A Framework for Automated Web Business Intelligence Systems

Daniel S. Soper
Department of Information Systems
W.P. Carey School of Business
Arizona State University
Daniel.Soper@asu.edu

Abstract

This paper proposes a contemporary architecture to guide the development of Automated Web Business Intelligence (AWBI) systems. AWBI systems are outcome-oriented software applications that utilize automated processes in order to extract actionable organizational knowledge by leveraging the content of the web. The goal of an organization in implementing an AWBI system is to gain competitive advantage by utilizing information garnered from web sources to inform corporate decision making. Although more and more organizations are using AWBI systems to gain competitive advantage, the fact remains that most companies have not yet introduced an AWBI initiative as part of their overall decision support strategy. To that end, a solid framework by which organizations can implement AWBI systems is both timely and desirable. This paper presents such a framework, and in so doing outlines a feasible approach by which organizations can adopt AWBI systems in support of their strategic decision making processes.

1. Introduction

The benefits of data mining to organizational decision making have been well known for quite some time. As the volume of information available online continues to grow at a logarithmic pace, many organizations are beginning to look to the web for supplementary sources of mineable business intelligence data that can be used to inform their decision making processes. Although the web is undeniably the most comprehensive and voluminous repository of information ever created, it is also fantastically convoluted and amorphous. For those brave souls who endeavor to mine this unforgiving landscape, these two disparate attributes make the web simultaneously both a blessing and a curse: On the one hand, the promise of information lurking somewhere in cyberspace that can be cultivated into competitive advantage is simply too tempting a treat to pass up. On the other hand, the technical difficulties that must be overcome in order to automatically locate and extract contextually relevant actionable knowledge from the web are nearly overwhelming. Despite these inherent difficulties, more and more organizations are relying on web-based data for decision making. The process by which those data are extracted from the web and made actionable in an organizational setting is known as web business intelligence or WBI [11].

This paper examines the issues involved in automating the web business intelligence process and proposes an architecture to guide the development of Automated Web Business Intelligence (AWBI) systems. AWBI systems are outcome-oriented software applications that utilize automated processes in order to extract actionable organizational knowledge by leveraging the content of the web. The goal of an organization in implementing an AWBI system is to gain competitive advantage by utilizing information garnered from web sources to inform corporate decision making. Although AWBI systems will never replace proprietary organizational data repositories as the primary source of information for corporate decision support systems, they certainly have the potential to become the principal source of external organizational decision making information. In addition to the AWBI architecture itself, this paper also presents and discusses a prototype AWBI system built by the author using the principles forwarded by the framework proposed herein. This prototype system clearly demonstrates the feasibility and efficacy of the automated web business intelligence architecture. The role of AWBI systems in the overall corporate decision support strategy is depicted graphically in Figure 1 on the following page.

Automated Web Business Intelligence systems are poised to revolutionize the organizational data gathering process. Although more and more organizations are deploying AWBI systems in an effort to gain competitive advantage, the fact remains that most companies have not yet introduced an AWBI initiative as part of their overall decision support strategy [12]. To that end, a solid framework by which organizations can implement AWBI systems is both timely and desirable. This paper presents such a framework, and in so doing outlines an approach by which organizations can adopt Automated Web Business Intelligence in support of their strategic decision- making processes.

0-7695-2268-8/05/$20.00 (C) 2005 IEEE

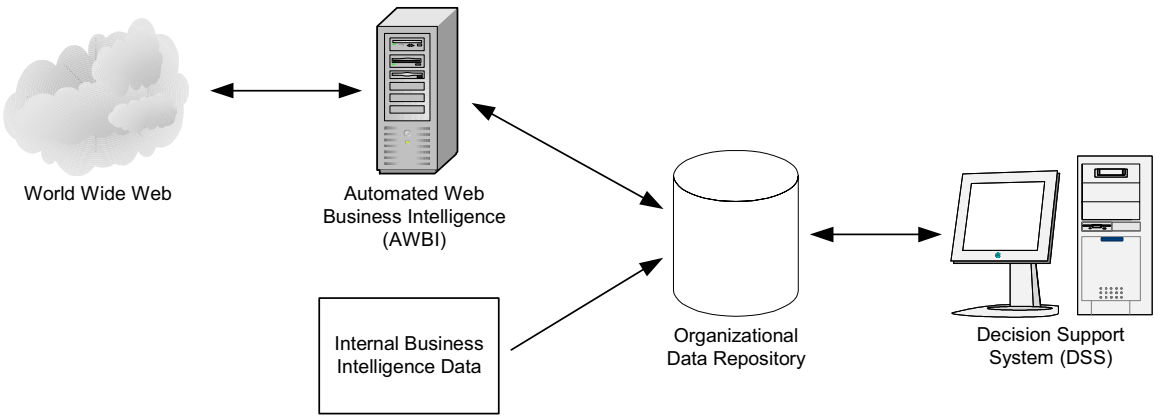

Figure 1. The role of AWBI in corporate decision support.

The remainder of this paper is organized as follows: Section 2 provides a review of related research literature and develops the foundations on which the AWBI framework is built. Section 3 presents the AWBI framework and discusses several issues involved with the implementation of an AWBI system. Section 4 presents and discusses a prototype system built on the AWBI framework. Section 5 concludes the paper by providing a brief summary and describing directions for future research.

2. Related Literature

Traditionally, the application of data mining techniques to the web has been referred to as web mining [2]. As noted by Cooley et al. [2], it is important to differentiate web content mining from web usage mining, as the term *web mining* has been generically applied to both processes. Web usage mining refers to the process of searching for user behavior patterns by mining the data stored in referrer logs, server access logs, and other web user behavior data repositories. Conversely, web content mining refers to the process of searching for contextually relevant sources of web data whose embedded information can be extracted and used to generate actionable knowledge. AWBI systems rely heavily on the web content mining process.

Unfortunately, mining content information from the web is not nearly as straightforward as traditional data mining, as the majority of the information on the web is primarily of a semi-structured nature [3]. Given this difficulty, several approaches have been taken to automatically extract embedded data from semi-structured web pages and convert it into a structured form suitable for further analysis. The most straightforward and prevalent approach to extracting data from a semi-structured web page is through the use of "wrappers". Wrappers allow for automated data extraction by identifying a specific set of HTML or XML tags that surround the target data. Although this approach is simple to implement, it is also very time-consuming and offers low scalability [4].

Conversely, several researchers have explored machine learning algorithms and pattern discovery as approaches to wrapper induction, automated data extraction, and information categorization from semi-structured web pages [e.g. 1, 4, 5, 8, 9, 14]. These systems are designed to use intelligent processes and pattern matching techniques to allow for the automatic extraction and categorization of potentially useful information from a semi-structured web source. Generally speaking, these approaches are more scalable than the manual wrapper construction approach, but they are also more difficult to implement. Furthermore, the systems themselves require an extensive degree of training before acceptable degrees of information extraction accuracy can be achieved [4].

Another noteworthy approach to web information location and extraction are the many web query languages. From an end-user's perspective, these languages function in much the same way as SQL; *viz.* the user identifies what kind of information she wants to retrieve, the source of that information, and some variety of filtering condition. The web query language will then execute her request against the web in much the same way that a SQL statement is executed against a structured relational data repository. Some of the more prominent web query languages are WebQL [12], which allows organizations to query the Internet for information about their competition, WebSQL [7], which allows for the querying of hyperlink paths, W3QL [6] which allows users to query for hyperlink structure and web page content, and Squeal [10], which allows for the querying of

the web by employing a just-in-time (JIT) database approach.

Srivastava and Cooley [11] developed a general architecture for web business intelligence systems that can be applied by individuals or organizations. Their work led to the identification of the two major components of web business intelligence systems that guide this research, namely (1) content acquisition, and (2) knowledge creation. The architecture proposed herein differs markedly from that forwarded by Srivastava and Cooley, however, as is described in the following section.

3. The AWBI Architecture

The AWBI architecture described in this section builds on previous research in this area, but differs from and extends it in several important ways. First, the framework described in this paper focuses specifically on *automated* web business intelligence systems; *i.e*. those web business intelligence systems in which content is acquired from the web entirely by an automated agent. AWBI systems by definition do not allow for manual data gathering from the web. Second, AWBI systems as operationalized herein are intended to provide a supplementary source of decision support information that can be integrated into an existing organizational decision making infrastructure. They are not intended to be standalone decision support systems. Third, the AWBI architecture is proposed only in support of organizational decision making for competitive advantage. AWBI systems do not fall within the realm of personal decision making agents such as those offered as services to individual web users in the financial, travel, and comparison shopping domains. The proposed AWBI architecture is depicted graphically in Figure 2.

As noted by Srivastava and Cooley [11], at the highest level web business intelligence systems can be conceptualized as performing two interrelated functions: (1) content acquisition, and (2) knowledge creation. Within the context of the AWBI architecture, content acquisition refers to the automated acquisition of relevant, detailed, and reliable information from the web, and knowledge creation refers to the generation of new knowledge through pattern discovery and prediction. The content acquisition component is by far the more problematic of these two functions. From a temporal perspective, content acquisition precedes knowledge generation and results in a structured, mineable data set. Whereas knowledge creation can be addressed by traditional data mining techniques, automated content acquisition is a far more challenging topic.

Content acquisition involves two distinct phases: (1) data retrieval, and (2) data extraction. These two phases are roughly analogous to the resource discovery and information extraction web mining subtasks identified by Etzioni [3]. Data retrieval is the process of gathering potentially relevant content from the web. It is important to note that data retrieval in the AWBI architecture relies on two separate sources of information: (1) established, reliable web sources, and (2) newly discovered web sources.

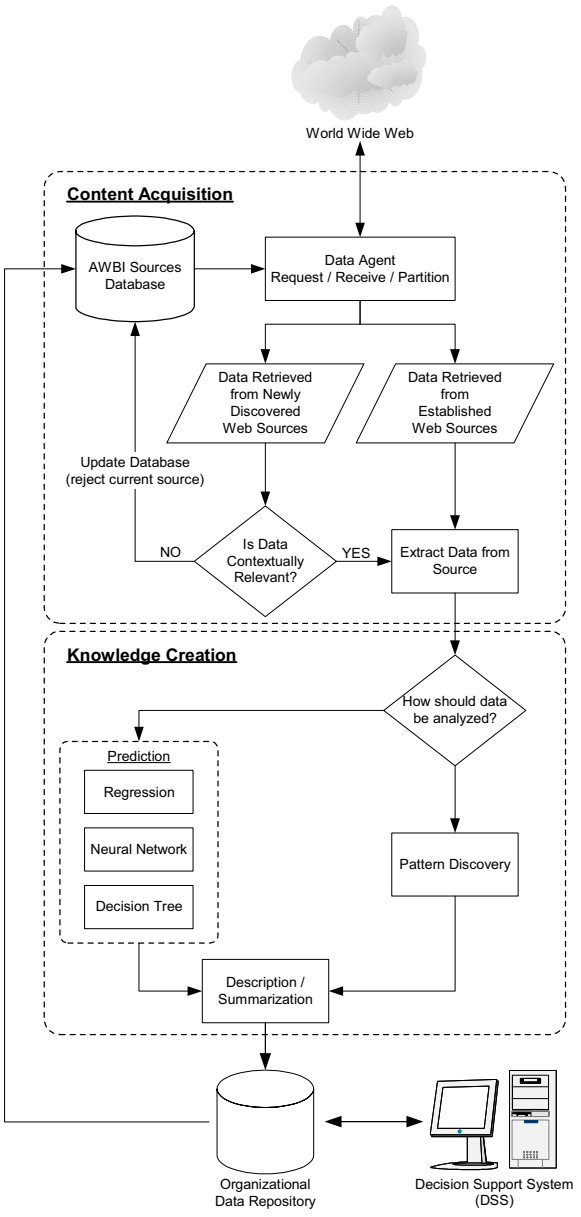

Figure 2. The AWBI architecture.

Every AWBI system should contain an automated exploratory agent, the purpose of which is to seek out and identify new sources of web data that may potentially be relevant to the organizational decision making process.

AWBI systems should not, however, rely solely on newly discovered web sources, but should establish semi-permanent relationships with web sources that are known to be dependable and trustworthy. These web sources can then be accessed at regular intervals by the AWBI system.

It should be noted that the overall dependability of a given web source for an AWBI system must be determined as a function of two dependability subtypes: (1) technical dependability, and (2) information dependability. The technical dependability of a web source refers to the availability and performance of the host on which the target information resides, whereas the information dependability of a given web source refers to the quality, reliability, and trustworthiness of the information made available thereby. A web source with low technical dependability, for example, may only be functionally available to serve information requests 60% of the time. Similarly, the information made available on a web source with low information dependability may only be accurate 60% of the time. In order for a web source to be considered acceptably dependable for use with an AWBI system, it must necessarily possess high degrees of both technical and information dependability. Furthermore, because AWBI systems are wholly reliant on the stability and availability of external data sources beyond the control of the organization, they should be designed to seek out redundant sources of information as a hedge against uncertainty. If a primary web source suddenly fails, the AWBI system should immediately and automatically utilize one of its redundant data sources to acquire the needed information.

A key component in the content acquisition process within the proposed AWBI framework is the data agent. The intent of the data agent is to broker the flow of information between the AWBI system and the web. It does this by negotiating with the AWBI sources database to request information from both known and unknown sources. When the requested information is retrieved from the web, the data agent partitions it based on prior knowledge regarding the relevance of the data source. If the source is established and is known to be dependable, then the data do not need to be examined for contextual relevance. Conversely, if the data are from a source unknown to the AWBI sources database, then the contextual relevance of the data must be determined.

Many automated techniques have been developed to ascertain the contextual relevance of a given set of data, the most prevalent of which is currently co-occurrence-based relevance ranking. Co-occurrence is a probabilistic approach to determining contextual relevance that assumes non-independence among search terms [13]. Models based on this approach determine the contextual relevance of a candidate document to a search query by employing non-linear weighting functions that are at least partially derived from the degree to which the index terms depend on one another. For example, given the search phrase “business intelligence”, an algorithm employing co-occurrence will assign a greater deal of relevance to a document containing the words “business” and “intelligence” if those words appear directly adjacent to one another in the document than it would if those words were to appear separately. From a design perspective, AWBI systems do not necessarily require their own intrinsic search capabilities. Rather, they can be designed to utilize the advanced retrieval and categorization functions of existing web search engines to facilitate their quest for new data sources.

Because none of the current ranking algorithms can be trusted to accurately determine the contextual relevance of a candidate web data source 100% of the time, a manual confirmation mechanism may be necessary. To that end, once the AWBI system has identified a new data source that it believes to be contextually relevant, the candidate data source can be presented to a user who will make the final decision as to whether or not the data source should be used. As such, a pragmatically designed AWBI system should limit the number of new data sources presented to the user for validation over a particular time span, so as not to be unnecessarily burdensome on her time. Within the AWBI framework, it is expected that many (if not most) of the new data sources will not be contextually relevant to the organization’s decision making requirements. To that end, when a data source is found to be irrelevant, it should be recorded as such in the AWBI sources database. This approach will prevent irrelevant data sources from being queried more than once, and will allow the AWBI system to “learn” what type of data sources the user finds most valuable. Given this information, the system can then refine its search criteria over time to yield more and more accurate results.

After a data source has been identified as being contextually relevant, it must be extracted from its original semi-structured or unstructured format. Ideally, an AWBI system should employ a machine learning algorithm as described previously to extract and classify the data embedded in the web source. This process must be fine-tuned to adhere to the specific extraction and classification requirements of the data repository on which the organizational decision support system relies. When this process is complete, the retrieved data can be analyzed with the objective of generating new actionable knowledge.

Contextually relevant information that is retrieved from the web can be either quantitative or qualitative in nature. As quantitative information is by definition numerically formatted, it lends itself exceptionally well to

prediction and summarization. Some of the most interesting and pertinent competitive information available on the web, however, is qualitative in nature. Analyzing information of this type requires that the information be either quantitatively transformed for analysis, or subjected to text mining and pattern discovery algorithms before it can be summarized and passed into the data repository on which the organizational decision support system relies. As with the extraction process, the means by which the data are analyzed must be determined in light of the specific extraction and classification requirements of the organizational decision support system.

4. The Prototype AWBI System

As a means of demonstrating the feasibility and efficacy of the proposed AWBI framework, two versions of a prototype system based on the architecture described in the previous section were constructed. The goal of both prototype systems was to use the web to extract information that would allow for the prediction of the direction of the Nasdaq-100 Index tracking stock (ticker symbol: QQQ) at the close of the following trading day. The extracted information would then be stored in a relational data repository and fed into a neural network for prediction. In both versions of this prototype system, the predictive neural network, which was constructed using the SAS Institute's Enterprise Miner™ software, was used as a proxy for an organizational decision support system.

The two separate versions of the prototype system were concurrently deployed to gather information from the web. These two systems were virtually identical, with the only difference being that the automated data gathering agent of one version (hereafter referred to as Version A) was allowed to look for and utilize new data sources while the other version's automated data gathering agent was not (hereafter referred to as Version B). This approach was taken to allow for a comparison of the predictive efficacy of the neural network constructed using the data gathered by Version A of the system against a baseline neural network constructed using the data gathered by Version B.

Both versions of the system were initially provided with a single web data source from which to gather information. The web data source was one of the more popular finance web sites, and due to its reputation was assumed not only to be technically dependable, but also to provide reliable and trustworthy information. As the information identification and extraction algorithms of the two versions of the system were identical, both versions initially identified the same fourteen quantitative information fields on the web source for retrieval. Both versions of the system were instructed to run once per trading day at the same time in the evening after the markets had closed.

The data gathering phase took place over the 100 trading days in which the financial markets were open between January 2, 2004 and May 25, 2004. During the first twenty trading days of this time span (January 2nd – January 30th), Version A of the system was allowed to suggest up to five new, non-redundant data sources per day that it believed to be contextually relevant to the information fields it had previously identified. The decision of whether to use the suggested data sources was left to the author. Both versions of the system were allowed to search for and suggest new redundant web sources for the previously identified information fields in the database for which they did not currently have a backup source. During this "learning" phase, Version A of the system expanded its set of contextually relevant information fields from fourteen to seventy-nine, and both versions of the system found redundant sources for each of their respective information fields. At the conclusion of the twenty day "learning" phase, the automated data gathering agent of Version A was instructed to no longer search for new data sources, and both versions were instructed to no longer search for redundant data sources.

The next eighty days in which the financial markets were open for trading (the period between February 2nd, 2004 and May 25th, 2004) comprised the primary data gathering phase for both versions of the prototype AWBI system. During this time, both versions of the system retrieved their target data from the web on a daily basis as described above and stored those data in a relational data repository. It should be noted that several retrieval failures from the data sources identified by Version A occurred during this time, however in every case the redundant data source was able to provide the necessary missing information. Automated data retrieval from the initial web source was accomplished dependably without interruption, thereby yielding a complete data set for Version B of the system. At the conclusion of the primary data gathering phase both versions of the prototype system were deactivated.

Two different neural networks were constructed with the goal of predicting the direction of the Nasdaq-100 Index tracking stock at the close of the following trading day: the first using the data gathered by Version A of the prototype AWBI system and the second using the data gathered by Version B. Both data sets were partitioned using stratified random sampling such that 75% of the data were used to train the neural network, 15% of the data were used for validation, and 10% were used for testing. All of the prediction variables in each data set were transformed to maximize their normality before being subjected to the variable selection process.

Predictive variables for each neural network were selected using an R-square selection criterion in which a candidate variable was rejected if its R-square improvement was less than 0.005 or if its stepwise R-square improvement was less than 0.0005. Two-way interactions were included in the variable selection process for both neural networks. The results of the neural network assessment for each test data set are shown in Table 1 below.

Table 1. Results of neural network assessment.

	Version A	Version B
Average Error	0.549	4.917
Sum of Squared Errors	3.045	8.944
Mean Squared Error	0.190	0.407
Misclassification Rate	**0.287**	**0.455**

As shown in the table, the misclassification rate for the data gathered by Version B of the prototype AWBI system was approximately 45.5%, whereas the misclassification rate for the data gathered by Version A of the prototype AWBI system was approximately 28.7%. These results indicate that the neural network created using the data gathered by the version of the prototype AWBI system that was allowed to look for and utilize new data sources was able to correctly predict the direction of the Nasdaq-100 Index tracking stock at the close of the following trading day approximately 71.3% of the time. This is in stark contrast to the baseline neural network, which was only able to correctly predict the direction of the Nasdaq-100 Index tracking stock at the close of the following trading day approximately 54.5% of the time.

Recalling that the two neural networks described above were used in this scenario as a proxy for an organizational decision support system, it seems clear that an organization that actively trades the Nasdaq-100 Index tracking stock would greatly prefer to rely on the prediction made by the neural network constructed using Version A's data as opposed to the neural network constructed using the data gathered by Version B. The observed difference in predictive efficacy between the two neural networks can be directly attributed to the automated identification and utilization of new web sources that was made possible by Version A of the prototype AWBI system. It can therefore be concluded that within the scope of this scenario, the prototype AWBI system built on the architecture described in the previous section led directly to improved decision making performance, thereby providing evidentiary support in favor of the proposed AWBI framework for use as a supplementary source of decision-making data for organizational decision support systems.

5. Summary and Future Research Directions

This paper proposed a contemporary architecture by which organizations can implement Automated Web Business Intelligence systems in support of their overall decision support strategy. The goal of AWBI systems is to automatically acquire contextually relevant information from the web regarding the environment outside the organization, and use that information to inform the corporate decision making process, thereby yielding competitive advantage. Although the prototype described herein lends support to the notion that AWBI systems can serve as a valuable addendum to corporate decision support endeavors, most organizations have yet to undertake an AWBI initiative. As such, the AWBI architecture presented in this paper is both timely and desirable.

Future work in this area will likely focus on specific aspects of the AWBI framework, as well as on the side effects of AWBI proliferation. One particular area of interest is that of corporate countermeasures to competitors' AWBI implementations. If major competitors in a given market are aware of each other's AWBI-related efforts, then it is likely that they will endeavor to interfere with those efforts. For example, if Company X has an AWBI system that extracts the prices charged by Company Y for goods and services from Company Y's website, and is using that information to inform its own product pricing strategy, then Company Y may intentionally make its prices difficult for Company X's automated agent to extract from the web.

Another area of future research interest for ABWI systems is that of qualitative corporate "buzz". Clearly, AWBI systems can be designed and implemented with the sole purpose of automatically monitoring competitive trends, news sources, and press release feeds. The intent of AWBI systems such as these would be to act as an ever-vigilant sentry, continuously trolling the murky waters of the web for any sign of competitive maneuvering or shift in a competitor's corporate strategy. The value to organizational decision makers of having this sort of information available to them in real-time can hardly be overestimated. Ultimately, AWBI systems of this sort may have the potential to replace corporate espionage activities altogether. To that end, research related to these types of AWBI systems may indeed be very fruitful.

References

[1] Chang, C.-H.; Hsu, C.-N.; and Lui, S.-C. "Automatic Information Extraction from Semi-Structured Web Pages by

Pattern Discovery." *Decision Support Systems*, 35, 1, (2003), 129-147.

[2] Cooley, R.; Mobasher, B.; and Srivastava, J. "Web Mining: Information and Pattern Discovery on the World Wide Web", in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, Newport Beach, CA, 1997.

[3] Etzioni, O. "The World Wide Web: Quagmire or Gold Mine?" *Association for Computing Machinery. Communications of the ACM*, 39, 11, (1996), 65-68.

[4] Guan, T., and Wong, K.-F. "Kps : A Web Information Mining Algorithm." *Computer Networks*, 31, 11-16, (1999), 1495-1507.

[5] Hammond, K.; Burke, R.; Martin, C.; and Lytinen, S. "Faq-Finder: A Case-Based Approach to Knowledge Navigation", *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous Distributed Environments*, AAAI Press, 1995.

[6] Konopnicki, D., and Shmueli, O. "WWW Information Gathering: The W3ql Query Language and the W3qs System." *ACM Transactions - Database Systems*, '98, 1, (1998).

[7] Milhaila, G.A. *Websql - a SQL-Like Query Language for the World Wide Web*, Master's Thesis, University of Toronto, (1996).

[8] Perkowitz, M., and Etzioni, O. "Category Translation: Learning to Understand Information on the Internet", in *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995, 930-936.

[9] Spertus, E. "Parasite: Mining Structural Information on the Web", in *Proceedings of the Sixth International World Wide Web Conference*, Amsterdam, 1997, Elsevier.

[10] Spertus, E., and Stein, L.A. "Squeal: A Structured Query Language for the Web." *Computer Networks*, 33, 1-6, (2000), 95-103.

[11] Srivastava, J., and Cooley, R. "Web Business Intelligence: Mining the Web for Actionable Knowledge." *INFORMS Journal on Computing*, 15, 2, (2003), 191-207.

[12] Sullivan, T. "Business Intelligence Keeps Tabs on the Net." *InfoWorld*, 23, 10, (2001), 33.

[13] Van Rijsbergen, C.J. "A Theoretical Basis for the Use of Co-Occurrence Data in Information Retrieval." *Journal of Documentation*, 33, 2, (1977), 106-119.

[14] Zamir, O., and Etzioni, O. "Grouper: A Dynamic Clustering Interface to Web Search Results." *Computer Networks*, 31, 11-16, (1999), 1361-1374.