

# Topic-Based Audience Metrics for Internet Marketing by Combining Ontologies and Output Page Mining

Jean-Pierre Norguet\*, Esteban Zimányi  
Department of Computer & Network Engineering, CP 165/15,  
Université Libre de Bruxelles,  
50 av. F.D. Roosevelt, 1050 Brussels, Belgium  
email: {jnorguet,ezimanyi}@ulb.ac.be

## Abstract

*In Internet marketing, Web audience analysis is essential to understanding the visitors' needs. However, the existing analysis tools fail to deliver summarized and conceptual metrics needed by organization managers and Web site editors. The reason is that HTTP transaction metadata mined by these tools do not include the text content sent to the browsers. In this paper, we first describe the various methods that we conceived to mine the Web pages output by Web servers. These methods include content journaling, script parsing, server monitoring, network monitoring, and client-side mining. Then, for a given ontology, we count the occurrences of ontology entries in the mined pages, and we compare the results to the term weights in the online pages. By aggregating the metrics in the ontology, we obtain audience metrics which should represent the Web site topics. Finally, we validate our approach with experiments on real data using SQL Server OLAP and our prototype WASA.*

## 1 Motivations and Related Work

The ease and speed with which information transactions can be carried out over the Web has been a key driving force in the rapid growth of electronic commerce and Internet marketing. In this context, improving Web communication is essential to satisfy the objectives of the Web site and of its target audience. Dedicated to this end, Web usage mining [14], a relatively new research area, has gained more attention. The strategic goals of Web usage mining are prediction of the user's behaviour within the site, comparison between expected and actual Web site usage, and adjustment of the Web site with respect to the users' interests. Web analytics [16] is the part of Web usage mining that

has most emerged in the corporate world. Web analytics focuses on improving Web communication by mining and analyzing Web usage data to discover interesting metrics and usage patterns. However, Web usage mining and Web analytics suffer from significant drawbacks when it comes to providing summarized and conceptual audience results.

Web analytics tools mine huge amounts of Web usage data and produce many reports. Most of these reports are very detailed and target Web designers and Web developers [19]. Summary reports like the number of visitors and the number of page views target the organization manager, but these results are poor. Finally, page-groups hits give the Web site chief editor conceptual results, but these are limited by page-granularity, page-temporality and page-volatility problems, and are therefore of little interest in most cases. All these limitations prevent Web analytics tools from being used at higher organization levels, where summarized and conceptual results are needed to take decisions with a more significant impact [10].

Research efforts in Web usage mining have mostly left Web analytics aside and have focused on more fertile research paths like usage pattern analysis, personalization, system improvement, site structure modification, marketing business intelligence, and usage characterization [14]. A potential contribution related to Web analytics was attempted with reverse clustering analysis [13], a technique based on self-organizing feature maps. This technique integrates Web usage mining and Web content mining to prioritize the Web site pages according to an original popularity score. However, the algorithm is not scalable, and does not answer the page-granularity, page-temporality, or page-volatility problems. These problems are better considered in the IUNIS algorithm of the Information Scent model [2]. This algorithm produces a list of term vectors representing the users' needs. However, the results are user-centric, suffer from polysemy and synonymy, and the algorithm scalability is unclear. Finally, other research projects related to

---

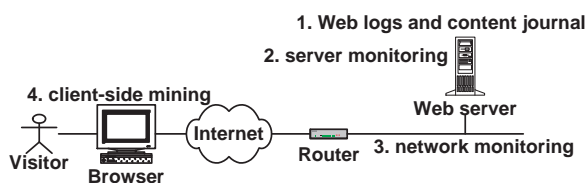
\*Jean-Pierre Norguet's work has been funded by a FNRS Research Fellow Grant.

Web usage mining are described in a recent survey [3], but to the best of our knowledge, none of these projects analyzes the content of the output Web pages to provide summarized and conceptual site-wide audience metrics.

To answer the need of such metrics, our approach aims at analyzing the Web content output by Web servers. Indeed, so far, little or no interest has been shown in the content of the output pages. This disinterest is explained by the lack of techniques to mine the output Web pages and by the high number of pages to analyze afterwards [14]. In Section 2, we present the methods that we conceived to mine the output pages: content journaling, script parsing, server monitoring, network monitoring, and client-side mining. These methods should allow to mine the output pages of any Web site. In Section 3, we explain how term occurrences in these pages can be counted and aggregated with respect to an ontology hierarchy in order to obtain audience metrics for Web site topics. In Section 4, we present and discuss the results obtained with our prototype WASA and SQL Server OLAP. In Section 5, we expose the limitations of the metrics and our future work. Finally, we describe the results exploitation process and conclude in Section 6.

## 2 Output Page Mining

The first step in our approach is to mine the Web pages that are output by the Web server. To this end, we have conceived a number of mining methods, each of them being located at some point in the Web environment (Figure 1).



**Figure 1. Mining points in Web environment.**

The Web environment is centered around the Web server hosting the Web site. The Web server is connected to the Internet Service Provider network, which is connected to the Internet via a router. At the other extremity of the Internet, visitors connect the Web site using their browser. Figure 1 shows a number of points in this Web environment where it is possible to mine the output Web pages: (1) the Web server file system, (2) the Web server running instance, (3) the network wire, and (4) the client-side machine. We call the corresponding mining methods (1) Web logs and content journaling, (2) server monitoring, (3) network monitoring, and (4) client-side mining. These mining points are similar to the communication meta-data mining points used in Web analytics tools. The main difference is the com-

plexity of the mining methods; accessing the page content requires a bigger effort than regarding the meta-data of the communication. In the next sections, we study each of the mining points, describing the corresponding mining method and discussing the pros and cons.

### 2.1 Log Files and Content Journaling

The Web server file system is the most used and simplest mining point in Web analytics tools. Web server log files contain the references of each output Web page, so it is possible to retrieve the page content by looking up the associated file in the Web server document directory. However, if the page content evolves over time, the page version at analysis time can be different of the page version at consultation time.

Most log formats store the request time along with the page reference. So, it is possible to retrieve at analysis time the content of a page as it has been output at consultation time, from the request time, from the page reference, and from a journal that stores the temporal evolution of the pages. To keep track of temporal evolution of the pages, we schedule a daily batch that maintains a *content journal*. Practically, the content journal is a list of entries that are made of a URI, a time period, and a reference to the archived file. This allows to retrieve at any time the exact content of a viewed page, even if the content of the online page has changed over time.

This method requires to mine the least amount of data and subsequently requires the least amount of computation to obtain the various metrics presented in Section 3. As dynamic pages are unique and volatile, content journaling works for static Web pages only.

### 2.2 Script Parsing

As seen in the previous section, a content journal can only be produced for static pages. In many Web servers, dynamic pages are generated from scripted pages, which usually hardcode a part of the text content while the rest is retrieved from a database. We could therefore write a compiler that takes the scripted pages as input and removes the script instructions to produce a pure-HTML page with the hardcoded content.

For experimentation, we implemented [11] a script-parsing compiler for Java Server Pages. The result has proven satisfying as long as the scripted pages hardcode most of the content. If more content is externalized, the extracted content is poor. Also, the compiler is dependant on the page scripting language and on the scripting language version. In the conclusion of our study, script parsing was abandoned in favor of server monitoring (Section 2.3).

## 2.3 Server Monitoring

Server monitoring mines the outgoing pages within the Web server instance. Major Web servers offer an API to interact with the Web server kernel. This allows to insert a custom plugin that saves the output pages onto the file system or into a local or remote database. Practically, a server-monitoring plugin registers with the Web server kernel and gets the control on the output data after it has been sent to the browser. With this method, any kind of file can be traced, including dynamic pages.

Plugins are executed within the Web server. This introduces crash risk into the Web server. Such a risk may be an obstacle to its adoption in critical Web servers. In addition, server monitor plugins are dependant on the Web server API. Different Web servers therefore require different ports of the plugin. However, the porting efforts are reduced by the fact that two products dominate the Web server market: Apache HTTPD and Microsoft IIS.

On the other hand, Web server plugins can access HTTP headers, which include the request file extension and the MIME type of output files. This allows to store dynamic Web pages only, ignoring binary files and static Web pages. Another advantage is that server monitors can transform the response before it is sent to the browser. For example, combination of output page tracing with data compression has proven to save bandwidth and reduce response time [11].

We tested server monitoring by developing *mod\_trace\_output*, a server-monitoring plugin for the Apache Web server.<sup>1</sup> The plugin traced output Web pages with success and passed robustness, performance, and scalability tests. More details about the plugin architecture, implementation, and benchmarks can be found in [11].

## 2.4 Network Monitoring

A network monitor runs in a network-promiscuity mode on the same Ethernet network as the Web server and captures the TCP/IP packets on the network wire. To reassemble Web pages from the TCP/IP packets, a network monitor realizes the following action steps: (1) store TCP/IP packets, (2) filter and group the packets of each HTTP transaction, (3) sort and concatenate the packets to rebuild the transaction, (4) get the metadata from the HTTP header, and (5) remove the header from the HTTP response header to get the Web page.

A network monitor introduces no risk in the Web server. In addition, it is independent from the Web server brand or version. On the other hand, network monitoring is CPU-intensive because it sorts and concatenates character strings. In addition, all files are reassembled before the file type can

<sup>1</sup>*mod\_trace\_output* is available as a SourceForge project: <http://trace-output.sourceforge.net/>

be known, therefore CPU time is spent to capture irrelevant files like images. Network monitoring works for those networks that send the packets on the wired line, like Ethernet networks, and the network monitor must be on the same subnet as the Web server. Finally, network monitoring cannot read encrypted conversations from secure Web servers.

## 2.5 Client-Side Mining

In client-side mining, a program is embedded in the output Web page and runs inside the visitors' browser. When the page is loaded, the program runs inside the browser; it parses the page and sends the page content to a dedicated mining server, which stores the pages content (Figure 2).

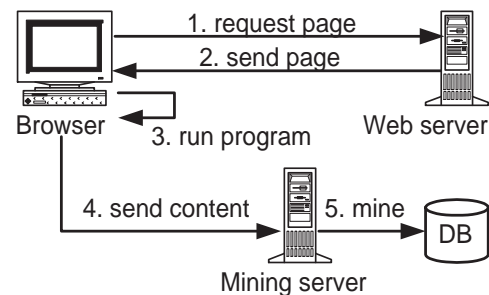


Figure 2. Client-side mining.

As the workload is distributed among the visitors' machines, this mining method can support high-traffic Web sites. To benefit of additional workload distribution, the embedded program can implement content processing (see Section 3). Client-side mining must be used when the publishing technologies involve page layout transformation in the browser, like client-side XML/XSL pages.

Visibility of client-side mining can be a problem. Indeed, visitors can feel unhappy to see that a program is running on their computer, is monitoring what they read, and is sending information to an unknown server. Another drawback of the method is the lack of control on the client side: evil visitors can tweak the program locally and send fake data to the mining server.

## 2.6 Summary

Most, if not all of the Web sites, can be handled by the above mining methods. Log file parsing combined with content journaling is a method that is easy to setup, runs in batch, and offers good performance. For dynamic Web sites, and when script parsing is not satisfying, the alternatives are server monitoring, network monitoring, and client-side mining. Server monitors are usually installed in secure Web sites, and network monitors elsewhere because of the lower risk. Client-side XML/XSL Web pages must

be mined from the client browser. The pros and cons described in each of the previous sections should help choose a method or combination of methods for any Web site, whatever the Web-server or content-publishing technologies.

### 3 Topic-Based Audience Metrics

Ontologies are basically lists of terms and the relations between them, designed to model a domain knowledge [4]. For the given Web site to analyze, we choose an ontology that matches the knowledge domain, that is the topics of the Web site. For each term in the ontology, we count the term occurrences in the output pages. Term counting in Web pages requires content processing operations like page unformatting and text tokenization [1, 5]. We also use stemming [6] to match the ontology terms and their grammatical variations in the pages. This counting gives a term frequency vector representing the term *consultation* over the mining period. If the Web site pages are static, the number of term occurrences in the online pages gives another term frequency vector representing the term *presence* in the Web site. Term consultation and term presence are two interesting metrics but are little representative because the list of terms is too long and because term interpretation suffers from polysemy and synonymy.

In most ontologies, the terms are hierarchically linked by a subclassing relation like *is a* or *part of* [17]. In these ontologies, the audience of the subterms contribute to the communication of the topics denoted by the superterms. Therefore, the audience metrics aggregation from the hierarchy leaves up to the root gives an indication of the audience obtained by the topics denoted by the terms in the hierarchy. Furthermore, the consultation-to-presence ratio gives an indication of the visitors' *interest* into the topics. If the top terms in the hierarchy represent the Web site main topics, the corresponding consultation, presence and interest metrics can be used as topic-based audience measures.

For example, an e-commerce Web site selling food products might use an ontology where the food topic is divided into vegetable and fruit, which are in turn divided into potato and carrot, and into apple and strawberry (Figure 3). The consultation and presence metrics for every terms are represented under the ontology nodes. Metrics aggregation from the leaves up to the root provides topic-based metrics. For example, the aggregated consultation for the *fruit* topic is given by the addition of the consultation for the terms {fruit, apple, strawberry}. The same is done for every topic, as well as for the presence metrics. The interest values are obtained by dividing the consultation and presence values for each topic.

Hierarchical aggregation of the term-based metrics into topic-based metrics can be easily computed and visualized using OLAP tools and a multidimensional model (OLAP

cube) [7]. In such a model, the ontology dimension should be designed as a *parent-child dimension* to support ontologies with any number of levels in each branch [9]. The time dimension can be designed with several levels like year, months, quarters, etc, and should have two important levels: week and day. Indeed, aggregation by week neutralizes the insignificant information contained in the weekly patterns [12]. Other dimensions can be added like physical geography, site geography, Web geography, pages, users, internal referrers, external referrers, and other variations of the time dimension [18]. The cube fact table contains daily term metrics, which are computed by content processing and term counting in the mined pages. The cube measures are consultation, presence, and interest, where the interest measure is a calculated member defined as the division of the first two measures. Feeding an OLAP tool with the cube and with daily term-based metrics allows to compute and visualize the topic-based audience metrics.

### 4 Experimentation

To test our approach, we developed a prototype called WASA.<sup>2</sup> In our case study, we analyzed our department's Web site cs.ulb.ac.be, which contains about 2,000 Web pages and receives an average of 100 page requests a day. The ontology was the ACM classification, which contains 1230 terms hierarchically linked by a part-of relation. The daily term metrics for the academic year 2003-2004 have been computed by our prototype WASA, using Web logs and content journaling. To test the OLAP computation, we introduced the metrics and the ontology into SQL Server. After processing of the cube, we formulated queries on various combinations of dimensions and measures, and we exported the results into Microsoft Excel for visualization.

We first produced a multi-line chart where each curve represents the visitors' consultation of the top ACM concepts (Figure 4). This chart can be intuitively related to the problem domain. For example, Computing Methodologies, Software, and Information Systems rank in the top, while many students follow these courses. Also, a peak of interest in Theory of Computation can be observed at the beginning of the academic year, when the 1st-year students start following the corresponding course in the computers room. Finally, the average consultation falls down during August-September, January-February, and May-June, which are the various exam periods for the students.

We also produced a bar chart representing the metrics for each of the top concepts (Figure 5). In the chart, the top three consulted concepts are Information Systems, Computing Methodologies, and Software. However, these concepts are major topics in the Web site, which is confirmed by high

<sup>2</sup>WASA stands for Web Audience Semantic Analysis

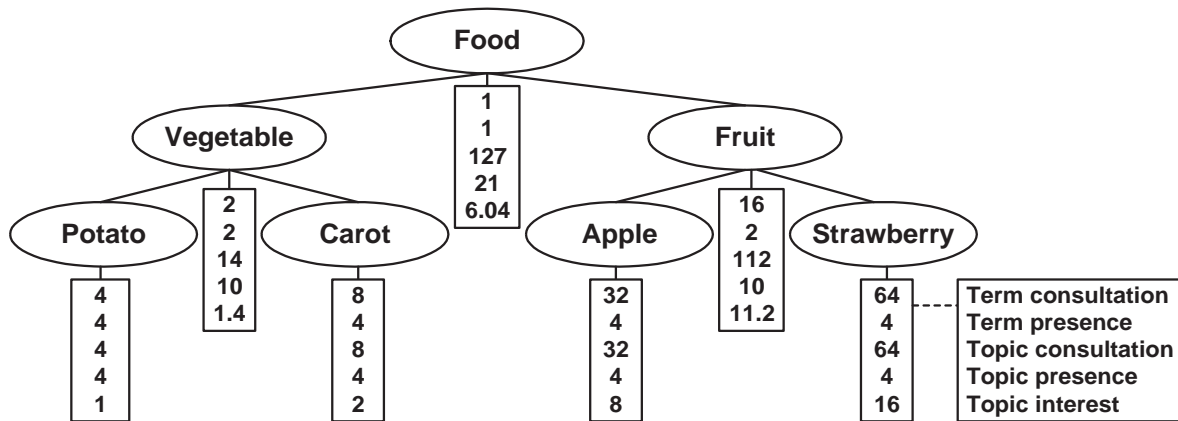


Figure 3. Hierarchical aggregation.

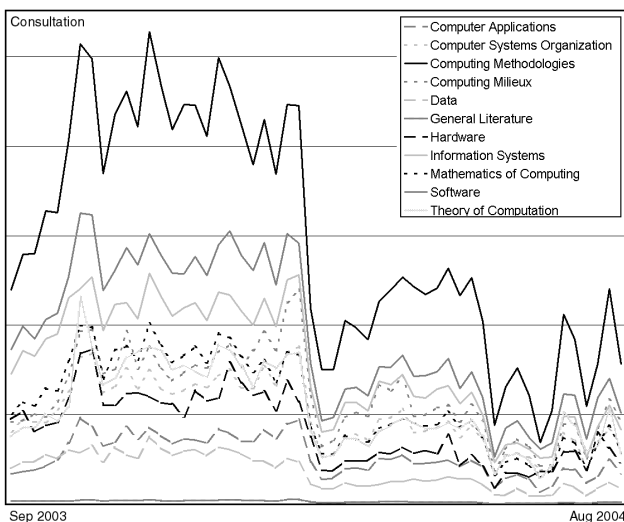


Figure 4. Consultation of the ACM classification top concepts on the cs.ulb.ac.be Web site during the 2003-2004 academic year.

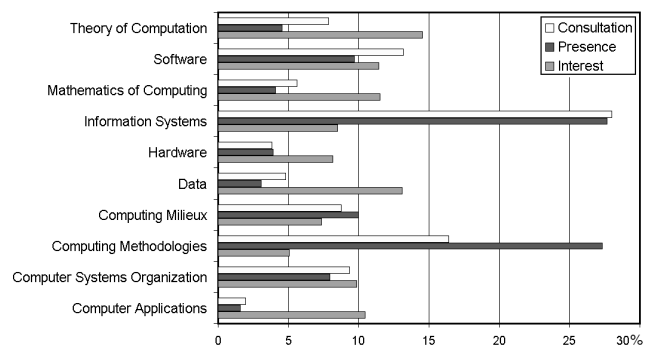


Figure 5. Audience metrics for the ACM classification top concepts.

presence values. Therefore, high consultation values are not representative of the visitors' interest, which is indicated by low interest values. The top three interesting concepts are Theory of Computation, Data, and Mathematics of Computing. We can see that the considered metrics can dramatically change the ranking of the concepts and should be interpreted carefully.

To test the influence of the ontology on topic-based metrics, we made the same experiments with Eurovoc, the European Commission's ontology. Eurovoc contains 6650 terms, and its domain knowledge include all the European Commission's fields of interest. Eurovoc knowledge domain is extremely broad, from sociology to science, while

the ACM classification knowledge domain was focused on computer science. Although Eurovoc contains about five times more terms than the ACM classification, it offers a poor coverage of the computer science domain. Therefore the results obtained with Eurovoc are difficult to relate to the Web site interests. This kind of problem is typical of very conceptual ontologies like Eurovoc [15]. This shows how the choice of the ontology is important for the results interpretation.

## 5 Future Work

As a natural continuation of the Eurovoc experiment, our future work will aim to study the benefits of improving ontology coverage with respect to the Web site knowledge domain. First, we will evaluate a manual approach where the Web site editors will enrich the ACM classification with Web site terms. This method will ensure an optimal improvement of the ontology coverage, the effect of which will be evaluated by running WASA on the enriched ontol-

ogy. Then, the manual enrichment will be compared against automatic and semi-automatic techniques [8].

The results obtained by our approach will be validated against WebTrends, a popular Web analytics tool. Although the results obtained are very different, there is a particular case of Web site where the results obtained by WebTrends should be comparable to those obtained by WASA. Indeed, if the Web site directories match the ontology concepts, the hits by directories obtained by WebTrends should be comparable to the interest by concept obtained by WASA.

Finally, although the complexity of our algorithms are linear, we will test the scalability of our prototype WASA on our university's Web site,<sup>3</sup> which contains a very high number of pages (about 50,000) and receives a very high number of page requests (about 200,000 a day).

## 6 Conclusion

In this paper we have presented our solution to answer the need for summarized and conceptual audience metrics in Web analytics. We first described several methods for mining the Web pages output by Web servers. These methods include content journaling, script parsing, server monitoring, network monitoring, and client-side mining. We showed that these techniques can be used alone or in combination to mine the Web pages output by any Web server. Then, we have seen that aggregating the occurrences of ontology terms in these pages can provide audience metrics for the Web site topics.

According to the first experiments with our WASA prototype and SQL Server, topic-based metrics prove extremely summarized and much more intuitive than page-based metrics. As a consequence, topic-based metrics can be exploited at higher levels in the organization. For example, organization managers can redefine the organization strategy according to the visitors' interests. Topic-based metrics also give an intuitive view of the messages delivered through the Web site and allow to adapt the Web site communication to the organization objectives. The Web site chief editor on his part can interpret the metrics to redefine the publishing orders and redefine the sub-editors' writing tasks. As decisions at higher levels in the organization should be more effective, topic-based metrics should significantly contribute to Web analytics and Internet marketing.

A condition to the wide adoption of topic-based metrics in Web analytics tools is the generalization of custom ontologies for the particular Web sites. Hopefully, the availability of large generic ontologies and the development of ontology enrichment techniques and tools, as well as the growing interest into the Semantic Web, should fill the gap with the continuous development of suitable ontologies.

<sup>3</sup><http://www.ulb.ac.be/>

## References

- [1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web, From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.
- [2] E. H. Chi, P. Pirolli, K. Chen, and J. E. Pitkow. Using information scent to model user information needs and actions and the web. In *Proc. of the SIGCHI on Human Factors in Computing Systems*, pages 490–497, 2001.
- [3] F. M. Facca and P. L. Lanzi. Mining interesting knowledge from weblogs: a survey. *Data Knowl. Eng.*, 53(3):225–241, 2005.
- [4] D. Fensel. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, 2000.
- [5] C. Fox. *Lexical analysis and stoplists*, pages 102–130. Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
- [6] W. B. Frakes. Stemming algorithms. In *Information Retrieval: Data Structures & Algorithms*, pages 131–160. 1992.
- [7] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, 2002.
- [8] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.
- [9] E. Malinowski and E. Zimányi. OLAP hierarchies: A conceptual perspective. In *Proc. of the 16<sup>th</sup> Int. Conf. on Advanced Information Systems Engineering, CAiSE'04*, LNCS 3084, pages 477–491. Springer-Verlag, 2004.
- [10] J. March, H. Simon, and H. Guetzkow. *Organizations*. Cambridge Mass. Blackwell, second edition, 1983.
- [11] G. Materna. Extraction par déformage du contenu de pages Web dynamiques semi-structurées. Travail de fin d'études d'Ingénieur civil informaticien, Faculté des Sciences Appliquées, Université Libre de Bruxelles, 2002.
- [12] J.-P. Norguet. Mise en ligne d'informations statistiques relatives aux accès à des serveurs Web. Travail de fin d'études d'Ingénieur civil informaticien, Faculté des Sciences Appliquées, Université Libre de Bruxelles, 1998.
- [13] S. A. Ríos, J. D. Velásquez, E. S. Vera, H. Yasuda, and T. Aoki. Using SOFM to improve web site text content. In *Proc. of First Int. Conf. on Advances in Natural Computation, ICNC 2005, Part II*, pages 622–626, 2005.
- [14] J. Srivastava, R. Cooley, M. Deshpande, and T. Pang-Ning. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD*, 1(2), 2000.
- [15] R. Steinberger, B. Pouliquen, and C. Ignat. Exploiting multilingual nomenclatures and language-independent text features as an interlingua for cross-lingual text analysis applications. In *Proc. of the 4th Slovenian Language Technology Conf., Information Society 2004*, 2004.
- [16] J. Sterne. *Web Metrics: Proven Methods for Measuring Web Site Success*. John Wiley & Sons, 2002.
- [17] G. Stumme and A. Maedche. Fca-merge: Bottom-up merging of ontologies. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence, IJCAI 2001*, pages 225–234, 2001.
- [18] M. Sweiger, M. Madsen, J. Langston, and H. Lombard. *Clickstream Data Warehousing*. John Wiley & Sons, 2002.
- [19] U. Wahli, J. Norguet, J. Andersen, N. Hargrove, and M. Meser. *Websphere Version 5 Application Development Handbook*. IBM Press, 2003.