International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:12, 2015

The Visualizer for Real-Time Analysis of Internet Trends

Radek Malinský, Ivan Jelínek

***Abstract*—**The current web has become a modern encyclopedia, where people share their thoughts and ideas on various topics around them. This kind of encyclopedia is very useful for other people who are looking for answers to their questions. However, with the growing popularity of social networking and blogging and ever expanding network services, there has also been a growing diversity of technologies along with a different structure of individual web sites. It is therefore difficult to directly find a relevant answer for a common Internet user. This paper presents a web application for the real-time end-to-end analysis of selected Internet trends where the trend can be whatever the people post online. The application integrates fully configurable tools for data collection and analysis using selected webometric algorithms, and for its chronological visualization to user. It can be assumed that the application facilitates the users to evaluate the quality of various products that are mentioned online.

***Keywords*—**Trend, visualizer, web analysis, web 2.0.

I. Introduction

IN recent years, the Internet has been experiencing a huge boom in social networking, blogging and discussing on online forums, and this is mainly due to the desire of users to share their thoughts and opinions on various topics, products and events around them. This development phase of the web, collectively Web 2.0, has reached to a such stage, where it is quite usual for ordinary users to share their opinions publicly online. Many web services are gradually adapting to this trend, and for instance, online shops allow for consumers to post comments on goods they bought. This facilitates the target customer in making the decision to purchase, the seller gets quick feedback on the goods and the producer determines what to improve on his products. However, it is not easy to analyze such comments when they are on multiple sources.

Web services diversity, a variety of technologies along with a different structure of individual web sites, all of these make the analysis of public opinions very difficult. Hence, it is difficult to get accurate feedback on product issues from multiple data sources for an ordinary user, as well as for a commercial company. It is therefore necessary to design a suitable metric for these various data sources that would reflect the semantic content of single pages in the better way and thereby improve a machine understanding of a text.

Our research is focused on the development of the web application, which brings together metrics for analysis and evaluation of Internet trends. Event, product name, name of the person or any expression, which is mentioned online, all of these can be defined as the Internet trend. Such trend assessment provides a chronological insight into a public opinion on specific search topic; for instance, the opinion on price, quality or other factors of any product, or information about the geographic spread of any article or event.

R. Malinský, I. Jelínek are with the Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo náměstí 13, 121 35 Prague, Czech Republic, (e-mail: malinrad@fel.cvut.cz, jelinek@fel.cvut.cz).

II. Related Work

Web search engines are the easiest way to find specific information in such diversified network for ordinary users. However, the search engines just return a list of web sites relevant to the user's search query and they do not provide any direct answer. The user is therefore forced to browse the individual sites and search for answers through consolidation of useful information from them.

The user's search queries can serve as a treasure for web data mining because they reflect the ideas of many users. Google started digging into this data source and revealed the service Google Trends. Google Trends reports a chronological summary of trend volume based on queries that people entered into the Google search engine. Numerous studies have been done that used Google Trends as a data source to reveal various types of information like a detection of influenza epidemic [1], prediction of economic indicators [2] or a state politics research [3]. Those studies reflect what people are searching for; however, they do not express what people think about what they are looking.

Many complex solutions to resolve public opinion are usually tailored to a specific purpose or data type. There is currently no widely acceptable solution for a trend analysis in such heterogeneous environment the Web 2.0 is.

Based on our research, we have revealed a web-based visualizer, which is an extension of the proposed framework [4] and thereby completes the application with end-to-end approach for a real-time analysis and monitoring of selected Internet trends on various data sources. The integrated trend visualizer allows fully configurable possibilities from collection of data from relevant web sources, through its analysis using selected algorithms, to a chronological view of analyzed data to users.

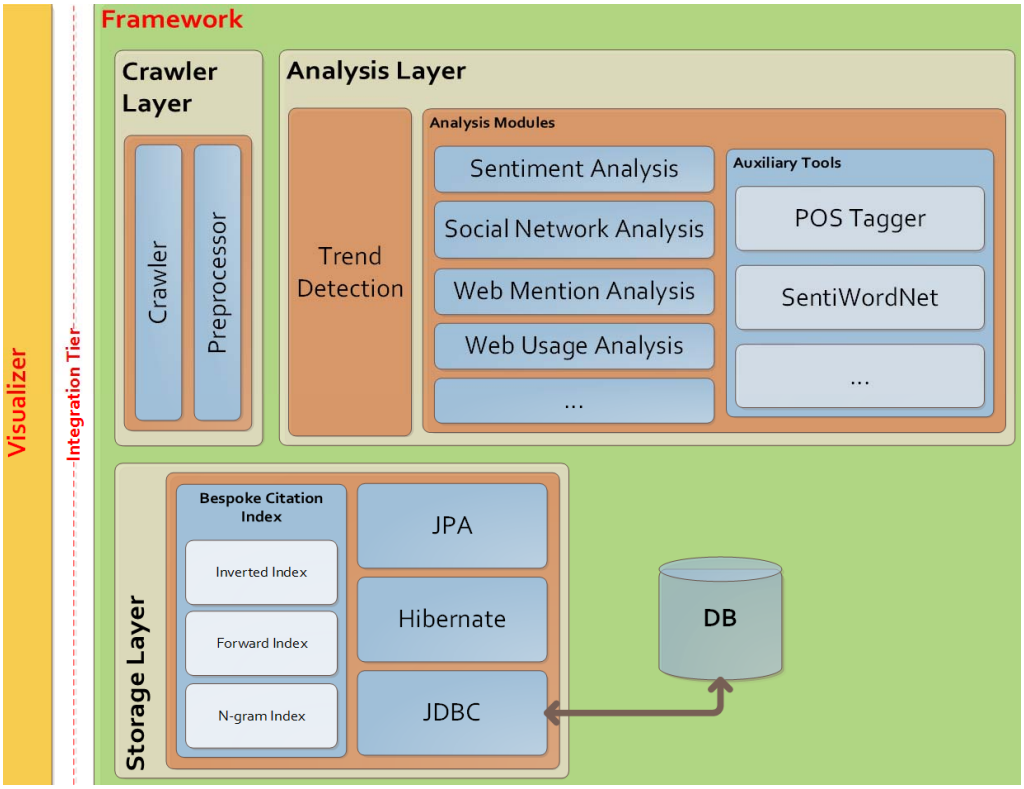

Fig. 1 The overall system architecture that provided end-to-end approach for real-time analysis and monitoring of Internet trends

III. Overall System Architecture

The overall system architecture (Fig. 1) is divided into four main interconnected layers, where each of them has its own functionality and does not affect the others. The framework architecture follows our previous research [4] which describes the individual layers and their functionality in detail. The system is implemented in the Java programming language and it provides an interface for user-created extensions.

A. Crawler Layer

The units at this layer are designed to collect data from the Internet and prepare them for subsequent analysis. The crawler is an automated unit that follows links on the web and stores a key content of all the visited pages. Each page is further analyzed at the html tag level and its content is divided into several categories such as main content, header, footer, metadata, navigation, advertisement, hyperlinks, etc. All the processed content is continually being stored during the html tag analysis in a database.

B. Analysis Layer

The analysis layer is fully configurable from the visualizer. Trend Detection unit is used to find a text that is relevant to the user defined trend. The trend can be even defined as a multi-word expression with a Boolean term to increase the precision. For instance, the trend "Ottawa earthquake" is better to restructure to the expression "ottawa AND earthquake". However, names and specific expressions, e.g. "bill gates", should stay in the same form and then be searched as the exact phrase.

The trend analysis is performed by user interpretation in Visualizer. That can be done by assembling a graph where each node represents an analytical algorithm, and the trend analysis is subsequently performed in the order in which the graph passes through. All the processed results are continuously being stored in a database.

C. Storage Layer

The layer is designed to quickly store and retrieve data from the database in order to be the trend evaluation displayed in real-time to the user. For this purpose, the implemented bespoke citation indexes are defined to optimize speed and performance for communication of the individual layers with database.

The visualizer communicates with a database on a higher level using object-relational mapping mediated by Java Persistence API (JPA) [5]. EclipseLink [5] is the default persistent tool in the proposed architecture; however, it can be anytime replaced by another tool due to the JPA specification.

IV. Trend Visualizer

Visualizer is the main system part which provides graphical user interface for a complete system configuration along with a trend definition, analysis adjustment and visualization of the analyzed results to the user.

The system itself is located at http://malinsky.eu/visualizer. The user visits the website and there he can track the progress and analysis of the all trends in the system, including those that were created by other users. The registration against an email address is necessary to create a new trend and to a configuration of its analysis. The configuration of a trend analysis includes data sources and data collection frequency definition, as well as the choice of a method for processing of collected data and selection of analytical modules.

A. Data Source Definition

The user defines uri and method of web data mining when creating a new data source. The data mining method is determined by the time period that defines the frequency of crawling and by the level of depth of hyperlinks that defines how far the crawler can move away from the original uri while browsing hyperlinks. The entire content of each of the visited pages is stored as it was loaded in a database; i.e. plain text in html format.

B. Web Page Preprocessing

The content of each stored web page is analyzed at the html tag level and then it is divided into several categories: main content, header, footer, etc. Another category can be created and rules for its recognition on a web page using an appropriate html tags or identifiers can be defined by a user.

C. Trend Analysis

Several analytical plans can be defined for each trend in the system. Every plan has defined a data source and categories, where the category describes which part of the source will be used for analysis. It is also possible to define a frequency of analysis; this is especially important if the content of the website itself is being changed. The last important part of the analysis plan is the selection of analytical modules, their configuration and assembling a graph where each node represents one module. Trend analysis is then performed by sequential pass of the assembled graph where the processed data is transferred from one node/module to the other node.

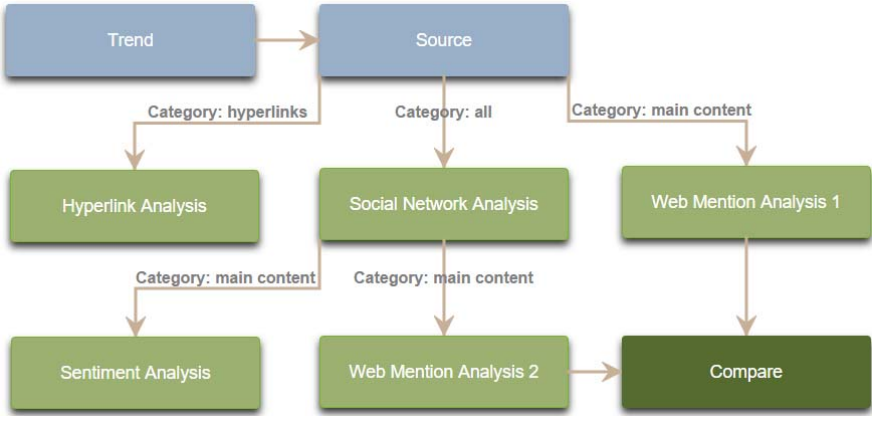

Fig. 2 Trend analysis is performed by passing through the assembled graph

An example of the assembled Visualizer graph is illustrated in Fig. 2. The whole graph is interactive and it can be modified through several mouse clicks. After a double-clicking on a node in the graph, the user displays detailed configuration settings and options for the node. Furthermore, the user is able to add additional nodes and define edges between them.

The blue colored nodes represent a selection. In this case, they determine which trend is analyzed and which web pages from a data source are used for analysis. The output of each node may have a defined category, which constitutes a part of the website that is passed to the analysis. All the outputs of the node are transferred to further analysis if there is no category selected. For instance, the edge between node Source and Hyperlink Analysis is marked by category "hyperlinks"; i.e. only the hyperlinks gathered from analyzed web pages are passed to the node Hyperlink Analysis. The user also has the option to specify which hyperlinks (from navigation, advertisement, main content, all, etc.) are used for the analysis; this option is available in the detailed settings of the node.

The light green colored nodes represent the analytical modules. All the outputs of the analytical modules are automatically displayed to the user in the form of tables and charts. As illustrated, the output of the individual analytical module can be used as input data for analysis in other modules. For example, the Social Network Analysis module is configured to use Degree Centrality to determine the most read web pages and the entire output is then passed to Sentiment Analysis and Web Mention Analysis 2 modules for further analysis.

The dark green colored nodes are used to compare the outputs of two identical analytical modules. The output of the comparison is also displayed to the user in the form of table and chart showing both outputs.

The outputs of the individual modules and also complete results are gradually displayed to the user in a chronological order in tables and charts. The update of the results to the user is almost instantaneous and independent of the user; data is

displayed to the user as soon it is processed or analyzed. This is caused because of the system architecture and the communication strategy with client web browser.

V. Visualizer Architecture

The visualizer architecture (Fig. 3) is based on Java Server Faces (JSF) [6] and supported by Primefaces [7] component library. The components utilize JavaScript and AJAX to provide a rich user experience with support for real-time content updates. The real-time content updates are important for automatic updates of results that are displayed to the user immediately after the data analysis.

The sequence of events resulting in a page update is as:

1) One portion of the data analysis is completed and the results are stored in a database. The trigger, which is part of the business logic is activated and the previously stored data are loaded from a database into the managed bean that manages a displayed web page with results.
2) JSF runtime re-renders the entire component tree stored on the visualizer server-side.
3) The component tree differences are calculated and page update is packed into the XMLHttpRequest object via the AJAX Controller. The XMLHttpRequest object then calls the callback function on a client-side.
4) The XMLHttpRequest callback function updates a web page Document Object Model (DOM) and thereby automatically updates the web page with the new data.
5) The same process is invoked when the user changes any data in the result table and thereby the result chart is automatically updated.

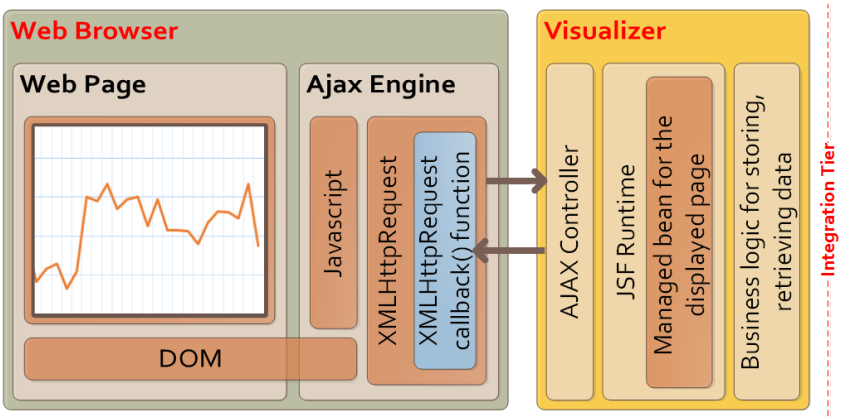

Fig. 3 The visualizer architecture and communication with a client web browser

A. Custom Analytical Modules

The system architecture is vertically scalable which allows the addition of new custom analytical modules. A module deployment section in the visualizer is prepared to install a new analytical module. A template to create a new module is prepared for the user in the same section. The template consists of two files Module.xml and Module.java. The first one defines the layout of a user interface fragment for the module configuration. The second file contains custom functionality and implementation of analytical algorithms. Main public class of the Java file has to implement the IAnalysisModule interface. This interface enforces the user to implement methods that are called during the analysis while passing the graph. The final implementation of the new module may contain multiple Java files with a source code. However, it is necessary to upload all implemented files back to the Visualizer, and then the user may immediately use the new module when assembling the graph.

VI. Conclusion

The complex web application for the end-to-end evaluation of selected Internet trends has been proposed. The application consists of two main parts: framework and visualizer. The framework combines the tools for collection and processing data from the Web, the analytical tools that provide algorithms for analysis of collected data, and the data tools designed to quickly store and retrieve data from a database. The visualizer provides graphical user interface for a complete system configuration along with a trend definition, analysis adjustment, and visualization of analyzed results to the user. The system architecture is vertically scalable which allows the addition of new custom analytical modules.

The new extensions provide more varied possibilities for a trend analysis and they can provide more precise results in a combination with existing algorithms. Based on the continued use of the application the recommendation can be formed about which algorithm to use for a particular task, eventually how to work with any data under a specific domain.

Acknowledgment

This research has taken place under the aegis of the research group Webing (http://webing.felk.cvut.cz) and has been supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS15/087/OHK3/1T/13.

REFERENCES

[1] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski and L. Brilliant, "Detecting influenza epidemics using search engine query data". Nature, vol. 457(7232), pp. 1012-1014, 2009. doi: 10.1038/nature07634

[2] H. Choi and H. Varian, "Predicting the Present with Google Trends". Economic Record, vol. 88, no. 2–9, 2012. doi: 10.1111/j.1475-4932.2012.00809.x

[3] S. Reilly, S. Richey, J. B. Taylor, "Using Google Search Data for State Politics Research". State Politics & Policy Quarterly, vol. 12(2), pp. 146-159, 2012. doi: 10.1177/1532440012438889

[4] R. Malinský and I. Jelínek, "Trend Analysis Framework". Proceedings of the 14th International Conference WWW/INTERNET. IADIS Press, vol. 14, pp. 161-166, 2015. ISBN: 978-989-8533-44-9.

[5] M. Keith and M. Schincariol, "Pro JPA 2". Apress Media LLC, New York, USA, 2013. ISBN: 978-1-4302-1956-9.

[6] Z. Wadia, H. Saleh, A. L. Christensen, "Pro JSF and HTML5: Building Rich Internet Components". Apress Media LLC, New York, USA, 2013. ISBN: 978-1-4302-5010-4.

[7] PrimeTek Informatics, "PrimeFaces". 2015 (online). Available: http://primefaces.org. (30 November 2015).