformly grow with the number of users.

This means that even if some scheme might look overly ambitious at first, it can benefit from the fact that a better user experience also means more user attention – triggering a positive feedback loop. This can spiral user attachment and involvement into web sites into heights previously unknown on the Internet. A business can then try to capitalize on this attention by directing some of it to commercial offers. This might either work indirectly by advertisements or directly by selling goods that under other circumstances would rarely find its buyers. This concept of selling goods from the "long tail" of the popularity curve was popularized by Chris Anderson [1].

1.4 Collaboration forms

As already mentioned, aggregating user contributions into a greater whole is essential for the service to be of value. In the context of the "Web 2.0", a lot of different collaboration forms were implemented for a number of purposes:

1.4.1 Wikis

By far the most liberal approach is to give each and every user the right to edit the contents of the page in any way he sees fit. This means that he can both

add new content or make new connections, but can also remove any content contributed by other users.

This model has been popularized by the Wikipedia and been successfully used to build a large number of so-called "Wikis" that mostly center around the idea of collecting knowledge about some topic. The main idea is that as long as the topic is objective enough, there will only be little debate between different editors over what the ideal contents of a page should be – so restrictions can be dropped in favor of a quick and unrestricted contribution process.

And even though this collaboration style is obviously prone to vandalism by malicious users, this approach has worked quite well in practice. This is because even though the model allows destructive change, all past contributions are archived automatically and can be restored quickly.

1.4.2 Comment Systems

The classic way for allowing users to contribute content to web sites is to allow attaching own additions at special points. In contrast to Wiki contributions, these additions are owned by the user and can not be changed by other normal users – if at all.

Most web pages show these additions as simple lists with author identification and time stamp for every single comment. Individual comments are in most cases sorted by submit order or some other criterion like writer and/or comment rating.

As this model allows clear distinction between individual authors' contributions, comment systems are better suited for content concerning subjective matters, like product reviews. On the other hand, in the presence of many users, comment counts might grow so rapidly that its usefulness peaks or might even start to decline.

1.4.3 Rating and Tagging

For coping with the problem of comment systems, it is necessary to aggregate and organize the collected user feedback before presenting it. In order to do this, it must take a form that can be processed automatically by computers. For both rating and tagging, this means that users simply contribute a value signifying agreement or disagreement with some interesting attribute of the object in question.

The contributed values can then be used for a statistical analysis that can help the web site organize the content better, like sorting by average rating. To this end, rating system have been implemented broadly and sometimes even recursively (meta-moderation).

Tagging follows a more general approach where the attributes voted upon are not fixed by the web site. Instead, users measure the strength of their personal association with freely chosen terms called "tags". This data can be used to allow for more elaborate content navigation, as it is by construction a very good source for measuring the relevance of a resource concerning a search containing the term. In most cases, this is done by specifying the own post to be a reply to an existing post.

1.4.4 Forums

Forums are similar to comment systems in that they allow users to add individual contributions to the web site which are time-stamped at creation and owned by the author. The main difference is that posts are independent contributions not bound to concrete objects, but the general topic of the forum or blog in question.

Characteristic for forums is that they do not see posts as stand-alone entities, but always in the context of other posts to the topic. This means that posts are always viewed on pages that group posts of different users topic together. A user is supposed to file his post into the appropriate topic when he contributes it to the forum.

1.4.5 Blogs

Blogs (short for We*b logs*), on the other hand, use a more free-form model where posts are meant to stand alone. Where needed, references to other posts are incorporated into the post itself using links. This has one serious advantage: As links can also point to external web sites, a blog article can not only reference posts from the same blog, but from all over the Internet.

For this to work well, it is critical to have links that directly and exclusively reference special posts. Such links are called "perma links" and are guaranteed to lead directly to the appropriate post for as long as the blog itself remains operational. At the core, this is all a blog needs to work.

This simplicity is one of the main reasons why blogs are so successful: Because of the low technical barrier, comfortable blogging services are abound. This lowers the entry barrier also for users that are not tech-savvy enough to author HTML, and makes blog discussions truly democratic in that anyone with an Internet connection can participate.

These properties make blogs the "purest" and most interesting form used for broadcasting Social Media. This is also why this work will largely focus on this collaboration form.

1.5 Blog Navigation

The alluring simplicity and robustness of the blog concept allows for truly unrestricted publishing by just about anybody. But as already mentioned, this is not enough to make Social Media work. The large information input has to be matched by better organization and aggregation, or readers might run into serious information overflow problems.

So for the Blogosphere to truly become the new data haven, additional systems are needed that help users to navigate the plethora of blogs and blog articles available. In this section, we will have a look at what users might use to navigate from blog article to blog article.

1.5.1 Links

In case a relevant blog entry is known, the obvious navigation choices are to use links provided by the blog author. This might be:

- Article links: In most cases references in blog articles were added to provide the context for the blog entry. As such, they are likely to be relevant.
- Other blog articles: As for the most part the owners use their blogs for writing about a fixed subject, other articles are also likely to be relevant.
- Blog links: Blog authors often provide a static list of blogs that they also consider relevant to the topic of their blog. Such link lists are common enough that they are commonly called "blogrolls".

It is also very common that visiting users are allowed to leave comments, for example with additional recommended links. In fact, this is an accepted way for blog authors to advertise their own blog articles.

1.5.2 Categorization and Tagging

Especially where the general scope of the blog is broader, there might be a lot of entries in only weak association. Many blog authors aim to counter this problem by classifying their blog articles into categories.

The viewer can then filter the blog articles to quickly locate posts with a common topic. If the author uses tags (see 1.4.3) for categorization, there is also a good probability that blog search engines will pick them up. With a link to the tag search page of such a search engine, the blog article can provide easy access to similar blog articles from other blogs.

1.5.3 Trackback

As already mentioned, blogs articles connected by a link are highly likely to be relevant to each other. This works both ways – but the link can only be followed in one direction. From a blog article there is no direct way to find out what other blog articles might have continued the discussion in the meantime.

Trackback (and Pingback) aim to solve this by providing a link notification mechanism. The idea is that a blog can be notified about new incoming links from other blogs using a standardized mechanism. The linking entries will then be shown as a list next to the referenced article, much like a comment.

1.5.4 Feeds

It is obvious that other articles from the same blog might be relevant – particularly *future* entries. This must be emphasized as blogs are no static entities and much of their power lies in their continued evolution. As a result, a user interested in the subject of a blog will also want to catch future entries. The problem is that checking a large number of blogs on a regular basis can quickly become tedious, even impractical.

The solution for this problem are so-called feeds. Using a standardized protocol (such as RSS or Atom), the feed provides a machine-readable list of recent blog articles with short descriptions and links back to the blog. This list can be checked automatically by feed readers, which might be implemented as computer software (like for example many popular E-Mail programs or Web browsers), but also as a web service (for example Bloglines). Many blogs provide links to such feeds on their pages.

In order to receive updates from a blog, the user will then pass this link to his feed service of choice. The service will then periodically try to identify new and interesting entries from all subscribed feeds and present the user with a summary list of all relevant activity.

1.6 Blog Search

Especially when the user has no blog to start with, he needs a service to come up with good start candidates from a rough textual description of the subject alone. blog search engines do exactly this by employing techniques similar to those used by the search engines of the World Wide Web.

1.6.1 Spidering

To find relevant blogs, search engines first try to index as much blog articles as possible. Like Web search engines, they employ so-called spiders that start with an initial list of blogs (or web sites) and then try to discover new ones by following links.

On the other hand, blog search engines are different from their Web counterparts in the aspect that they mostly deal with a quickly changing data base. In order to stay current, blog search engines must be fast to re-scan content once an update occurred. As it is impractical and impolite to scan a blog page too often, search engines use so-called *ping servers*. These servers take notifications from blogs that have updated recently and direct the spidering process accordingly.

1.6.2 Searching

Given a good view of the Blogosphere at large, search engines now have to try to identify blogs relevant to a search query by the user. There are a number of criterions that can be considered:

- Search phrase matching: The phrase given must be fully or partially present in the blog. Most search engines also try to search for semantically equivalent phrases.
- Article recency and post frequency: A user searching blogs is in most cases interested in a current discussion. So blogs with recent entries are generally to be preferred – as are blogs that seem to have a good history at staying up to date.
- Incoming links: Other people linking to the blog or the concrete blog post generally indicate interest in the contents. Links can be from blog articles and blog rolls, but also from normal web sites (in case such data is available).

  This method can be improved further by considering network effects. If a blog or web site is highly ranked in the view of many users, links from such a site will also be a greater indicator of the target's quality. This approach was popularized by PageRank as used by the Web search engine Google [13].

- Other user feedback: We already mentioned numerous services built around blogs that may yield additional data. Feed reader services, for example, can use their data to provide statistics about their users' subscriptions. If there is a tagging system employed on blog articles, its data also comes in handy.

1.6.3 Spam

As blog search engines become more reliable and more widely used, it becomes increasingly attractive for web site or blog owners to try to increase their search rank. This is done by specifically targeting the criterions search engines use to rank their site (see last subsection).

To this end, malicious users create so-called *spam blogs* (or "splogs" for short) that serve no other purpose than to boost the ranking of their blog or site. This is done by

- Showing some content that might look relevant to a search engine. This is often done by copying content from other blogs or sites.
- Updating regularly.
- Linking with the keywords to the page to boost.
- Getting linked to by other spam blogs.

As a search engine can not tell wether the creators of two blogs are related, a naïve search implementation can easily be fooled by large networks of such spam blogs. To counter these trends, modern search engines use a number of techniques to detect and filter out such deception attempts, for example by checking for duplicated content or applying machine learning (see [10]).

1.7 Blogosphere Structure

1.7.1 Typology

Because of its openness, the Blogosphere is by no means homogeneous. In fact, there are various criterions blogs are categorized by (see also [7]):

- Authors: The stated author might be a real person, writing his personal blog under his own identity. On the other hand, companies also increasingly use blogs for publishing PR content, thus the writer might actually speak for an organizational entity (*corporate blogs*).

  In extreme cases, an author might even decide to disclose his identity completely – for example because he or she fears the consequences of an association with the posted content.

  Another option is to not have a fixed set of authors at all. For content, these web logs accept contributions from web users and employ automatic or manual filtering in order to decide what actually appears on the blog.

- Form of blog articles: The actual content posted on a blog can also take many different forms. Typical blog article texts can range from few words

to many pages of text. Especially the practice to publish really short blog entries (*micro-blogging*) has gained considerable popularity through specialized services like Twitter.

Many blogs even do not publish textual content anymore, but concentrate on other media like audio (*podcasts*) or video (*video blogs*). Others might even only publish links.

- Function: Not every blog posts new original content. Many blogs instead concentrate on filtering and organizing data from other blogs or web sites. The own contribution is often limited to a short description or comment about the referenced entity.

  These "hub" blogs are actually a vital part of the Blogosphere as they help to structure the blog world and provide valuable pointers for users not familiar with the topical structure of the blogs surrounding a given topic.

- Topic: The real strength of Blogosphere is obviously the variety of its topics. Blogs can talk about just about anything. The range spans everything from people just writing about whatever comes to their minds on their personal blogs to highly specialized sites like "watch-blogs" that narrow their activity to the monitoring of one declared entity.

1.7.2 Static Topology

Despite its unregulated nature, the Blogosphere is to a remarkable degree self-organizing. This stems from the fact that given the size of all blogs summarized, nobody can really be aware of all relevant content. This is especially true for the blog writers: Each one will only have a limited set of content sources, many of which might actually be other blogs.

On one hand, this is obviously a good thing, as the collective of blog authors can keep track of a lot more than one writer alone. On the other hand, however, it also leads to the obvious effect that after a blog references a web site, other blogs are also more likely to feature the link. Once the blog "buzz" has reached enough momentum, the actual quality of the linked entity can quickly start to be irrelevant. Better alternatives might get drowned out because their attention failed to reach the critical mass.

This "Winner takes it all"-effect is reflected mathematically by the *Power Law*. In this case, the law predicts that the attention level will decrease exponentially with the blog rank. This means that few "A-list" blogs will receive an unproportional large amount of the traffic, gaining in turn considerable power over the minds of blog readers.

Some even see this as a danger to the democratic ideals of the blogging community and are searching for ways to diminish it.On the other hand, it must be noted that Power Laws have been observed in a lot of areas of social networking, so it might be human nature to work this way.

1.7.3 Dynamic Topology

The unique property of blogs in comparison to classic Web content is that it doesn't see itself as a static entity that must be changed in order to evolve.

Instead, the whole blog philosophy is that the current state is always defined by new posts and references appearing while older posts are slowly falling out of the public focus.

This means that the structure of the Blogosphere is highly dynamic and can change quickly. When a new topic emerges, discussion might start from a limited set of blogs and continue to spawn discussion on other web logs, creating additional cross links between the participants. After a while, the attention dies away and blog interconnections start to weaken accordingly.

Such time-limited activity surges in message streams have been termed *bursts* by Kleinberg [8]. Kumar et al [11] first applied the concept to the Blogosphere. This mechanisms can be used to extract interesting temporal properties from the Blogosphere. BlogScope, for example, automates the detection of such bursts and attempts to identify the defining keywords [2].

Chapter 2

Motivation

One of the interesting properties of the Blogosphere is that it is open by design. Just as everyone is free to participate in ongoing discussions, one can also just watch it unfold. In theory, this allows a deep look into the minds of a lot of people.

This data is obviously very interesting and useful for a lot of different purposes. Wether a person is interested in the newest information about the topic, just trying to track the mood, or looking to jump into the discussion himself, keeping up with trends in the Blogosphere is equally desirable.

Social media marketing experts, for example, already actively monitor the blog network for reactions to their products. This allows them to quickly assess public reaction to their communication. Tech-savvy marketeers could even write a corporate blogs themselves as a tool to quickly interact with the Blogosphere, for example by publishing corrections when a rumor starts spreading.

To detect situations like this, a user needs a quick view where the current "hot spots" within a given topic are. Standard blog search engines for the most part only allow searching for blog articles sorted by date and/or a general authority score like PageRank (Google) or Authority (Technorati). This is of no immediate help.

In this work, we will therefore aim to evaluate methods how to gather article data and produce better views on the Blogosphere trends by looking at the recent developments around our topic.

Chapter 3

Methodology

In order to identify current trends in the discussion surrounding a given topic, we choose a time interval and collect a set of blog articles that were written in this time. In a light-weight approach, we use existing blog search engines for this task.

The resulting link set is then analyzed and extended as needed in order to build a network that captures interdependencies inside the group in a meaningful way. This means that we are interested in information about link structures between articles and the publishing times of entries.

The result is a network with blog articles as nodes and directed edges connecting entries referencing each other. Edge weights are given by post time differences. This graph can then be subjected to a number of network analysis methods that help the user identify interesting trends in the observed part of the Blogosphere.

The following sections will look at the details of each step and evaluate the presented strategies where possible.

3.1 Blog Search Services

The task of finding recent blog articles talking about a topic is a pretty complicated one and requires large-scale automated collection of data on the Web.

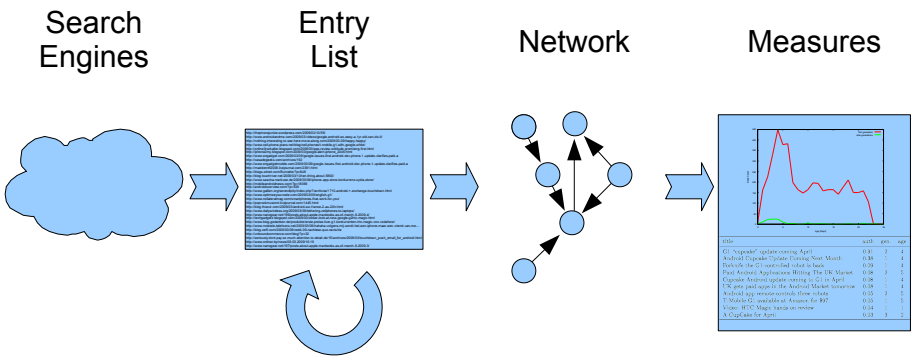

Figure 3.1: Methodology Overview

It is further complicated by the problem that blog articles age quickly, so ping servers (see 1.6.1) need to be employed to keep the index up-to-date.

On the other hand, this work is already done professionally by blog search engines. Providing the required data should be easy given their existing data base. Consequently, we will aim to use a set of blog search engines to provide us with the initial blog article set.

3.1.1 Criterions

In order to measure the acceptability of a blog search service, we must first state what we indent to ask of it. As we are interested in recent blog articles around a topic, we will use a query of the following form:

> What are all blog articles concerning topic P that were written and a time window T?

This means informally:

- We want *blog articles*. For now, a blog article at minimum has a perma link, a time stamp and textual content that can be scanned for links. Forum posts do not qualify for lack of proper perma links, for example.
- The articles must be about our topic. Even if we managed to give an accurate topic description, search engines might still fail to detect the topic of a blog article. This happens when the search engine looks outside the article text for matches – for example in comments or navigation links.
- Another dangerous problem is that search engines might fall for spam blogs entries (see 1.6.3) that only pretend to be talking about the subject at hand. Search engines employ a large set of heuristics to battle this problem, which might work more or less well depending on the amount of effort on both sides.
- On the other hand, we obviously want search engines to return a mostly complete set of all matching articles written. This not only means that the service needs high coverage. Another important factor is the possibility of the spam blog detection weeding out legitimate blog articles afterwards.
- Additionally, we want the blog articles to be from our given time period. This means that the blog search engine must have a time stamp for all scanned entries. Additionally, the blog search engine must be up-to-date in order to deal with queries concerning the recent past – which will be typical as we are mostly interested in new developments.
- Lastly, the blog search service must obviously be able to understand our query. This means that we can communicate our topic and time constraints to the search engine in a meaningful way.

  For most search engines, topic constraints are expressed by giving keywords that must appear in all returned results. There are obviously some pitfalls to be avoided: The description could be ambiguous. A search for "Volt" might for example produce results about electronics when the user was actually interested in the new car model.

	Technorati	Google	Bloglines	Icerocket	BlogPulse
time	(✓)	✓	✓	✓	✓
total	220	447	197	480	318
good	194	226	116	449	272
off topic	7	196	6	6	10
invalid	19	25	75	25	36
coverage	25%	29%	15%	57%	35%

Table 3.1: blog search engine evaluation

3.1.2 Evaluation

In order to get a better view of the quality of blog search services, we tested them using an example query. The results where then listed and manually checked for the above-mentioned criterions.

The query used for this section is "Henrietta Hughes". This query was used because it is the name of a homeless women that publicly spoke to the president of the United States of America, asking him to help her. The event and the reactions sparked some discussion on a lot of political blogs.

This example was chosen mainly because the discussion has clear-cut time margins: The incident happened exactly one week before this test was conducted. Furthermore her name was not published prior to the event, so it should be expected that this test should indeed return *all* sources that refer to her. It is therefore a good example to compare search services to each other.

3.1.3 Results

The resulting data is given in table 3.1. All search engines combined returned 781 unique blog article URIs. This set is taken as the baseline for measuring the coverage of blog search engines – which is surprisingly low overall, especially for the bigger search engines. A reason for this might be that search engines try to remove duplicates from their search results. As we aim to gain a complete view of the blog network, this behavior is undesired.

It must also be concluded that the aggregated blog article set most likely will not contain all articles. The given coverage measure is therefore only a very rough upper limit.

The bad sources are further categorized by what criterion was violated. Results were marked "off topic" if they were spam blogs (generated) or talked about a different topic. This includes the case were the topic is only mentioned in comments or elsewhere on the web site. It is interesting to note that only very few spam blogs appeared in our test. This is most likely a side-effect of the relatively fresh topic.

Entries were judged "invalid" if the page was not reachable using the given URI, where written in a language that made manual topic detection impossible, or where not a blog article at all (e.g. a blog list page).

In the following sections, we will introduce the tested search engines and explain the results in more detail.

3.1.4 Technorati

Technorati is the most established blog search engines on the Web today – and one of the first engines to concentrate purely on blogs.

Search results are ranked by Technorati using not only per-article measures, but also by the reputation of the containing blog. This reputation is represented as a numerical "Authority" score. This score is calculated by counting the number of unique blogs that have linked to the blog in the (recent) past. Thus blogs with high incoming link counts are likely to have a high authority from the point of view of Technorati.

Another concept pioneered by Technorati is the usage of tags (see 1.4.3). The author of a blog article is free to associate his posts with whatever tags he sees fit. The crawler checks all visited entries for these tags and uses them later as a high-priority clue for what the article is about. Technorati also offers special pages that group together blog activity around a given tag, complete with user-contributed explanations of what entries with this tag are about.

Technorati shows a mediocre performance in our test, only returning one out of four entries from our reference set. This might be a side effect of the increasing amount of spam plaguing the blog networks. As Technorati is a popular target, it is likely their spam detection algorithm is forced to operate very selectively.

Furthermore, Technorati does not seem to allow to search for entries from a given time period. This can be sidestepped, though, by exploiting that Technorati allows to sort results by publishing date. So the spider can just stop reading the list at the first post that falls outside our time interval.

3.1.5 Google Blogsearch

Google has been the biggest player in the Web search market for quite some time and has lately expanded into a lot of other areas – from E-Mail and map services to blog searching. Consequentially, their blog search service also liberally uses data from other services.

This means, for example, that a much larger data set can be scanned for incoming links, so attention from outside the Blogosphere can be detected, too. This applies not only to the Web as crawled by their Web search service, but also other available data like groups and potentially even chat and E-Mail exchanged using their services [5]. In fact, they can directly use the known PageRank values of blogs as an authority measure similar to Technorati's Authority.

In our test, Google showed severe problems with detecting the topic of blog articles. A lot of the results did not talk about the topic at all. The search engine was often fooled by navigation links pointing to other blog articles. Sometimes there was even no visible connection to the topic at all, which might be artifacts of small web page changes.

3.1.6 Bloglines and Ask.com

Bloglines is one of the oldest and biggest feed aggregator services on the Web (see 1.5.4). As such, it allows registered users to subscribe to feeds and presents the user with a personalized report on all current activity on his blogs of interest.

As an additional service, Bloglines also provides blog searching. For this task they can rely heavily on data about their user's blog subscriptions. As

this is relatively hard to manipulate, it could make Bloglines' blog search more resistant against spam blogs. Bloglines was later acquired by the Web search engine Ask.com (former Ask Jeeves) that now features Bloglines' blog search engine on their Web site.

For our test query, Bloglines performed pretty bad, returning the least total results of all blog search engines. Of these results, a lot were actually redirects through intermediate services like Google's Feedproxy or FeedBurner, with the direct links to the actual blog articles often appearing elsewhere in the search results. Furthermore, Bloglines never seemed to return more than 200 unique results, even when claiming to have found more.

3.1.7 Icerocket

Icerocket is a relatively unknown blog search project by the American billionaire Mark Cuban. Little is known about their underlying technology, though their web site offers a wide variety of search-related services like a page showing entries concerning a topic appear near-realtime.

For our test query, Icerocket performed really well, returning not only the most total results, the most good results, but also having the best rate.

3.1.8 BlogPulse

BlogPulse specializes in detecting certain trends in the Blogosphere. This is achieved by analyzing blog articles for key words and phrases that may describe the topic. The collected data can then be used to perform a number of different analyses, for example topic popularity graphs [6].

On the other hand, BlogPulse also offers standard blog searching as required for our test. It also compares favorably to the other search engines with the second-best coverage of 35%.

3.2 Blog Article Processing

In order to gain information about the blog discussions, the found blog articles must be analyzed automatically. For this work, we are especially interested in the dynamic aspects of the discussion. We will try to collect for each article:

- The time when it was posted
- References to other blog articles

Even though this is pretty basic information, it is still a challenging task to build a reliable collection method. The reason for this is that blogs are only defined by how people *use* them (compare 1.4.5). The technical side of blogs is only standardized to the bare minimum – a couple of convention that have managed to gain more or less popularity over the years.

Consequently, it is not enough to settle for only one information source when analyzing blog articles. Instead, a number of heuristics must be employed in order to get results with an acceptable reliability. For this, there are three mayor information sources: The results from the search engines, the blog article web pages, and the associated feeds (see Figure 3.3). The following subsections will look into the details of extracting useful information from those.

Figure 3.2: Blog Analysis

3.2.1 Search Engines

blog search engine generally provide the user with additional information about the blog articles found. Especially blog search services often give a rough time stamp for each returned results in the style of "found 2 days ago". This can be seen as an estimate for the time the article was posted, so parsing it will provide basic timing information.

Additionally, engines often show the user an excerpt of the found article to allow identification of the context the search term was used in. This is also valuable information as it is guaranteed to be text from the part of the Web page that the engine deems its "main matter". This will get useful in further analysis.

3.2.2 Article Source Code

For an article to be viewable in the Word Wide Web, it needs to be encoded using HTML. This format standardized is to a some degree and can be parsed. In theory, this makes all information accessible that is also visible to an human visitor.

In practice, the extraction of the relevant data is not trivial. The reason for this is that HTML is mostly a representation format and does not directly specify what the function of each page element is. If a link appears, it could just be meant for navigating to the next article. An automatic process will not be able to tell for sure.

So while scanning the source is the only definite way to get all links, it is also very prone to generate false positives. Large blogs might have a few hundreds of links, of which only a handful are really references from the article body. Following these links without additional consideration will most likely produce more noise than usable data. Especially when one takes into consideration that a lot of blogs recommend own articles for further reading – which will obviously point to legitimate blog articles.

3.2.3 Article Source Fragments

To make the link search more specific, it obviously helps to narrow down the space searched. For this task, we need information about the location of the main message body in the page's source code. This is the point where even tiny document pieces as provided by search engines come in useful: By searching for the given phrases, we get a good idea about where the actual content resides.

Only searching a small environment around the match can increase the effectiveness greatly, even though it is still prone to all problems mentioned in the last subsection. This is especially true as recommendations are often incorporated into the article page directly below the main article text. To get around this problem, custom-tailored solutions must be employed, like stopping the search when certain phrases appear ("Related Stories/Articles/Posts").

Furthermore, this heuristic might even perform worse by excluding relevant parts of the document. This might happen if the article is prepended with a summary of itself. On the other hand, this might even turn out to be desirable for long articles talking about a diverse field of topics.

Excerpt "... Lorem ipsum dolore. At vero eos et accusam ..."

```
<div class="content">
 <h1>Lorem</h3>
 <p class="summary">Lorem Ipsum</p>
 <p>Lorem ipsum <a...>dolore</a>.</p>
 <p>At vero eos et accusam.</p>
 <h2>Related Articles</h2>
 <p><a...>Dolore Magna</a></p>
</div>
```

Figure 3.3: Fragment Detection Example

Algorithm

The concrete implemented algorithm works on a DOM-style tree representation of the document's HTML data. It involves three steps:

1. First, the excerpt is split into words. We then gather for each word all occurrences in the document tree and save the respective nodes. If a word was not found, the word is removed from consideration.

2. Next we try to find the lowest node that is parent to at least one of the occurrence nodes of each word. To do this, we maintain a set of parent node candidates that is initialized using the occurrences of an arbitrary word from the excerpt. It helps efficiency to select a word with few occurrences.

   We then loop through all words. If it is detected that one of our candidates is not parent to any of the occurrence nodes, we replace it by its own parent node until it contains the word at least once.

   If in this process one candidate becomes parent of another at some point, we can safely remove the higher node from the set.

3. After the iteration, we have one or more nodes that contain the occurrences of all words from the excerpt. We can now go through the subtrees starting at these nodes and look for link tags.

   If we encounter a text node that matches one of our list of end-markers, we cancel the search. Note that we should require a match of the whole text node content. This makes it less likely that we might find a match in the text body, though we might still be fooled by code like:
   `<p>‘<b>Related Stories</b>’ is still problematic...</p>`

It is also important to note that the fragment heuristic might at worst degrade to a full text search, as once we have the root node as candidate, we will not change the set anymore. We will later have a closer look at how often this happens in practice.

3.2.4 Feeds

It is one of the few conventions that really caught on within the Blogosphere – that all blogs should have a feed to allow monitoring of its contents (see 1.5.4). It is also not difficult to locate feeds relating to a web site, as the web page can simply be scanned for appropriate links. Ideally, the site will even use the

	engine	feed	fragment	full text
stamp from	2.293	984	-	-
excerpt from	2.293	989	-	-
links from	-	729	1.232	2.287
link count	-	5.711	8.523	195.931

Table 3.2: Blog Article Analysis Statistic

appropriate `link` tag. This makes feeds a promising entry point for automated analysis.

As already mentioned, feeds are primarily meant to allow automated feed readers to stay up-to-date about information appearing on the site. To allow this, blog feeds must provide the most relevant information about current blog articles using a known format like RSS or Atom. In most cases this encompasses the full or abridged contents and a time stamp for each entry. This is obviously very valuable for our purpose. Even if the feed summary does not contain the full text or links are stripped, we can use the excerpt to guide fragment detection (see previous subsection).

On the other hand, feeds have one big disadvantage: As they only aim to publish updates, they will only contain a small set of the most current articles. This means that only very recent entries can be examined using this method. Additionally, the usefulness of this method diminishes for blogs with high traffic. This is obviously a problem as the most active blogs also most likely contribute a good chunk of to the discussion we are trying to track.

3.2.5 Evaluation

To evaluate the effectiveness of the presented approaches, we used the search query "Android G1", which at the time of this writing referenced the first and only mobile phone using the Android platform designed by Google. The maximum age for search results was set to 21 days. As there was steady discussion about this topic within the Blogosphere, this will most likely only comprise a part of the ongoing discussion.

The query resulted in 2795 results from the four introduced blog search, which means we excluded Google because of its poor performance in the search engine evaluation. After applying the filtering algorithms from the next section, this set was somewhat reduced to 2293 articles (for details see 3.3). A statistic of the quantity of information extracted is shown in Table 3.2.

As shown, all search engines returned some sort of time stamp and an excerpt with each results. On the other hand, for only 989 articles (about 43%) our method was able to locate a corresponding feed entry. Of these 989 feed entries, in turn only 729 contained links. Under the pessimistic assumption that almost all blog articles have links, we therefore have solid link data for only about 32% of the valid search results.

This is low enough to justify the usage of the fragment heuristic. Let us have a look at the link statistic. It is important to note that the table only lists *new* links. So if the fragment did not contain any links that were not already found in the feed, it will not count for either row. As we have excerpts for all

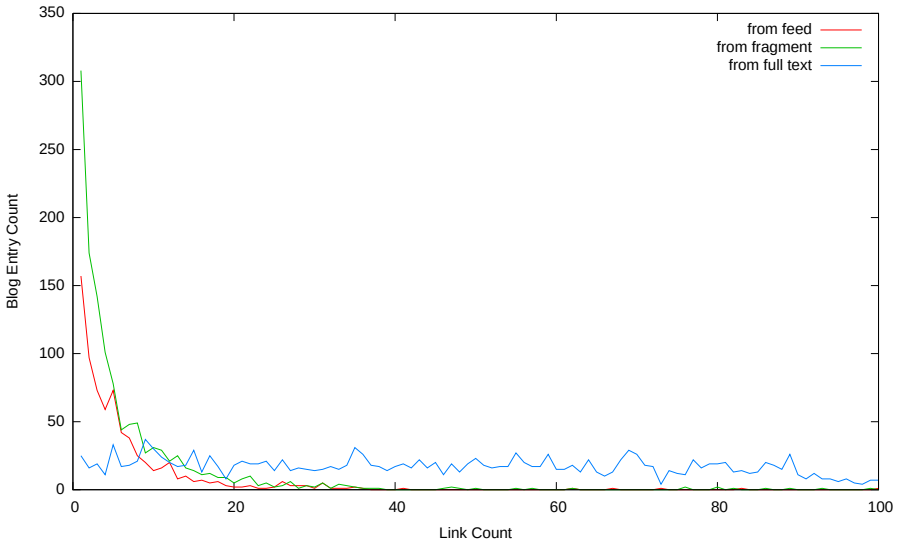

Figure 3.4: Link Sources

pages found, the fragment detection algorithm was also applied on all results. This means that for the remainder of the 1217 pages listed, either

- there were no links in the fragment or
- all links were already known. There are actually 464 articles for which there where feed links, but no fragment links. It is likely that both heuristics produced the same link set. This also means that feeds and fragment combined managed to extract links from about 1696 articles (73%).

Additionally, the low relation of fragment links found per page with fragment links suggests that the fragment in many cases only added a few extra links. The data in Figure 3.4 supports this conclusion.

The graph also shows that fragment link count and feed link count behave very similarly when compared to the count of links typically extracted from the full document. This suggests that our fragment heuristic indeed does not often use a too big chunk of the document.

3.3 Iteration

As already noted in the section on blog search engine evaluation (see 3.1.2), it must be expected that the search engine results only cover a subset of the articles. In general, we can even suspect that we might be missing relevant results that were linked from sits from our network.

3.3.1 Link Quality Filtering

To counter this problem, we replicate part of the search engine's job and follow some links to web pages that were not in the set that was initially returned. As we can not visit all available links, we have to select those that look promising.

In order to select these links, we score each link depending on where it was found. We arbitrarily assign a score of 1.0 for links found in the feed and 0.5 or 0.1 for links found in the page source code, depending on wether the respective link was what we determined to be the body fragment.

A link target is then scored with the sum of the scores of all links pointing to it. We then use some arbitrary score threshold like 2.0 to form an initial set of "promising" links.

It should be noted at this point that it in practice the task of deciding wether two links point at the same page is actually non-trivial due to redirects. These are unfortunately pretty common within the Blogosphere due to services like Google's Feedburner that channel a lot of blog access through their links. As it is impractical to follow all links to check if and where the redirect occurs, we must live with some bias to services without redirection.

3.3.2 Topic Filtering

After we have selected a set of links to follow, we obviously do not really know what these links are actually pointing to. Because of the link score, we have a vague hope that there might be relevant posts in there. On the other hand, we can be confident that there will also be a lot of junk.

Detecting the topic of a web page is, in general, a fairly complicated problem where sophisticated algorithms exist. For our purposes, we only do a basic full text match. This is reasonable as we can assume that our topic description is already in a form that is optimized for this kind of matching – as it was already a requirement for the blog search engines to produce meaningful results in the first step.

3.3.3 Blog Filtering

Finally, we are only interested in blog articles. This is obviously more complicated, as we first need to have an idea of what we are willing to call a blog article. In fact, when we introduced the blog concept (see 1.4.5), we sidestepped giving a proper technical definition. By our description, a blog can be anything were posts can appear that have a proper mechanism for linking to them.

The reason for this is that we are trying to analyze blog discussions. And almost by definition, any website is free to join such a discussion – as long as the contributions are linkable. So in order to get a complete view of the discussion process, we want to be equally liberal.

Consequentially, we will have to rely on heuristics. The strongest indication that a site might be post-oriented is the existence of a feed (see 1.5.4). This is because feeds are typically used to track activity on a web site. This activity, in turn, often takes the form of posts appearing.

This is obviously a very rough heuristic and needs additional mechanisms to produce a decent detection rate. The following errors are possible:

- The site might not be post-oriented. Wikis, for example, might allow users to track changes using a feed. Fortunately, these pages do not seem to be linked too often.

- Even if the link does point to a page from a blog-style web site, it is not guaranteed that we actually have a blog article page. Blogs often have

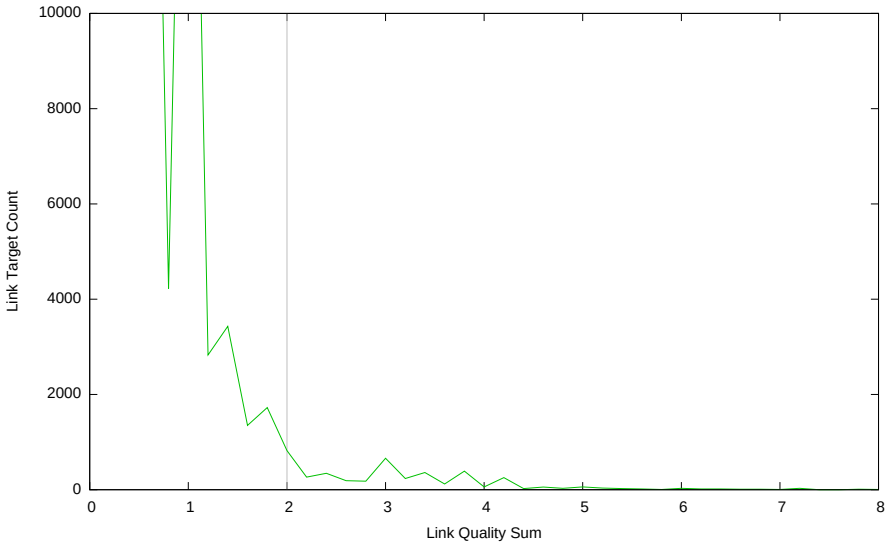

Figure 3.5: Link Quality Frequency

special pages that provide navigational help and information about the blog. The page most typically linked is the main blog page, which can be easily detected as it is the root of the whole site in most cases. Others, like category or tag pages, are harder to filter and again require hand-made filter rules.

3.3.4 Iteration Process

After applying the presented filters, we end with a new set of links. As we tried our best to check most of the criterions manually, we consider to be probable candidates to fit into our query. To distinguish the new links from those that were returned by the search engines, we will from now on call them "generation 2" and "generation 1" links, respectively.

The obvious next step is to again apply our analysis methods to extract link and time-stamp data from the new link generation. This, in turn, might improve link quality of some link targets, resulting in another generation of blog articles.

This process can be iterated until we get a stable set of links where every link target above the given quality threshold was visited. The threshold should be chosen so that this happens quickly, as we do not want the iteration process to last too long.

3.3.5 Evaluation

This evaluation uses the data from the same query that was used in the last section (see 3.2.5). That is, a search for "Android G1" spanning 21 days.

The link quality threshold used was 2.0. This means that there need to be 2 high-quality links from a feed, 4 mid-quality links from fragments or 20 low-quality links found by full text search in order for a link target to get visited.

gen	total	good		off topic		no blog		invalid	
1	2.794	2.293	82%	73	3%	316	11%	112	4%
2	835	158	19%	328	39%	333	40%	9	1%
3	213	48	22%	81	38%	79	37%	2	1%
4	27	19		5		2		1	
5	12	5		7		0		0	
6	1	1		0		0		0	

(the last 3 generations are too small for meaningful comparison)

Table 3.3: Filter Statistics Per Generation

Figure 4.2 shows why this value was chosen. Lower values seem to result in too high follow rates which quickly result in a snowballing behavior.In our case, this threshold resulted in 835 links being selected for the second generation.

After applying the presented iteration, we got the results shown in Table 3.3. We can see that each generation has roughly about 20% good links by our standards, the rest being off topic or not blog articles at about the same rate. Links were marked invalid when they could not be followed for technical reasons (like using a protocol other than HTTP or the web server not responding). Due to the fact that every new generation was substantially smaller than the previous one, the iteration died off quickly.

It is also interesting to look at the filter statistics for the results from the blog search engines (generation 1). The count of off-topic and non-blog results is at about the same level as found in the manual survey from section 3.1.2. On the other hand, it is quite surprising how many of the results had to be deemed invalid.

Chapter 4

Analysis

In the last sections, we have described how we can collect a set of blog articles for some given topic and annotate those with information about referential and temporal dependencies. In this section, we aim to process this information and collect statistics and measures that allow us to get a better feel for what actually was happening around the topic.

4.1 Graph Construction

The foundation for our analysis will be the *blog article graph*. This graph has one node for each blog article in the data set and an edge for each link that was detected.

Our base data set can be the result from an arbitrary analysis step. In this section, we will compare results from the first generation to those of the last generation. Additionally, we try to eliminate some noise:

- Web pages only get a node if they were not filtered and have a valid time stamp. This means we will now only include those results where we are really confident that they are at least similar to blog articles.
- Links only found using full-text search do not produce an edge.

We could try to us certainty values at nodes and edges instead, but unfortunately, this complicates work with some of the algorithms we are going to use. Note that by eliminating nodes without time-stamp we might substantially reduce the number of usable results from later generations. This is because we did not find those using a search engine, so the sole source for stamp data is the unreliable feed entry matching method (see 3.2.4).

4.2 Basic Graph Characteristics

Table 4.1 shows some basic properties of the generated graph for our running example from sections 3.3.5 and 3.2.5. Of the 231 additional nodes listed in subsection 3.3.5 only 73 (32%) had a time-stamp and could be taken over into the graph as nodes. This rate is somewhat lower than our result from section

	nodes	edges	inclusiveness	mean degree
first generation	2292	409	0.201	0.178
all generations	2365	634	0.251	0.268

Table 4.1: Blog Article Graph Characteristics

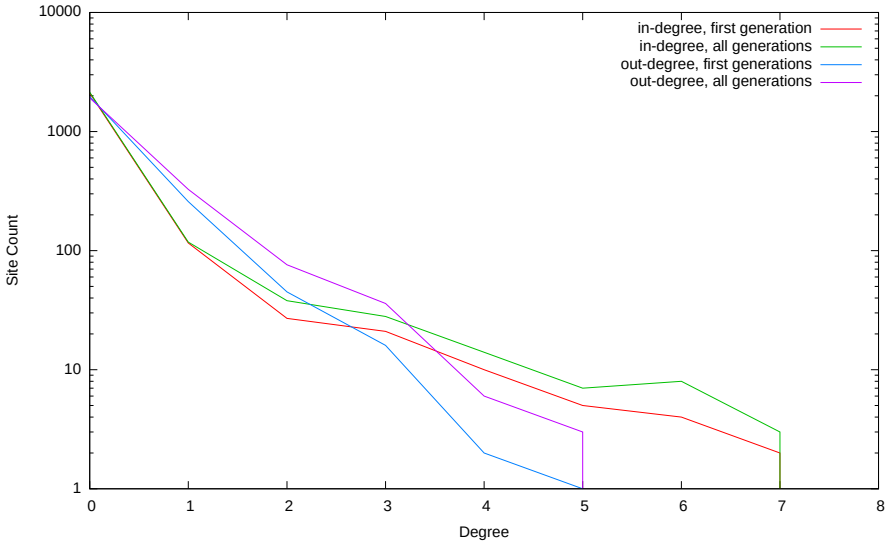

Figure 4.1: Node degrees

3.2.5 (43%) and hints that our blog article detection method might be worse than the one used by professional blog search engines.

Furthermore, it is clearly visible that we have a lot of isolated points (80% in the first generation) that do not connect to any other node from our graph. This changes somewhat for the later generations. The main reason is that we chose the newly added nodes indirectly by in-degree, so the iteration process tends to include highly connected nodes. The additional 73 nodes alone contribute over 200 edges, which pushes the mean degree quite a bit. A look at the in- and out-degree statistics in Figure 4.1 further supports this view.

4.2.1 Time Stamps

Another interesting basic statistic is the distribution of the article ages, as plotted in Figure 4.2. We see that both first generation and later generation articles show a peak at young articles. This could have a couple of reasons:

- For the first generation, it could be the case that blog search engines favor newer results and let older results fall out of their index faster. Bloglines, for example, seems to silently bound the number of results returned.

  This can also be backed up by looking at the average age of results with a certain popularity between search engines in Table 4.2. It is clearly visible that the age tends to decline for results that were returned by a lot of engines.

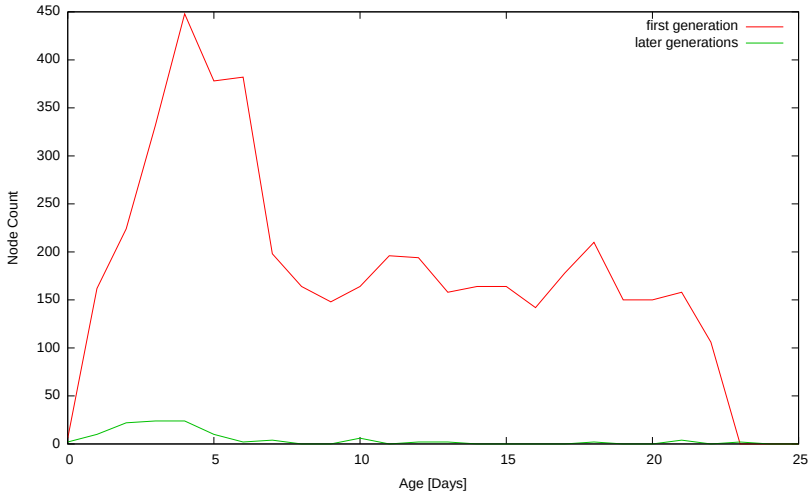

Figure 4.2: Node Ages

popularity:	1	2	3	4
average age:	9.85	7.69	4.28	2.14

Table 4.2: Age by Search Engine Popularity

- In later generations, the feed entry match method is more likely to obtain time stamps for newer results as feed entries age quickly (see 3.2.4).

  An alternate explanation could be that the peak from the first generation translates over into subsequent generations, though in practice some delay should obviously be expected, which does not seem to be present.

- Lastly, optimally we are actually looking at a peak of activity in the Blogosphere. As described, we can not be sure about our data set being representative with respect to time, so we can not tell for sure.

  On the other hand, the peak also show up in trend graphs generated using the trend analysis tools of BlogPulse and BlogScope, shown in Figure 4.3. The test run was conducted on 16th March, so the activity around 12th March matches nicely with our peak at the article age of 4 days.

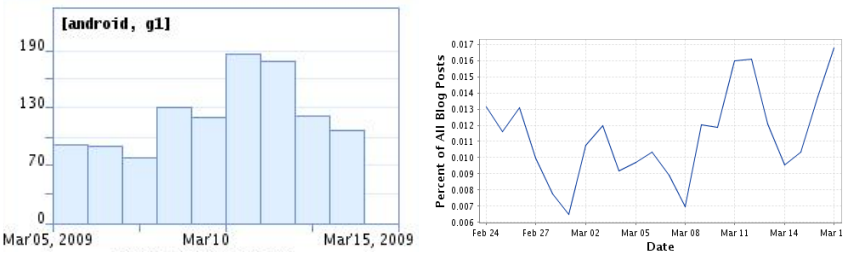

Figure 4.3: Trend graphs from BlogScope and BlogPulse

title	in-deg	gen.	age
G1 “cupcake” update coming April	50	2	4
Android app remote-controls three robots	30	2	5
Google blocks paid apps for unlocked G1 users	21	1	18
Google blocking paid Market apps from Dev Phone 1	20	1	19
Android Dev Phone Update: Version 1.1!	16	1	6
Android Cupcake Update Coming Next Month	16	1	4
Paid Android Applications Hitting UK Market	15	2	5
Software Update Available for Android Dev. Phone	10	2	6
Google G1 Covers	9	2	133
Google Android Phone T-Mobile G1	9	2	145

Table 4.3: Top 10 Results by In-Degree

4.2.2 Central Nodes

Up to this point, the analysis focused on general properties of the graph that tell us mostly about how good a job the previous steps did. But the starting point for the whole analysis was that we wanted to know what was going on inside the Blogosphere.

Consequently, we will now try to supply the user of our analysis with data about what appears to us as the focal points of our network. The most basic method is to look for articles that have a lot of incoming links – many blog search engines also use this as a basic quality measure for their results.

When we search for such nodes in the graph for our example, we get the top candidates listed in Table 4.3. We can make a number of observations about this list:

- Just from looking at article titles and ages we can guess that the reason for the spike in the date graphs was actually an update to G1 going by the name “Cupcake”.
- A lot of the articles are from generation 2. On one hand, it is obviously not surprising that the new nodes have high in-degree, considering the selection algorithm. On the other hand, there seems to be no good reason why this results should not have been returned by search engines in the first place.
- Two of the results appear to be way out of our time window. This must be attributed to our lack of date filtering. As older articles have a higher probability of having incoming links, this has the potential to skew the results.

Another interesting read is the list of nodes with high out-degree, shown in Table 4.4, albeit again for implementation reasons. We can see that later generations appear here, too, which is good news in terms of the overall effect of the iteration on connectivity.

Apart from that, we can see that we spidered at least two automatically-generated blogs – the “Engadget Mobile” article duplicated contents from the site of the same name. And Techmeme generally only publishes automatically generated link lists.

title	out-deg	gen.	age
Engadget Mobile	8	1	4
Android dev phone gets Android 1.1 update	5	1	5
Techmeme Top Items March 11	5	1	3
How To Import ... Using Your T-Mobile G1	5	4	80
T-Mobile Launching Cupcake Update for G1 in April	4	1	3
anroid-apps (sic) für das t-mobile g1	4	1	21
Anroid-Apps (sic) für das T-Mobile G1	4	1	20
Top 10 Most Expensive Google Android Apps	4	1	17
Pre Browser Almost 4x Faster than iPhone?	4	2	3
7 Apps And Feature Updates I Want For Android	4	3	78

Table 4.4: Top 10 Results by Out-Degree

4.3 Hyperlink-Induced Topic Search

Judging node quality just by in- or out-degree is not a very sophisticated way to find good nodes. This is because it ignores a lot of known information about the network. Kleinberg [9] proposed a different approach. His method was meant to be used for web search, but the core algorithms can be adapted for our use.

The core idea is that not all links or graph edges are equal. A link from an article that is talking about a topic in-depth is more significant then a link from an article that only mentions the link target "by the way". The trick is that we can measure this, again, using our information about the link structure: If the article mentions many good articles, it is likely to be a good one itself.

This leads to Kleinberg's two node scores for "hub" and "authority" quality, that are recursively defined by:

- The hub score is the sum of the authority values of all outgoing edges
- The authority score is the sum of the hub values of all incoming edges

These values can be determined simply by starting with a value of 1 for each node and quality value and then in alternation adjusting the hub and authority values to adhere to their respective definition. If the values get normalized after every step, it can be proved that the values will eventually settle.

4.3.1 Application

When applied to our example article graph, we get the results listed in Table 4.5. Most of the article titles look familiar, as they also appeared prominently in the in-degree-table (Table 4.3). This is hardly surprising as Kleinberg's algorithm obviously generally still favors nodes with high in-degree.

The main difference of the HITs algorithm is that it counts how the referencing nodes are connected to the rest of the network. Consequently, we should expect priorities to shift towards topics where articles are more densely connected to each other.

The best-connected topic in our case is, not surprisingly, the "Cupcake" update again. Here it manages to take the best three spots by a considerable distance. The "Android robot" subtopic is pushed down a bit, but manages

title	auth	gen.	age
G1 “cupcake” update coming April	0.91	2	4
Android Cupcake Update Coming Next Month	0.35	1	4
Cupcake Android update coming to G1 in April	0.13	1	4
Vodafone HTC Magic Details Surface	0.11	2	28
Android app remote-controls three robots	0.06	2	5
Cupcake Android update coming to G1 in April	0.05	1	4
Paid Android Applications Hitting The UK Market	0.04	2	5
Forknife the G1-controlled robot is back	0.03	1	4
UK report on Android Cupcake is ‘just a rumor’	0.03	4	3
UK gets paid apps in the Android Market	0.03	1	4

Table 4.5: Top 10 Results by Authority

title	auth	gen.	age
G1 “cupcake” update coming April	0.91	2	4
Android Cupcake Update Coming Next Month	0.38	1	4
Forknife the G1-controlled robot is back	0.09	1	4
Paid Android Applications Hitting The UK Market	0.08	2	5
Cupcake Android update coming to G1 in April	0.08	1	4
UK gets paid apps in the Android Market tomorrow	0.08	1	4
Android app remote-controls three robots	0.05	2	5
T-Mobile G1 available at Amazon for $97	0.05	1	5
Video: HTC Magic hands on review	0.04	1	1
A CupCake for April	0.03	3	2

Table 4.6: Top 10 Results by Authority in Weighted Network

to place another relevant article in the top 10 list. Apart from the HTC magic result, it is now clearly visible that the list is dominated by three main subtopics.

4.3.2 Weighting

The presented algorithm still somewhat ignores the dynamic nature of the discussion. In fact, we still have one bit of information from the network that can help the algorithm in finding the interesting nodes: The time stamps.

To make this more explicit, we are interested in discussions which recently started and are gaining momentum quickly. This can be achieved by modifying the HITS algorithm to apply weights to each edge. For this to work, we must only replace the authority and hub value sums with the appropriate weighted sums.

We choose the weights to decrease with the age of the edge target. This has the obvious effect of favoring younger nodes. Additionally, we indirectly push nodes that have a lot of surrounding recent activity, as hubs must be younger and also point at recent articles to build their hub score. Thus, we will ideally detect the initial push when a key article starts to circulate through the blogs.

Table 4.6 shows this effect: Though the recent spike(s) at article age 4 still dominates the list, we also get fresher articles like the HTC video review.

Chapter 5

Conclusion

To close this work, let us review what has been done and what is still missing. We did manage to produce a meaningful article graph starting from the results of blog search engines. We evaluated each step in order to validate that the used heuristics produce meaningful results.

At the end, we managed to produce a list of articles that seem to be a good representation of what was happening around the chosen topic at the time of generation. The weighted algorithm seems to be much more sensitive to quick developments – by time-filtering the graph it can actually be shown that it manages to detect the later top result within hours of publication.

What's obviously missing here is good way to quantify the quality of the results. Additionally, in practice we want the algorithm monitoring the Blogosphere continuously. This could be done by reading feeds from blog search engines to stay current about new articles and adding them to the data base on-the-fly. This might result in a usable and fast system for detecting activity spikes even as they are still in the process of building up.

Bibliography

[1] Chris Anderson. *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion, 2006.

[2] Nilesh Bansal and Nick Koudas. Searching the blogosphere. In *Proceedings of the 10th International Workshop on Web and. Databases, WebDB 2007*, Beijing, China, 2007.

[3] Michael K. Bergman. The deep Web: Surfacing hidden value. *Journal of Electronic Publishing*, 7(1), 2001.

[4] Tim Berners-Lee and Mark Fischetti. *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco, September 1999.

[5] Andriy Bihun, Jason Goldman, Alex Khesin, Vinod Marur, Eduardo Morales, and Jeff Reynar. US patent 20070061297: Ranking blog documents, September 2005.

[6] Natalie S. Glance, Matthew Hurst, and Takashi Tomokiyo. BlogPulse: Automated trend discovery for weblogs. In *WWW 2004 Workshop on the Weblogging Ecosystem*, New York, NY USA, May 2004. ACM.

[7] Susan C. Herring, Lois Ann Scheidt, Sabrina Bonus, and Elijah Wright. Bridging the gap: A genre analysis of weblogs. *Hawaii International Conference on System Sciences*, 4:40101b, 2004.

[8] Jon Kleinberg. Bursty and hierarchical structure in streams. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 91–101, New York, NY, USA, 2002. ACM Press.

[9] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[10] Pranam Kolari, Tim Finin, and Anupam Joshi. Svms for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[11] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of Blogspace. In *WWW '03: Proceedings of the twelfth international conference on World Wide Web*, pages 568–576. ACM Press, 2003.

[12] Tim O'Reilly. O'Reilly network: What is Web 2.0, September 2005.

[13] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.