A Web Mining process for e-Knowledge services

M. Castellano[1], F. Fiorino[2], F. Arcieri [2], V. Summo[2]
& G. Bellone de Grecis[2]
[1]*Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy*
[2]*Global Value – ACG srl – An IBM Company, Italy*

Abstract

The purpose of this paper is to describe a process of Web Mining in order to support specialized e-Knowledge services. Here is proposed a new reference architecture based on an orchestration of reusable building blocks, with well defined tasks and the ability to interoperate among them. The system is designed to support a decision maker in a service-oriented way, by adopting a clear separation of tasks: crawling, pre-processing, information extraction, information retrieval, text mining and presentation of results. It allows the analysis of Web information by extracting, selecting, processing and modelling huge amounts of data, in order to discover rules and patterns in a distributed and heterogeneous content environment of informative resources. Finally, as a case study, the Reputation Management process is presented.
Keywords: Web Mining, text mining, Web crawling, information extraction, information retrieval, reputation management.

1 Introduction

The digital universe known as the World Wide Web is a very huge place that includes literally billions of Web pages, and is estimated to continue to grow at an accelerating rate of 7,3 million pages per day (Cyveillance, 2003). Moreover, with this amount of data available online, the WWW is today considered a popular and interactive medium to disseminate information. At the beginning, it was an instrument primarily used by universities and research communities; nowadays it represents a tool of easy access and insert of information [3, 8]. Moreover, the available information is extremely distributed and heterogeneous:


WIT Transactions on Information and Communication Technologies, Vol 37, © 2006 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA060261

80 percent of this data is unstructured, such as HTML page, e-mail, spreadsheets, symbolic text, images, hyperlinked data, audio and videos, all difficult to be interpreted and analyzed. These factors give rise to the necessity to create intelligent systems that can effectively mine the Web for knowledge discovery. To achieve this goal, *Web Mining* can be broadly defined as the discovery and analysis of useful information from the World Wide Web. In this paper, firstly, we introduce a process of Web Mining for the realization of e-Knowledge services, then our crawling architecture and our mining engine are described; finally a case study is presented. It is important to point out that the purpose of this paper is not to present new algorithms or other techniques, but to show how to adopt them in a workflow that allows the optimization of the Web Mining process.

2 The Web Mining process for the knowledge discovery

The primary objective of a Web Mining process is to discover interesting patterns and rules from data collected within the Web space. In order to adopt generic data mining techniques and algorithms to Web data, these data must be transformed into a suitable form.

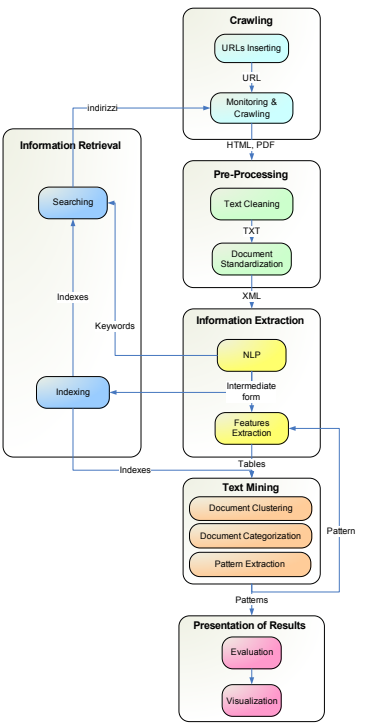

Figure 1: The Web Mining process for building an e-Knowledge service.

In Figure 1 we describe the workflow of the overall Web Mining process for the realization of an e-Knowledge service. The idea is to connect specific research domains such as Information Retrieval, Information Extraction, Text Mining and so on, and to put them together in an innovative process of workflow defining several phases and steps, as shown in Figure 1. As remarkable result, by adopting this methodology every e-Knowledge service may be represented and implemented, moreover they can share common activities, facilitating reuse and standardization.

2.1 Crawling

Nowadays, the real problem in the world of the Web is that there is no catalogue of accessible URLs; the only way to get them is to scan pages of interest in order to find hyperlinks to new other pages not yet gathered. This is the basic principle for crawling the Web. It starts from a given set of URLs (URLs Inserting step), progressively fetch and scan them, in turn in an endless cycle, for new URLs (Monitoring & Crawling step), that represent a potentially pending work for the crawler [9]. The main challenges for a crawler can be summarized into two main issues:

- Externally, it must avoid overloading Web sites or network.
- Internally, it must deal with huge amount of data. Unless it has unlimited computing resources and unlimited time, it must carefully have to decide which URLs to scan and in which order; the crawler must also decide how frequently to revisit pages it has already seen.

At the end of the Crawling phase many different types of documents (HTML, PDF, DOC, etc) are collected from the Web.

2.2 Pre-processing

The fetched documents are cleaned and then standardized for further analysis; in the Text Cleaning step, major problems rise up with erasing all the un-necessary information such as HTML and XML tags and titles and meta-tag content in order to avoid problems of spamming. Then, in the Document Standardization step it is helpful to convert documents into a XML standard format in order to identify theirs parts. The main reason for identifying the pieces of a document is to allow selection of those parts that will be used to extract features.

2.3 Information extraction

This phase includes operations that transform textual information into numerical vectors. The first step uses the Natural Language Processing (NLP) techniques in order to obtain an intermediate form useful for the next step of feature extraction.

- *Natural Language Processing*: the NLP includes a set of techniques, such as Tokenization, Lemmatization, Vector Generation, Part-of-Speech Tagging, Feature Recognition, works on textual data in order to extract useful

information. The main operation is to break the stream of characters into words or tokens and reduce their number through various techniques such as consulting Stop-word list [1], converting plurals into singulars, adopting stemming technique by stripping the root of its derivational and inflectional affixes (i.e. suffix, prefix, and infix). That is, all words are transformed to their canonical form in order to create the intermediate form, ready for further elaborations.

- *Feature Extraction*: This step identifies facts and relations in text. It often happens that this includes a person, place, organization or other distinct object. Feature extraction algorithms may use dictionaries to identify some term and linguistic patterns to detect others. The extracted term has usually to be in canonical or standard form. This makes indexing retrieval and the other steps which follow more accurate.

2.4 Information retrieval

The Information Retrieval phase deals with the access and the organization of the information items coming from the Web; unfortunately this is not a simple problem because the aim is to translate an information need into a query that can be processed by the Web crawler. The main steps of the IR phase are:

- *Indexing*: This step reduces the amount of text in a document by keeping and managing its key words. It is possible to define a number of parameters, including the number of sentences to extract or a percentage of the total text to extract. The result includes the most significant sentences within the document.
- *Searching*: It is used to search internal document collections or external Web document starting from indexes and keywords. A wide range of text search option may be utilized such as Boolean (and/or/not), proximity, wildcard, segment, numeric range, relevancy-ranked natural language searching, fuzzy search, concept search, etc. As result this step produces a list of URLs that to crawl.

2.5 Text mining

This phase includes the application of generic data mining techniques and specialized algorithms for text. Most Text Mining objectives fall under the following steps:

- *Document Clustering:* A cluster is a group of related documents and clustering is the operation of grouping documents on the bases of some similarity measure, automatically without having to pre-specify categories. The most common clustering algorithms that are used are hierarchical, binary relational and fuzzy. The main factor in a clustering algorithm is the similarity measure such as Textual Similarity, Shared Word Count, Word Count and Bonus and Similarity Cosine.

- *Categorization:* In this step documents are classified into predefined categories. This operation has been adopted when it is possible to identify

the main topics of a document collection. There are two approach to create the classification: in the first one the categorizer uses a pre-defined thesaurus that defines a set of domain-specific terms and relationship between them; in the second one it creates its own thesaurus starting from sample documents.

- *Rules based:* In this step unknown patterns are identified using trend and association analysis. This operation requires the biggest effort in the Web mining process and it can be executed using different techniques:
  - *Predictive Text Mining:* This technique is used for identifying trends in documents collected over a period of time. Trends can be used, for example, to discover that a company is shifting interests from one domain to another
  - *Association Analysis:* This technique identifies relationships between attributes, such as the presence of one pattern implies the presence of another pattern, in a given set of documents. For example, *Neurosoft SA, Intrasoft* → *Take over* could be a rule that has been discovered.

2.6 Presentation of results

Presentation techniques are finally adopted to render patterns and display results. The purpose of the first step is to evaluate the degree to which the extracted models fit the business objective and if this is not the case, determine the reasons why the model is not good or has been overlooked. In the second step, discovered patterns, results and conclusions are presented and distributed to users. A graphical representation of the document collection supports users in quickly identifying the main concepts or topics.

3 A reference architecture for building e-Knowledge services

In this paragraph we propose a Knowledge Discovery architecture [5] as a modular and flexible structure. The purpose is to generate and make available e-Knowledge services by using Web mining techniques, by extracting, selecting, processing and modelling huge amount of data in a distributed and heterogeneous content environment of informative resources, coming from the Web. As our architecture wants to be an e-Knowledge services provider, a basic requirement is that each service has to be built as a module, independent from other ones. Each e-Knowledge service will represent the result of an orchestration of reusable building blocks, with well defined tasks and able to interoperate among them. Moreover, the possibility to add one or more service in the system must be guaranteed every time without doing any substantial change, so to have full control and a complete supervision. Another important consideration is the flexibility. Usually users have different business goals in mind when they need to discover hidden knowledge in their data. Hence, the architecture should be flexible in supporting various mining techniques and algorithms. This can be achieved by providing a clear separation between the

process logic and the e-Knowledge services that the architecture has to provide. Our proposed system is mainly composed by the following architectures:

- An architecture for Crawling and Monitoring the Web that makes use of Web Content Mining and Web Structure Mining techniques to retrieve information from the Web.
- A mining engine for Pattern Extraction that takes the user and the various e-Knowledge services through all three stages (information extraction, pattern extraction and evaluation) of the knowledge discovery process in a unique workflow.

3.1 An architecture for crawling and monitoring the Web

The architecture we are going to describe realizes a Web Crawler for the extraction of useful knowledge from the Web. The reference solution covers the first step of the Web Mining process [12, 13], dealing with automatic retrieval of all relevant documents and ensuring at the same time that the non-relevant ones are fetched as few as possible. Documents to be examined will be semi-structured, such as html and xml pages, and not structured, such as simple text.

The architecture fits the guidelines of a Focused Crawler [10], as it is designed to only gather documents on a specific topic, thus reducing the amount of network traffic and downloads. It can also be seen as an Incremental Crawler [11], as it retrieves Web pages already examined whose content could has previously been modified. In accordance to a specified service to provide, the administrator of the system creates a high interest URL list. After the crawling of these documents, the most interesting keywords are extracted through Information Retrieval algorithms and, opportunely combined, represent the input for a remote search engine. The aim of the last operation is to enlarge the set of documents in the field of interest with the URLs returned by the search engine. This is useful to discover new and more relevant information. As search engine, we have considered Google tools. In the next figure, the Web Crawler Architecture [7] is shown.

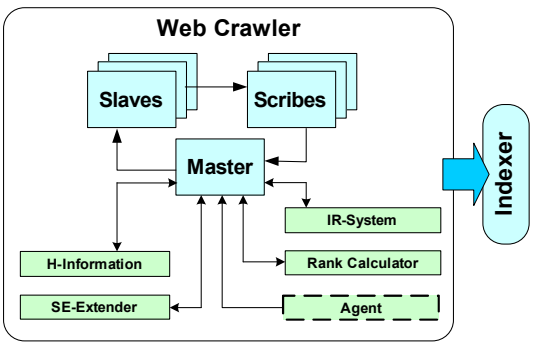

Figure 2: The Web Crawler Architecture.

The main components of the Web Crawler are as follows.

Slave. Slaves are responsible to accede to the Web with the scope to crawl the pages associated to the URLs indicated from the *Master*. Each *Slave* has given an own set of pages. In this way, different *Slaves* can operate at the same time to render more efficient the access to Web sites, reducing the answer times and avoiding simultaneous accesses to the same resources [2]. The documents retrieved from the Web are html and xml pages, pdf, rtf, doc and ps files.

Scribe. Scribes receive the Web pages crawled from the *Slaves*. For each *Scribe* there is a *Slave*. The *Scribe* parses the entire page and captures different information according to the type of the examined document. For example, in the case of html and xml pages, the *Scribe* parses them and considers all the tags of the body. Some information, such as titles and metatag contents, are considered not so important in order to avoid problems of spamming information [9].

Master. The Master is the core of the Web Crawler and allows different functions:

- sending URL pages to crawl to the *Slaves* according to the *H-Information* list;
- opportunely balancing the workload to assign to the various *Slaves,* avoiding the risk to execute simultaneous accesses to more pages of the same site;
- receiving information from the *Scribes* about the document previously extracted. For example this is useful to verify that crawled pages are not already present. If not, they will be stored, otherwise they will be not considered;
- receiving from the *IR-System* the keywords of the retrieved documents and sending them to the *SE-Extender* with the task to make a new query;
- sending to the *Rank Calculator* the URL list received from the *SE-Extender*, that will give back a new list containing only the URLs having a ranking value higher than a fixed threshold.
- sending to the *H-Information* the new URLs useful to widen the starting list of the pages to crawl;
- analyzing links from all retrieved documents to control if a high number of them are directed to Web pages still not gathered. In this case these new pages will be crawled;
- re-executing the process of the crawling according to the information coming from the *Agent*.

H-Information (Human Information). The H-Information represents an interface that allows the system administrator to create a list of high interesting Web sites to be crawled. This list will be continuously increased by the Master with the news URLs computed by the *Rank Calculator.*

IR-System (Information Retrieval System). The IR-System has the task to extract meaningful keywords from text document by using Information Retrieval techniques [1]. Information Retrieval has the primary goal of indexing text and searching for useful documents in collections, but we have mainly adopted it for feature extraction, similarity, modeling, document classification and categorization and filtering.

SE-Extender (Search Engine Extender). The SE- Extender receives from the *Master* the keywords extracted by the IR-System, fit them as input to a search engine, such as Google, and returns a URLs list.

Rank Calculator. The Rank Calculator calculates a ranking metric for URL returned from the search engine. By resuming some characteristics of the well known PageRank algorithm [2], our algorithm recursively defines the importance of a page *A* to be the weighted sum of the importance of the pages that link to *A*, by considering the number of outgoing links of each page. The system assigns to the URL list created by the *H-Information* a value equal to 1, the maximum possible value, because of the importance of these pages for the service to supply. In this way, if a page is not linked from none retrieved document, it is ignored because its ranking value results shorter than the fixed threshold.

Agent. The Agent has the responsibility to re-execute the whole process of crawling according to the needs of updating the Knowledge Repository, for example after well defined time expiration.

As remarkable results the proposed Web Crawler architecture is an efficient retrieval system of information from the Web, useful for providing e-Knowledge services for e-Government and e-Business organizations. The primary goals are to retrieve quality information from the Web trying to reduce the acquisition of non-relevant documents for the service to provide. As scalability, through the component of the Master, the architecture is able to manage the access to Web sites by reducing the necessary time for crawling and, at the same time, by avoiding an overload of them. Moreover, the system automatically re-executes the whole process of crawling through the *Agent* component, in order to refresh information stored in the Knowledge Repository.

3.2 A Mining Engine architecture for the Pattern Extraction

Mining systems and technologies are currently considered as enablers for business intelligence systems, because they improve the quantitative and qualitative value of the knowledge available to decision makers. Nowadays, the architecture for a mining system [6] has a remarkable impact especially for large business environments, where data from numerous sources needs to be accessed and combined to provide comprehensive analyses, and work groups of analysts require access to the same data and results.

The referenced Mining Engine [5], designed in Figure 3, represents a distributed Web mining tool, where a set of e-Knowledge services are managed and made available. Main components are a Controller, a Kernel and a set of Miners.

When an e-Knowledge service request comes to the system, this is received and analyzed by the Controller and then forwarded to one or more Miner to provide the results. The Miners are building blocks that can be used to build a complex application. In this case, the application is represented by an e-Knowledge service and it may be formed by one or more Miner that represents the business logic of the service.

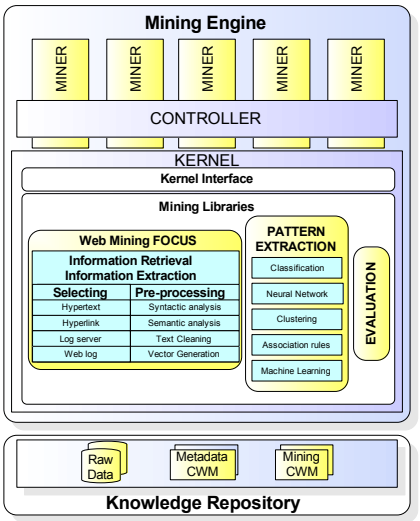

Figure 3: The Mining Engine Architecture.

A Miner has different tasks:

- To load the mining models associated to a required e-Knowledge service;
- To call the Web Crawler if it needs data from the Web for a specific service;
- To drive a training process in the Kernel, by activating the process of the KDT in order to rebuild mining models according to the presence of new data in training sets.

The Kernel represents the core of the system and covers the knowledge discovery process by working on data and by building new mining models; it is mainly composed by the following components:

- Web Mining Focus holds the phase of the Information Extraction, where preliminary steps of the Knowledge Discovery in Text, such as selecting and pre-processing of hypertext, hyperlinks or logs, take part.
- Pattern Extraction where a set of Web Content, Structure and Usage Mining techniques [7–9], together with other Data Mining algorithms, are collected for the analysis.
- Evaluation interprets the utility and the carefulness of the extracted patterns. This last step is important because only a part of the extracted patterns is really of interest and observes the knowledge and the objectives of the service required by the user.

The Kernel activity is supported by the Mining Libraries that represent a set of Data and Web Mining algorithms, able to solve each stage previously described.

The main advantages in adopting our approach are reuse, standardization of e-Knowledge services, easier integration of diverse development staff, easier introduction of new technologies, more flexibility in combining mining operation and algorithms together and scalability through a distributed environment.

4 A case study: the Reputation Management process

An organization's most valuable asset is its reputation because when corporate reputation is damaged, it puts the business at risk and this can quickly cost millions of dollars in lost revenue. The ability to measure reputation and manage outcomes is extremely elusive but it is a critical feature of company success and marketing strategies. The data sources for a reputation application can be found everywhere: Web sites, documents, forum, blogs, RSS feeds, etc. Understanding the impact of new media on corporate reputation is a growing challenge and an important component of marketing intelligence. A powerful way to detect the complex signals of emerging reputation trends is adopting a Web mining process and new analysis techniques based on text mining. The Web mining process looks for patterns and trends in natural language text and measures subtle differences in words and phrases that are important for the organization and its reputation. According to the Web mining process described in the previous paragraph, the process of creation of reputation management can be divided into steps, i.e. Crawling, Pre-processing, Information Extraction, Text Mining, Information Retrieval and Presentation of Results, where each step can be made up by one or more Miners that collaborate among them. More in detail, the *Crawling step* firstly aims to gather data from unstructured sources such as Web sites and blogs. Moreover, to get a complete view of emerging trends, the best reputation solutions must also face the electronic versions of newspapers, magazines, trade journals and wire services. The value of a reputation application is only as good as the sources that are mined. In our experimental scenario there is only a selected set of sites to crawl, basically:

- Inbound Links: the Web sites referenced from our pages;
- Outbound Links: the Web sites that references our pages.

The *Pre-Processing* step transform and deliver all fetched information into a standard format. In the next step of *Information Extraction* there is the customization and the integration of specific domain knowledge to successfully refine the sheer volume of data available with NLP and Feature Extraction techniques. In detail in the part-of-speech tagging step (miner):

- There is an association between every keyword and their noun form (positive or negative).
- There is an association between every keyword and their verb form (positive or negative).

The core step of *Text Mining* allows one to make sense of complex information by searching through large amounts of unstructured data. Text Mining can be performed by a collection of methods from various technological areas such as statistical methods, neural networks, inference rules, logical proposition, taxonomies and ontologies. Each technique is implemented as a specific Miner in order to balance the flexibility and adaptability of the reference architecture. In this case study we adopt classification methods in order to search keywords such as: Innovation, Tradition, Flexibility, Reliability, Quality, Trust ness, Solidarity and Impartiality; that represent important topics for a corporation. In

the *Information Retrieval* step, patterns such as word proximity and sentence structure identify additional meaning from text and extract important information for further crawling. For a best search, each reputation application needs to be fine-tuned for relevant meaningful keywords and their relations. Finally, the *Presentation of Results* step facilitates the discovery of trends and patterns over time. Tracking both the rise and fall of the trend helps to provide clues on the emerging or declining importance of issues or comparing an organization to its competitors. Finally there are two main views:

- A Static matrix where the frequencies of the keywords, both positive and negative, are shown for our company and for its competitors;
- A Dynamic chart for our company for the tracing of the trend of the searched keywords.

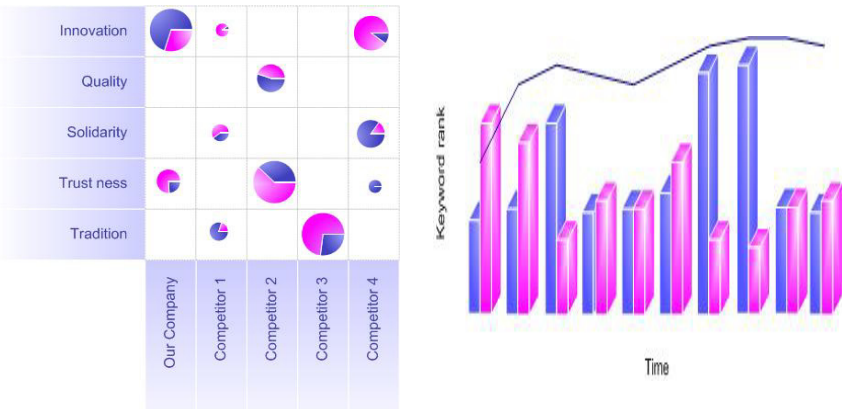

Figure 4: Experimental results.

These results furnish to company decision makers a way for

- Tracking public's perception of company, brand, products and services;
- Comparing issues against competitors;
- Access the full documents.

5 Conclusion

In this paper we have presented a new approach to build a Web Mining process able to provide new value added e-Knowledge services, such as the Reputation Management Service. As remarkable result, it has been shown how to implement the process through a Web Crawler architecture, as information retrieval system for data coming from the Web, and a Mining Engine, able to decouple the management of e-Knowledge services and their building. As future works, an improved and extended set of e-Knowledge services will be designed and realized, using the system building blocks and inserting new ones.

Acknowledgement

The authors acknowledge the financial support provided by the Italian Ministry of Education, University and Research which has made possible the realization of this work as result of our research activities.

References

[1] Baeza Yates, R. and Ribeiro Neto, N, Modern Information Retrieval. Addison Wesley, Essex, England, (1999).
[2] Brin, S. and Page, L., The Anatomy of a Large Scale Hypertextual Web Search Engine. In Proceeding of the 7th International World Wide Web Conference. Brisbane, Australia, pp. 107-117, (1998).
[3] Carlson, B., Drinking from the fire hydrant. From ZDNET, http://zdnet.com.com/2100-1107-5056947.html, (1998).
[4] Castellano, M., Pastore, N., Arcieri, F., Summo, V., Bellone de Grecis, G., A Flexible Mining Architecture for Providing New E-Knowledge Services, In the Proceedings of the 38th HICSS, Hawaii Int. Conference On System Sciences, Big Island, Hawaii, Computer Society Press, (2005).
[5] Castellano, M., Pastore, N., Arcieri, F., Summo, V., Bellone de Grecis, G., A Knowledge Center for a Social and Economic Growth of the Territory. In Proceedings of the 38th HICSS, Hawaii Int. Conference On System Sciences, Big Island, Hawaii, Computer Society Press, (2005).
[6] Castellano, M., Pastore, N., Arcieri, F., Summo, V., Bellone de Grecis, G., A Model-View-Controller for Providing e-Knowledge Services. In Proceedings of DATA MINING 2004, 5th International Conference on Data Mining, Text Mining and Their Business Applications, Malaga, Spain, (2004).
[7] Castellano, M., Pastore, N., Arcieri, F., Summo, V., Bellone de Grecis, G., A New Architecture for Crawling the Web. In proceeding of e-Society 2004, IADIS International Conference, Avila, Spain, (2004).
[8] Chakrabarti, S., Data Mining for Hypertext: A Tutorial Survey. ACM SIGKDD Explorations, Vol. 1, No. 2, pp. 1-11, (2000).
[9] Chakrabarti, S., Mining the Web: Discovery Knowledge from Hypertext Data. Morgan Kaufman Publisher, San Francisco, USA, (2003).
[10] Chakrabarti, S. et al, Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. In Proceeding on the 8th International Word Wide Web Conference. Toronto, Canada, pp. 1623-1640, (1999).
[11] Cho, J. and Garcia-Molina, H., The evolution of the Web and implications for an Incremental Crawler. In Proceeding of the 26th International Conference on Very Large Data Base. Cairo, Egypt, pp.117-128, (2000).
[12] Cooley, R. et al, Web Mining: Information and Pattern Discovery on the World Wide Web. In Proceeding of IEEE International Conference Tools with AI. Newport Beach, California, USA, pp. 558-567, (1997).
[13] Etzioni, O., The World Wide Web: Quagmire or GoldMine? Communication of the ACM, Vol. 39, No. 11, pp. 65-68, (1996).