Comparing Semantic Web and Data Mining

Veljko Milutinovic[1],

Abstract - **The fields of semantic web and datamining are currently emerging and creating lots of scientific and commercial interest. The two fields are typically analyzed in isolation from each other. This paper represents an effort to treat them as two different approaches to the same final goal, and to treat them comparatively. In addition, it explains the essential issues of the two approaches, and gives some predictions about the future development trends.**

I. INTRODUCTION

A major goal of both datamining and semantic web is efficient retrieval of knowledge from large databases (single or distributed) or the Internet. In this context, the knowledge is treated through a synergistic interaction of information (data) and their relationships (links within a typical relational database or links on the web). Synergistic interaction implies also the cases in which the meaning of data differs from the cases when data is represented in isolation, to the cases when data is linked with other data, which is a special challenge for research efforts aimed at efficient knowledge retrieval.

If datamining and semantic web are compared from the point of view of how they facilitate retrieval of knowledge, a major difference is in the placement of complexity. In the case of datamining, complexity is (conditionally speaking) placed at run time and retrieval time. In the case of semantic web, complexity is (conditionally speaking) placed at compile time and design time.

In the case of datamining, data and knowledge are represented with simple mechanisms (typically based on HTML) and typically without metadata (data about data). Consequently, relatively complex algorithms have to be used, which means that complexity is migrated to the retrieval request time. In return, there is no complexity at system design time – one uses well developed algorithms and their standard implementations.

In the case of semantic web, data and knowledge are represented with complex mechanisms (typically based on XML), and with plenty of metadata (sometimes, a byte of data – a name – may be accompanied with a megabyte of data – descriptive information related to that name). Consequently, relatively simple algorithms can be used for data retrieval, which means that complexity placed at the data retrieval time is minimal. However, large and sometimes relatively sophisticated metadata have to be created at system design time – one has to invest large efforts into the metadata design, preprocessing, postprocessing, and general maintenance.

[1]Veljko Milutinovic, Fellow of the IEEE School of Electrical Engineering, University of Belgrade, Serbia

Major knowledge retrieval algorithms used with datamining are neural networks, decision trees, rule induction, memory based reasoning, and many others. Consequently, the stress in the datamining review part of this paper is on algorithms.

Major metadata design, processing, and maintenance tools used in semantic web are XML, RDF, and ontology languages. The ongoing research concentrates on issues like logic, proof, and trust. Consequently, the stress in the semantic web review part of this paper is on tools.

The rest of this paper is divided into three parts: an overview of datamining, an overview of semantic web, and conclusions that include trend predictions. With this final issue in mind (trend predictions), the two overview parts stress the point to be elaborated in the trends prediction part.

II. DATAMINING

This section contains a condensed overview. A detailed overview can be found in [1], which is a tutorial. That tutorial can be found on the web site of the author, and was presented many times at conferences, in house for industry, or as a university course, worldwide. Primarily, the issues are stressed which represent either the important bottlenecks of the approach or the potential solutions for the general problem of recognition of semantics in cases when data may change its meaning from one context to the other.

There are three major differences between datamining and database engineering: (a) Uncovering the hidden knowledge, (b) Treating the huge n-p complete search space, and (c) Implementing a multidimensional interface to the user.

With databases, one can do only the data retrievals conceptualized at the database design time. If a query is placed which is planed at the database design time, the database will deliver the requested information. However, if a query is made which is not predefined, the database will deliver a question mark! On the other hand, a datamine is supposed to be able to deliver answers even in such cases. This means that a major difference is in layers of intelligence that have to be placed on the top of a database, to create a datamine.

Next, traditional databases are typically much smaller compared to datamines, especially if datamining is done in the context of the entire Internet. This extra-large size means that linear search algorithms (sometimes used in the database environments) are absolutely useless in datamining environments.

Finally, the retrieved knowledge (in the case of datamine search) has to be presented to the user in a way which is easy to comprehend, especially in situations when the

meaning is dependent on the context. This requires complex graphical interfaces. On the other hand, in the case of database search, information is comprehensible even if presented in the form of tables or histograms or similar.

One possible definition of datamining implies that it represents automated extraction of predictive information from memory (large databases or the Internet), or communication lines (cell phones or data channels in general). With this in mind, the rest of this section concentrates on datamining problem types, algorithms, models, as well as some available software.

One can talk about a number of different problem types in datamining (data description and summarization, segmentation, classification, concept description, prediction, and dependency analysis), but in real systems, most of the time, one can recognize a combination of several problem types. This is important to know, because some of the algorithms (to be elaborated later) work better for one problem types, while other algorithms work better for other problem types. Consequently, if we have a combination of problem types, we have to use a combination of algorithms. As it will be seen later, especially in the case of less complex and less expensive tools, one tool supports one type of algorithm. So, treating a problem with various algorithms typically implies the usage of several tools.

One widely used class of algorithms is neural networks. These algorithms are especially useful if the nature of the problem is not well defined, and it is difficult to determine an exact explicitly defined algorithm for problem treatment. The approach uses an analogy with biological neurons and utilizes the so called artificial neurons.

Another widely used algorithm is decision trees. This algorithm is especially useful if all decision making parameters and conditions are well defined, and precise processing rules can be created. The approach uses if-then-else and case structures, to define all relevant rules.

Still another widely used algorithm is rule induction. This algorithm is used in situations when various opinion creators/leaders have different opinions, and it is not possible to set precise rules. Instead, a statistical set of rules is created, and it is allowed that various rules of the set contradict with each other. The approach uses rule definitions with specifications of confidence levels and weights.

The memory based reasoning approach is used much more widely than in datamining alone; it is used also in court practices, etc. This algorithm is used in situations when we have to reduce the problem size, in order to be able to apply more sophisticated algorithms only to a subset of cases that can not be resolved with memory based reasoning. The approach uses the concept of history size and majority logic.

Other algorithms of interest include logistic regression, discriminant analysis, generalized adaptive models, genetic algorithms, simulated annealing algorithms, etc. For research results of the author, in the domains of these algorithms, the interested reader is directed to the web site of the author [3].

The major datamining model (framework for the application of above mentioned algorithms) is the CRISP model which tries to decompose each problem into six different stages, and to apply the relevant algorithms to each stage separately (divide and conquer).

A comparison of 14 different tools is given in [1]. Each tool supports a different algorithm, and their cost (at the time of our research) spans the range of three orders of magnitude, which is a clear indication of the fact that the field is still in its development stages.

An important research issue in this emerging field is how to combine different algorithms, models, and tools, for maximal performance, especially in cases when the meaning of the required knowledge depends on the context.

III. Semantic Web

This section contains a condensed overview. A detailed overview can be found in [2], which is a tutorial. That tutorial can be found on the web site of the author, and was presented many times at conferences, in house for industry, or as a university course, worldwide. Primarily, the issues are stressed which represent either the important bottlenecks or the potential solutions for the general problem of recognition of semantics in cases when information changes the meaning from one context to the other.

The central elements of web today are the information portals responsible for indexing, referencing, and maintenance of data collections. The elements added by semantic web are metadata (S+), and they enable the information portals to be able to do a number of newly added sophisticated functions like interpretation, negotiation, planning, decision making, ratings, trust services, and many other ones. So, semantic web is an extension of the current web that enables computers to be more helpful to the real needs of their users.

The introduction of semantic enables the implementation of a number of qualitatively new concepts and applications on the web, like context awareness (linking based on the meaning of information elements, rather than on the predefined URLs), filtering (visited pages can be rated, which can later on be used for generation of automatic recommendations), annotations (one can add comments to the information on the web, which can be shared by future visitors of the same or related pages), privatization (one can create his/her own database of information from the web).

A layered model of semantic web implies 7 layers. The tower of semantic web is build on foundations consisting of metadata and URIs (Universal Resource Identifiers). The concept of URI is more general than the concept of URL. One URL refers to a specific web page, while one URI may refer to a finer granularity (subset of a web page, or even a single word on a web page). Consequently, semantic coverage can be made more sophisticated!

The major three development strategies of semantic web are: evolution support (building new techniques on the top of the existing ones), minimalist design (making large progress through small steps), and inference (based on the

predicate logic). Such a strategy is enabled by the existence of the concept of the called XML stack.

New vocabularies can be defined with RDF. As indicated before, with RDF one can combine simple metadata (atomic metadata) into more sophisticated metadata (molecular metadata). In this way, one enables that the semantic level of metadata is on the same level as the semantic level of typical user queries. This capability of RDF is enabled with the mechanism called reification. Another mechanism of importance is collections; it enables semantically related knowledge to be grouped, for easier handling.

Ontology is a specification of a conceptualization. Conceptualization is an abstract (simplified) view of the world that we wish to represent for some purpose. In other words, if we need to know only about one aspect of a problem, then all non-related knowledge has to be eliminated; however, without any negative impact on the semantics.

The most popular ontology languages are DAML+OIL or OWL. The OWL Lite is a subset of OWL. In these systems, the body of the ontology consists of classes, properties, and instances. The major component of an ontology is a taxonomy (class hierarchy). The major ontology related problem today is how to treat semantic ambiguities.

IV. Conclusion

This paper gives a comparative overview of datamining and semantic web, and underlines the urgent need for research leading to better concepts and tools for treatment of semantic ambiguities! For a more detailed treatment of these subjects, an interested reader is referred to the references of this paper, or to the proceedings of IPSI conferences [4].

References

In addition to the references listed here, an interested reader can consult also 7 different books, coauthored/coedited by the author of this paper, at his web site. Information form these books was also used in preparing this paper. A common characteristic of these 7 books is that for all of them, a Nobel Laureate wrote a foreword (7 different persons). They are related to IPSI conferences [4].

[1] Jovanovic, N., et al, 'Tutorial on Datamining,' galeb.etf.bg.ac.yu/vm/, May 2004.

[2] Vujovic, I., et al,'Tutorial on Semantic Web,' galeb.etf.bg.ac.yu/vm/, June 2004.

[3] Milutinovic, V., 'Web Site,' galeb.etf.bg.ac.yu/vm/, July 2005.

[4] 'IPSI Conferences Web Site' www.internetconferences.net, August 2005.