Tracking Visitor Engagement in the Blogosphere for Leveraging Rankings

Patrick Hennig*, Philipp Berger*, Christoph Meinel‡
Maria Graber†, Jens Hildebrandt†, Stefan Lehmann† and Cathleen Ramson†
Hasso-Plattner-Institut
University of Potsdam, Germany
*Email: {patrick.hennig, philipp.berger}@hpi.uni-potsdam.de
†Email: {maria.graber, jens.hildebrandt,stefan.lehmann,cathleen.ramson}@student.hpi.uni-potsdam.de
‡Email: office-meinel@hpi.uni-potsdam.de

***Abstract*—Current blog search engines use rankings, such as BIImpact or B2Rank, focusing on the link structure and other meta criteria externally extracted for blogs. A good, but due to the unavailability, not often used criteria is the visitor engagement. This metric can leverage the quality of a ranking extremely. For this reason, we propose to gather visitor information from blog authors by providing a new blog plugin. This plugin on the one hand tracks the visitor information and on the other hand provides important analysis information for the blog author. Finally, this leads into a win-win situation for both, the blog search engine and the blogger. The benefit of this plugin is to provide analytics based on the blog where the plugin is installed as well as analytics based on the whole community of bloggers. With this information a blogger is able to gain significant knowledge advantage.**

I. Introduction

In the World Wide Web more than several hundred million blogs have been created over the last years. Due to this huge amount of blogs a big effort has been made to rank those blogs and Figure out what are the most important and most influencing blogs. Rankings such as BIImpact [1] or B2Rank [2] has been developed and evaluated on big data sets of the blogosphere. The authors were able to proof the need for a blog specific ranking. Nevertheless, they were not able to include the blog's visitor engagement in their blog rankings because these kind of information are not accessible via external crawling. Since this information is a very promising criteria for blog rankings, this paper presents an approach how visitor information can be retrieved from bloggers in an easy way.

This very difficult task is accomplished by developing a plugin that can be installed and used by bloggers. This plugin is able to retrieve anonymous visitor information from its host blog. Of course, it is very difficult to convince a blog author to use such a plugin. Hence, our plugin does not only retrieve visitor information, it also provides a wide range of very useful analytics and rich visualizations for each blog author based on the comprehensive knowledge of the underlying blog search engine. With this information a blog author gains a big advantage in knowledge for his own blog and his community. Finally this comes to a win-win situation of both, the blog author and the blog search engine.

For tracking the user information the plugin uses external visitor tracking services that will be replaced by more sophisticated visitor engagement measurement techniques in the future. The blogger is able to decide which service fits best to his infrastructure or his security needs. Finally, this anonymously gathered visitor data is integrated in the BlogIntelligence[1] BIImpact score.

To provide rich analytics for the blogger, data provided by BlogIntelligence are used within the plugin. Visualizations for a single blog as well as others based on the whole community help the blog author to decide about which topics he should continue writing or which topics will be hot during the next days. Further, our plugin provides the possibility to look at the whole community of a certain blog. E.g. how a blog performs during the last days compared to other blogs that are writing about similar topics. We integrate the plugin into the WordPress[2] blogging system. The user interface is embedded into the WordPress system as shown in Figure 1.

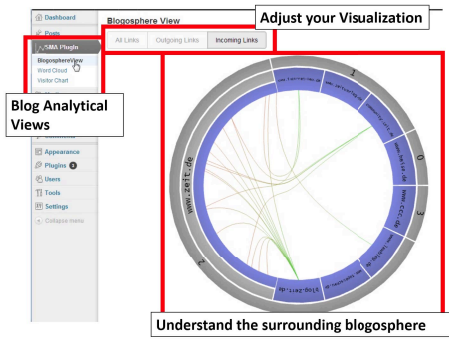

Fig. 1. Our Wordpress blog plugin offers blog analytics for the blogger and collects visitor data for blog search engines.

II. Project Scope

With a wide circulation of more than 200 million *weblogs* worldwide, weblogs with good reason are one of the most important data streams in the World Wide Web. Therefore, weblogs offer access to latest information discussed in the real

[1]http://www.blog-intelligence.com
[2]http://www.wordpress.com

978-0-7695-5137-1/13 $26.00 © 2013 IEEE
DOI 10.1109/SocialCom.2013.57

world. Since writing posts in weblogs goes along with a high editorial effort, the available information is of major interest. However, for a user it is becoming harder and harder to gain an overview of all discussions in the blogosphere. It is almost impossible for a user to extract information from the web, especially from the blogosphere. Hence, a system that collects information from the blogosphere and presents it to the user in a very meaningful way would be of great use.

Therefore, mining, analyzing, modeling and presenting this enormous amount of data is the overall aim of the project the presented work is integrated in. This enables the user to detect technical trends, political climates or news articles about a specific topic. Most approaches of mining and analyzing such a huge amount of data focus on offline algorithms which use pre-aggregated results. This is in contrast to the continuously growing nature of the World Wide Web. As a result, including the latest data is one of the key aspects of data mining on the web. This is exactly the topic covered by the *BlogIntelligence* project.

Ranking blogs and post is the major success factor of such an tool. Due to the unavailability, for the proposed ranking algorithm in BlogIntelligence [1] a last important aspect is missing. In particular this is the visitor engagement. Therefore the plugin described in this paper helps to improve this ranking by gathering the visitor engagement of blogs and as a consequence deliver real-time analysis for the blog authors

III. Related Work

This paper presents a Wordpress plugin that is designed to track visitors of a blog. The visitor statistics should support ranking of weblogs.

There are diverse blog ranking algorithms. For example, Berger [3] introduce a blog ranking based on the post text consistency. Although other approaches like Bross et al. [4] propose the visitor engagement as additional metric, they are mostly limited to one blog-like discussion website or lack of an approach to actually collect the necessary visitor data. Thus, we propose a plugin that can also be integrated in standalone blogs.

For tracking users, Peterson and Carrabis [5] specify a detailed metric for measuring the visitor's attention and interaction on a website, which they called Visitor Engagement . They use visitor statistics like session duration and page views per session. Likewise, Oberle et al. [6] track users by their query strings. They present a semantic enrichment of weblogs. They map the query string of the visitor to a feature vector that is used to model the visitor's behavior. This kind of user engagement measures are very complex, but also very promising. Nonetheless, we stick to simple access statistics at the first attempt.

Atterer et al. [7] present a method for user interaction tracking on websites. They use both an HTTP proxy to track the requested URLs and client-side scripting. The proxy adds to the requested website JavaScript code that is used to track mouse movements and the interaction of the visitor. Nevertheless, we currently stick to existing user tracking solution eg. Google Analytics and Slimstat.

A variety of research projects attempt to visualize the massive amount of data generated by social networks. The visualizations of choice therefore have to reveal high-level pattern in the given data.

The blogosphere is often represented as a node-link structure like described by Herring et al. [8]. In such visualizations blogs are represented as nodes while hyperlinks or other relationships are represented as edges. Most of these layouts are limited to a few hundred blog entities. Otherwise the layout would suer from edge cluttering.

To overcome this issue current visualizations introduce several kinds of semantic zooming. For example, only the most important blogs are shown in a map of the whole blogosphere. By zooming into certain parts of the map, less important but theme related blogs get visible. Relationships get only visible by selecting a certain object of the map. This reduces visual cluttering. These techniques can be found in Bross et al. [9].

Recently developed layout algorithms try to keep and even illustrate underlying patterns by aligning edges onto the high-level graph structure to reduce visual clutter. Force-directed edge bundling as described in Holten et al. [10] presents a self-organizing approach that models edges as elastic springs attracting each other. Despite its good results this approach is not quite applicable to web applications due to its high complexity.

Other layout approaches depend on additional information of the graph structure. Hierarchical information enables the usage of the hierarchical edge bundling technique proposed by Holten [11]. Hierarchical edge bundles illustrate the underlying pattern by bundling edges according to the hierarchical structure of the given blog data. Normally not applicable for general website structures, this approach can make use of the compositional relationship between blogs, posts and comments. Additionally, the retrieved blog cluster data by BlogIntelligence can be used to enhance the hierarchical structure of the blogosphere [3].

Another big topic in visualizing social network content deals with the visualization of trends. Since the blogosphere is considered as one of the fastest information spreading networks, a change in the popularity of certain topics can be observed there first. Identifying those changes is a valuable ability for companies as well as blog administrators. Therefore, trend visualizations have to illustrate changes in popularity for selected topics. By highlighting those changes valuable information can be generated.

As a result, the bigger part of traditional trend visualizations heavily relies on historical data to show the evolution of a trend over time. Themerivers for example is an alternative approach to stacked bar charts to visualize continuous data over time [12]. Although very powerful Themerivers share the same problem with many other line-based visualization types, they are very space consuming. Therefore, sparklines provide a minimalistic approach by visualizing a single metric over time for a given data item. This metric is depicted as a small line chart to integrate it in running text or visualize multiple sparklines [13]. SparkClouds combine this technique with traditional tag clouds to enrich each word with a sparkline to show additional temporal information [14]. On one hand this leads to a compact visualization for many data items, but

on the other hand there is a loose of comparability between single elements.

IV. Tracking

The main goal of the plugin is to collect visitor statistics. This chapter explains which data is provided by the developed plugin to rank a blog according to the number of visitors and which tracking techniques are used by the plugin.

To improve the blog ranking of BlogIntelligence, the plugin tracks information about how many users access a blog in a given time interval. Bross et al. [1] incorporate in their ranking mainly the number of visitors per time interval. Although more sophisticated measures of visitor engagement can easily be implemented, we focus in this paper on the measure that the authors highlight as important for the BIImpact score. Therefore, the plugin collects data about the number of visitors per blog. To ensure an accurate rating the plugin ignores recurrent visits of one visitor in the same time interval.

In addition to blog-level visits, the plugin also tracks the visitors on a post level. These information allow a detailed analysis of the interests of the visitors. For instance, this level of detail uncovers if the main part of the blog visitors only read some old posts. In contrast to these blogs, there are many blogs where the visitors are only interested in new posts. This could be supposed for news blogs. For ranking, these two blog types can be handled differently. In addition to these information, post-specific statistics allow to indicate the most and least popular topics.

A. Tracking techniques

To gather the described data, the plugin uses a tracking tool. There are some tools that are capable of gathering the required data. The plugin can track data using one of the two common tools: *Google Analytics* or *SlimStat*. Google Analytics collects visitor data and save it on servers provided by Google. SlimStat is a Wordpress plugin which saves visitor data in the local Wordpress database. For each tracked blog, only one of these two tracking alternatives is needed. By supporting both, the blog administrator can choose which alternative he prefers. Additionally, this makes our plugin easy portable to other platforms like Blogger.com[3].

1) Google Analytics: Google Analytics is a free statistic Web service provided by Google. It is currently the most popular statistic service for websites. If the service is used, it tracks the visitors of the website. To enable tracking, a page tag has to be included into every website, where users should be tracked. A page tag, also called tracking code, is a small JavaScript code fragment. [15]

The tracking code is executed in the browser of the visitor. It sends the necessary tracking data to Google. The tracking data is then stored on servers from Google. This data can be accessed by a website or with the Core Reporting API[4] provided by Google. To get the tracking data from Google the Core Reporting API is used.

Requests to the Core Reporting API consist of dimensions and metrics[5]. Metrics represent visitor statistics that are segmented by the dimension vector. Dimensions are common criteria like host name, date and country.

Access to the Core Reporting API is limited to 10 000 requests per day with a maximum of 10 concurrent[6]. One request can contain up to seven dimensions.

User-specific dimensions are possible using custom variables. The value of a custom variable is set in the tracking code. Custom variables can be accessed as dimensions over the Core Reporting API[7].

2) SlimStat: SlimStat is an open source analytics plugin for Wordpress[8]. If visitors click in the front-end, they are tracked and stored in the local Wordpress database.

SlimStat only stores a few attributes per click like the time, the URL, the browser and similar information. In exchange for these limited information they are specific to Wordpress blogs. For example, SlimStat can also store the content type of the website that was requested. Examples for the content type are *post* or *home*.

For distinguishing the visitors, SlimStat assigns each click to a visitor session. In the database each click is stored with a visit id. Two clicks that belong to the same session get the same visit id. This allows to count the total number of clicks as well as the actual number of visitor per page.

B. Comparison

Section IV-A explained the different technical insights for the two tracking techniques SlimStat and Google Analytics. This section discusses the advantages and disadvantages of both techniques.

a) Installation: SlimStat is a Wordpress plugin that has to be installed by the blog administrator. In contrast to SlimStat, the blog administrator does not need to install additional software to track the visitors with Google Analytics.

b) Data location and access: The visitor statistics of SlimStat are stored in the local Wordpress database. Therefore, the plugin can directly access the data. Whenever a visitor reads the blog, the visitor statistics of the BlogIntelligence database can be updated immediately. This leads to real time updates of the provided visualizations. In contrast to SlimStat, the visitor statistics of Google Analytics are stored on the Google servers. Consequently, real time updates of the database or the visualizations are not possible. Nevertheless, the statistics can be polled from the Google servers when they are needed. How the statistics from Google can be inquired, is explained in chapter V.

[3] Blogger.com

[4] https://developers.google.com/analytics/devguides/reporting/core/v3/

[5] https://developers.google.com/analytics/devguides/reporting/core/v3/reference

[6] https://developers.google.com/analytics/devguides/reporting/core/v3/limits-quotas

[7] https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingCustomVariables

[8] http://wordpress.org/extend/plugins/wp-slimstat/

c) Platform specialization: Both SlimStat and Google Analytics can track the described data, except of the content type. Google Analytics is a general tracking technique for any kind of web pages. Therefore, it does not provide specialized data for blogs. In contrast, SlimStat is specialized for Wordpress blogs and can track the content type of a website in a blog.

d) Data range: Google Analytics does not track specialized data for Wordpress, but the data is more sophisticated than the data range of SlimStat. For example, Google Analytics also tracks the domain of the website where the visitor comes from. If a visitor uses the search engine Bing to find the blog, Google Analytics tracks Bing as referrer. In this case SlimStat would only track the landing page.

e) Portability of the plugin: Until now, the plugin can only be used with Wordpress, but it can be ported to other blog platforms like *Blogger.com*. On other platforms, SlimStat cannot be used any longer. It is applicable for Wordpress only whereas Google Analytics can be used on all platforms.

f) Data quality: In the following scenario, the data of one blog at different times will be not fully comparable. First, SlimStat is installed, so it is used for tracking the visitors of the blog. After a while, the blog administrator uninstalls SlimStat to use Google Analytics for tracking. The visitor statistics of the blog are collected over the whole time, but the tracked data differs between SlimStat and Google Analytics. The reason for it lays in different tracking techniques. For example, Google Analytics uses JavaScript to track the visitors. Consequently, when a visitor with disabled JavaScript visits the blog, he will not be tracked with Google Analytics, but with SlimStat.

g) Conclusion: In conclusion, the preferred tracking technique for the blog administrator and for the blog ranking may differ. The blog administrator may prefer Google Analytics because he installs our blog plugin and it works. There are no further dependencies that has to be installed by the administrator. Nevertheless, some blog administrators take care of data protection and do not want that Google stores their visitor statistics. In contrast to the administrator, the blog ranking profits from blog-specific data. Therefore, SlimStat may fit better. But when the plugin is ported to other blog platforms, Google Analytics is much better. In addition, the use of Google Analytics enables to change the data range that is stored in the BlogIntelligence database. All available statistics are already stored on Google servers and can be polled if needed. For SlimStat, the polled data is fixed and can only be changed by updating the plugin.

V. ARCHITECTURE

This chapter gives an overview of the main components of the infrastructure. The plugin should track the statistic data and display visualizations. Therefore, the plugin needs read and write access to the BlogIntelligence database. This database already contains data about previously crawled blogs. Besides the visitor data, the visualizations use existing data about trends and hyperlink structures from BlogIntelligence.

As shown in Figure 2 the plugin tracks clicks from visitors on the front end of the blog and sends the tracking data to the BlogIntelligence database. In the database the tracked data is stored with a blog-specific key which is called *API key*. Section V-B explains how this API key prevents unauthorized access to the data.

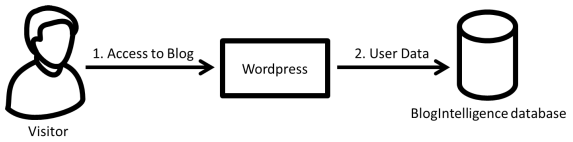

Fig. 2. General principle of the visitor tracking

As described in chapter IV two different tracking tools are used: SlimStat and Google Analytics. If the selected tracking technique is SlimStat, the plugin updates the BlogIntelligence database directly. If Google Analytics is used, the tracking data is stored on servers from Google. This data has to be written in the BlogIntelligence database in an independent process. The following sections will give a more detailed understanding of these processes.

A. Architecture

The infrastructure includes the following components: the BlogIntelligence database, a Wordpress installation with the developed plugin, servers from Google and a server for polling visitors statistics from the Google servers. The communication between these components is shown in Figure 3.

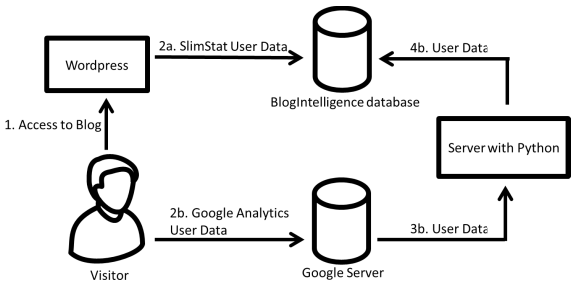

Fig. 3. Communication between the components of the infrastructure

The BlogIntelligence database has a built-in application server which provides a restful API.

For updating the visitor data this application server gets the data over the URL as GET-parameters and writes them into the database. If the plugin uses SlimStat and a visitor clicks on the front end, the plugin counts all visitors in the current hour. Afterwards, the plugin calls the application server API with these new information. In Figure 3 the steps *1* and *2a* describe this process.

For plugins using Google Analytics, updating the visitor data is a bit more complex. The steps 1, 2b, 3b and 4b in Figure 3 show the communication for Google Analytics. The tracking data is stored on servers from Google. In order to make this data available to the BlogIntelligence project it has to be transferred to the projects database. This is done by an additional server. On this server a cron job is installed that periodically runs a python script that transfers the data from Google servers to the BlogIntelligence database.

B. Security

The plugin should be available online. Therefore, it is accessible by everyone and the interface between the plugin and the Application Server, providing a RESTful API, has to be available for the public. Nevertheless, the visitor statistics should not be accessible to everyone. Because of that, there need to be a security constraint to enable data access only for the blog administrators. Additionally, every administrator should only be able to read the data corresponding to his blog and every plugin should only write data for its own blog. Other persons like visitors should neither be able to read the data nor manipulate it.

To ensure these constraints, the plugin uses a pair of unique identifiers, called API keys. There is a *server API key* and a *user API key*. Both keys are calculated when the plugin is installed and stored until the plugin will be uninstalled. Directly after the generation of the keys, they are sent to the BlogIntelligence database. After that, every communication between the plugin and the database is controlled by these keys. If no API key is sent for a request or the transferred API key can not be found in the database, the request is denied. As a result, other persons can read the database interface from the plugin code, but they can not send requests in the name of an existing blog.

If the plugin writes into the database, it sends the tracked data together with its corresponding server API key to the database. On the database every table belonging to this plugin contains a column for the server API key. For requesting visualization data, the plugin also transfers its server API key to the database. Thus the database can determine the visualization data corresponding to this plugin installation.

The user API key is needed for tracking with Google Analytics. Plugin installations that use SlimStat for tracking also have an user API key to enable switching to Google Analytics later. But as long as the plugin uses SlimStat, it does not need the user API key.

Google Analytics allows to store own values which are called custom variables as discussed in section IV-A1. This kind of variable is used for the storage of the user API key. Google Analytics tracks the visitor over JavaScript code that belongs to the plugin and is running in the browser of the visitor. That means every click of a visitor on the front end calls Google Analytics. Each of these calls include the user API key of the plugin.

The JavaScript code that calls Google Analytics can be read by every visitor. Consequently, the visitor can read every data that is sent to Google Analytics. To avoid making the API key known to a visitor the plugin should not send the secret server API key to Google Analytics. Because of that, a user API key is generated. Even if the visitor reads the user API key, he could not use it to access the data on the BlogIntelligence database. For such type of access the server API key is needed.

Thus only the user API key is stored on the Google servers. Whenever the data from the Google servers should be stored in the BlogIntelligence database the server API key will be needed. For this process there is a python script on an additional server that reads the data from the Google servers and writes them into the BlogIntelligence database.

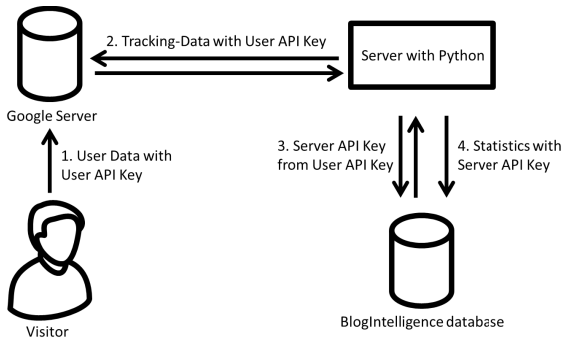

Fig. 4. Communication using API keys

This python script first has to detect the corresponding server API key for the user API key from the Google server. For this reason there is a table on the BlogIntelligence database that maps user API keys to server API keys. To enable the python script to use these mapping, a special database user is required who can access this table. Only the python script should have the credentials for this user and thereby only the python script can determine the corresponding server API key to a given user API key. The communication using API keys is shown in Figure 4.

The python script periodically polls the data for all tracked blogs from the Google servers, determines the corresponding server API keys and then writes the data into the BlogIntelligence database.

C. Database schema

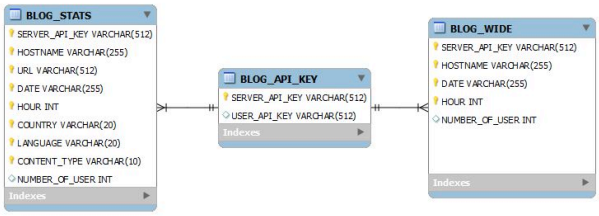

Fig. 5. Database schema for the tracking data

To distinguish between the blog-level and post-level data the visitor statistics are stored in two different tables. The BLOG_STATS table stores post-level data whereas BLOG_WIDE stores blog-level data. In addition, there is a table to store API keys which are described in section V-B. Figure 5 shows the schema of the described tables. The table for the blog-level data stores the number of visitors per blog domain and time period. The chosen time period is one hour. The post-level data describe visitor statistics per post. The visitor statistics on post-level are the number of visitors per URL, time period, country, language and content type. This table is not restricted to posts but can also track other URLs of the blog, for example the front page.

VI. Blogger Benefits

To get access to the internal visitor statistics of a blog the proposed approach has to rely on the blog administrators cooperation. In exchange for their visitor data the proposed plugin provide several visualizations based on the data collected by BlogIntelligence. This enables blog administrators to contribute to BlogIntelligence while profiting not only from their own data, but additional data retrieved by BlogIntelligence. The blog administrator should get a high benefit from the visualizations. Therefore, the proposed visualizations should center around the contributing blog to fit their needs.The plugin contains three visualizations in the categories blogosphere, trend and blog-intern visualization as described below.

A. Blogosphere Visualization

Using a visualization of the blogosphere can generate new insights for blog administrators about how the blogosphere around them is structured. By visualizing its link structure, possible paths a visitor could come to a certain website are illustrated.

Node-link structures are commonly used to visualize the blogosphere like described in Kouper et al. [8]. Frequently, this results in visual clutter for a large amount of blog entities. To address this issue we propose the circular bundle view approach presented by Holten [11]. Using this approach graph nodes, namely clusters, blogs and posts, are projected on the outline of a circle. Relations between these elements are drawn in the middle of the circle. In order to highlight their hierarchical structure, graph nodes are represented in a nested way using an inverted version of a radial tree layout [16].

Relations are bundled according to the hierarchical information available for blog data using the hierarchical edge bundling algorithm proposed by Holten [11]. The visual bundling leads to a reduction of visual clutter as well as marks the fundamental relationships between blog clusters. The result is shown in Figure 6.

h) Data: For the proposed plugin the blogosphere visualization should illustrate content around a certain blog of interest. Therefore, the visualization is limited to the neighboring parts of the blogosphere. The layout is based on data of blogs which have a hyperlink relationship to the blog where the plugin is installed. Due to the self centered approach the given blog of interest is weighted stronger relatively to the other depicted blogs. Additionally, posts of the blog of interest are depicted.

Relations in the inner part of the layout shows the linkage structure of the given part of the blogosphere. So blog administrators can identify from the visualization which blogs they link to as well as which blogs refer to their content.

High-level pattern are revealed by the bundling approach applied on the relations. Using the bundling algorithm points, strong interconnection are highlighted. For example, areas of high bundling refer to frequently linked blogs. To summarize, the bundle view visualization can answer questions of structural relationships to related blogs.

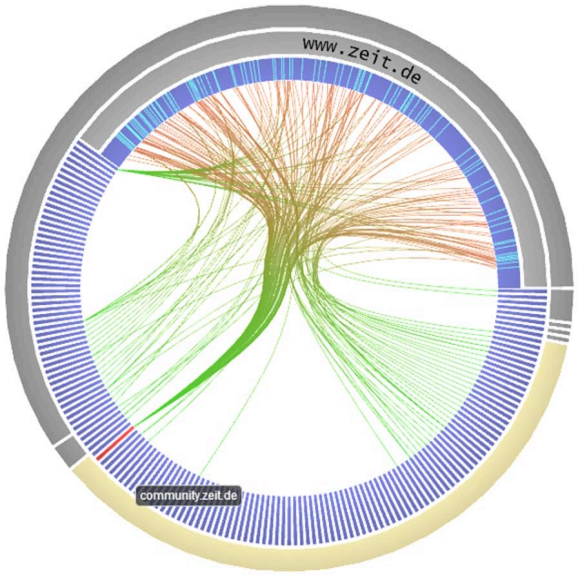

Fig. 6. Visualization of the neighboring blogosphere of www.zeit.de

i) Interaction techniques: Besides the choice of a proper layout the selection of meaningful interaction techniques can drastically enhance the quality of a visualization and the insights generated by it.

Due to a strong bundling the sign of single connections can get fade. To still enable the reckoning of specific node-level connections the visualization supports to highlight certain elements by hovering over them. By hovering over a node its hierarchical dependencies become highlighted. For example, if the blog of interest is selected, its posts as well as its cluster will be highlighted. Additionally, all elements that are linked to the selected node are highlighted to show their structural dependencies. Clicking on a node opens up the chosen website to quickly examine the content behind the displayed data.

While working with the visualization not all links are desired to be displayed for every use case. Therefore, the incoming or outgoing relations can be filtered. The filtered visualization frequently results in clearer insights by displaying only a part of the entire linkage structure.

B. Trend Visualization

Knowledge about which topics readers are currently interested in, enables blog administrators to adjust their writing on the needs of their audience. Writing about topics in current interest could attract more visitors to a blog and lead to a more successful blog.

Illustrate multiple metrics over time clearly is a yet unsolved issue although many research projects focused on this topic. Typical approaches are often space consuming or restricted to display only a single metric. Due to these issues the proposed visualization unlike many approaches in this field does not explicitly show evolutionary information for example in form of a timeline. However, the proposed approach illustrates the current state of a trend while approximating its evolution using a single value per data item.

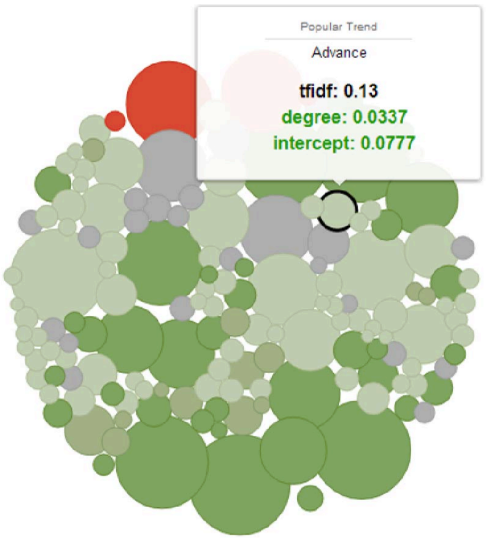

Fig. 7. Trend Visualization using important words of www.heise.de

For the developed plugin the trend visualization should illustrate the trend of blog related topics. Therefore, the diagram depicts words that are used in the blog of interest as circles which is illustrated in Figure 7. The circles are positioned according to a force layout applied on them. Circles are dragged by a gravitational-like force to the middle point of the layout. Additionally, an electrostatic-like force acts between a pair of circles to keep them apart from each other. To deal with a massive amount of words the barnes-hut algorithm [17] is applied to overcome the otherwise quadratic complexity for calculating forces. This algorithm uses a quadtree to approximate forces of groups of words to dramatically decrease complexity [18].

j) Data: Three metrics are applied for each word in this visualization: The tf-idf value, the approximated trend change and the current popularity. The other two metrics are generated by a linear regression algorithm. These metrics represents the degree and intercept with the y-axis of the linear function described by Hennig et al. [19]. Hence, they define an approximation for the development of a trend.

The term frequency-inverse document frequency, in short tf-idf, is a statistical approach that reflects the importance of a word to a collection of documents. The average tf-idf of all posts in a blog is represented by the size of the corresponding circle. More important words are depicted as larger circles and take a greater area in the diagram. The intercept marks the current popularity of a certain topic. This metric is visualized by the color of a circle where green circles indicate highly popular words while red circles indicate words out-of-favor.

The degree, the change of a trend, is mapped to the position of the circle. Therefore, the visualization makes use of a bubble metaphor. Words with a higher increasing trend have a greater buoyancy and are positioned higher in the diagram than words with stable or decreasing trends. This technique results in a layout from words with greatly increasing trends on top to falling trends at the bottom of the chart.

k) Interaction techniques: It is important for blog administrators to get a quick overview which topics of their blogs are in an increasing or decreasing trend. Therefore, words with certain trends can be filtered out. For example, to Figure out which topics are currently not interesting for visitors, only words with an decreasing trend could be displayed.

C. Blog-intern Visualization

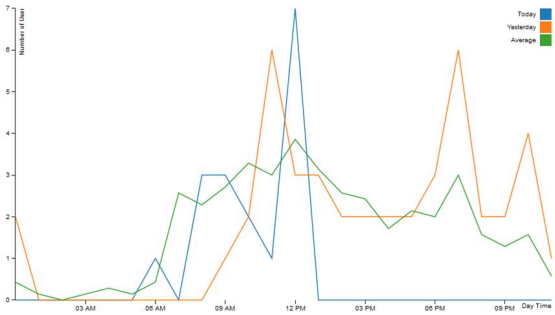

Fig. 8. Visualization of the visitor distribution of www.internetworking-buch.de

Besides the trend of topics in their blogs and their connection on related blogs blog administrators are especially interested in the behavior of their visitors. With these information blog administrators can get insights in how popular a post actually is.

The most common visualization to show visitor activity in this field, is the basic line chart as described by Burton [20]. Line charts have an ease of access while allowing a good overview for a small amount of data. The implementation of the proposed visualization based on chapter 3 of Getting Started with d3 [21]. The resulting diagram is shown in 8. The visualization is extended by the ability to depict multiple lines as well as smooth transitions between different data sets.

l) Data: The diagram shows the distribution of the visitor activity throughout the day. Therefore, the visualization based on the data the proposed plugin collects. The x-axis represents the time of day and the y-axis represents the number of visitors reading the blog in a certain hour. So blog administrators can easily identify peak times of visits throughout the day. Multiple data sets are shown at the same time: The number of visitors today, yesterday and the average of the last seven days. Thus, the visitor activity can be compared between different days of the week.

m)Interaction techniques: To get a more fine-grained insight into blog-level visitor statistics the underlying data can be filtered by specific websites. By selecting an URL the number of visitors are updated. The lines and the y-axis are adjusted to fit the updated content. A comparison between posts is eased via a smooth transition between the two representations.

VII. Future Work

The current analytics capabilities are limited to simple visitor statistics. For advanced ranking algorithms we need to incorporate more data like the social relation of the user to the blogger, the influence of the visiting user etc. Even

identifying a user and track him over time is a special topic of interest. Nevertheless, the current implementation allows that the BlogIntelligence ranking can access the visitor statistics of a blog. Thereby, we expect an improvement of the current BIImpact Ranking

In addition to a wider data range, the integration of more visualizations is necessary to motivate more blogger to install our plugin and thereby contribute to a more accurate ranking of the blogosphere. The motivational effect of these visualizations also has to be evaluate using a qualitative study of bloggers.

Furthermore, we expect that the usage of the plugin motivates bloggers to actual change their blogging behavior according to the visualized surroundings of the blogosphere. This effect also needs to be evaluated by monitoring e.g. the change of topics or publishing frequency.

VIII. Conclusion

The proposed plugin enables the tracking of visitor data for Wordpress blogs. This data is usually not accessible for external blog search engines. To incorporate visitor data into the ranking mechanisms of blog search engines we proposed a blog plugin that offers the user a variety of visualizations to motivate the author to voluntary share his visitor data.

The visualizations give the blog author a overview of frequently discussed topics in his own and adjacent blogs. Further it enables the blogger to adjust its topical focus to currently trending topics and gives an overview of linked and linking blogs of other authors.

The data send by the blog plugin currently consists of simple visitor statistics that can serve as a first indicator for the actual popularity of a blog, which can be integrated in the blog ranking mechanism of blog search engines like BlogIntelligence. For tracking we use different techniques e.g. SlimStat and Google Analytics. Further visitor engagement measurements will follow.

We developed a safe communication between the blog search engine and the blog plugin to secure the user data. Thus, the BlogIntelligence project can use the developed plugin to gain access to visitor statistics to extend and improve a blog ranking as described by Berger [3] and Bross et al. [1].

References

[1] J. Bross, K. Richly, M. Kohnen, and C. Meinel, "Identifying the top-dogs of the blogosphere," *Social Netw. Analys. Mining*, vol. 2, no. 1, pp. 53–67, 2012.

[2] M. A. Tayebi, S. M. Hashemi, and A. Mohades, "B2rank: An algorithm for ranking blogs based on behavioral features," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 104–107. [Online]. Available: http://dx.doi.org/10.1109/WI.2007.28

[3] P. Berger, P.Hennig and C. Meinel, "Ranking blogs based on topic consistency," 2012.

[4] J. Bross, K. Richly, M. Kohnen and C. Meinel, "Identifying the top-dogs of the blogosphere," *Social Netw. Analys. Mining*, vol. 2, no. 1, pp. 53–67, 2012.

[5] J. Carrabis and E.T. Peterson, "Measuring the immeasurable: Visitor engagement," Web Analytics Demystified, White Paper, 2008.

[6] D. Oberle, B. Berendt, A. Hotho and J. Gonzalez, "Conceptual user tracking," in *Advances in Web Intelligence*, ser. Lecture Notes in Computer Science, E. Menasalvas, J. Segovia, and P. Szczepaniak, Eds. Springer Berlin / Heidelberg, 2003, vol. 2663, pp. 955–955. [Online]. Available: {http://dx.doi.org/10.1007/3-540-44831-4_17}

[7] R. Atterer, M. Wnuk and A. Schmidt, "Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction," in *Proceedings of the 15th international conference on World Wide Web*, ser. WWW '06. New York, NY, USA: ACM, 2006, pp. 203–212. [Online]. Available: http://doi.acm.org/10.1145/1135777.1135811

[8] S.C. Herring, I. Kouper, J.C. Paolillo, L.A. Scheidt, M. Tyworth, P. Welsch, E. Wright and N. Yu, "Conversations in the blogosphere: An analysis," in *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, 2005, pp. 107b–107b.

[9] J. Bross, P. Schilf, M. Jenders and C. Meinel, "Visualizing the blogosphere with blogconnect," in *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*. IEEE, 2011, pp. 651–656.

[10] D. Holten and J.J. Van Wijk, "Force-directed edge bundling for graph visualization," in *Computer Graphics Forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 983–990.

[11] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 741–748, 2006.

[12] S. Havre, B. Hetzler and L. Nowell, "Themeriver: Visualizing theme changes over time," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE, 2000, pp. 115–123.

[13] E.R. Tufte, *Beautiful evidence*. Graphics Press Cheshire, CT, 2006, vol. 23.

[14] B. Lee, N.H. Riche, A.K. Karlson and S. Carpendale, "Sparkclouds: Visualizing trends in tag clouds," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1182–1189, 2010.

[15] J. Cutroni, *Google Analytics*. O'Reilly Media, Incorporated, 2010.

[16] W.H. Smith, *Graphic statistics in management*. McGraw-Hill, 1924.

[17] J. Barnes and P. Hut, "A hierarchical 0 (n log n) force-calculation algorithm," *nature*, vol. 324, p. 4, 1986.

[18] R.A. Finkel and J.L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[19] P. Hennig, P.Berger and C. Meinel, "Identify emergent trends based on the blogosphere," 2013.

[20] B.G. Andreas, *Experimental psychology*. New York: Wiley, 1972.

[21] M. Dewar, *Getting started with D3*. Sebastopol, CA: O'Reilly, 2012.