Semantic Mining of Web Documents

Svatopluk Fronk, Ivan Jelínek

Abstract: *According to the increasing volume of information available on the Internet, the problem of its effective retrieval became acute during the recent years. A very promising approach is the Semantic Web technology, which allows the information to be interpreted both by humans and by the machines. There are several proposals (e.g. RDF (Resource Description Framework) or OWL (Web Ontology Language)); however, the research in this area is still at its beginning. Although these relatively new and revolutionary languages have many advantages, there are some negatives as well – i.e. lack of personalization possibilities for the access to the information and quantum of information already presented across the web in other formats.*

Key words: *Semantic Web, web access personalization, RDF, OWL.*

INTRODUCTION

The internet is *the* phenomena of the last decade. Almost "every needed data" is already to be found on the Web and the volume of available information is still growing. However, the speed of growth is not always an advantage – a normal user often spends hours to find exactly what he has been requested. Having such an amount of data, it is necessary to find ways how to filter the results of the search effectively and respect the goals of the concrete user – this is a problem of personalized access to the data. Besides the need of personalization, there are new approaches to allow the data to be interpreted and used not just by humans but even by *machines*, the idea of Semantic Web [1]. Further, we will discuss some potential problems concerning the Semantic Web and suggest possible solutions, as well as the techniques of personalized access, which could be eventually used in connection with the Semantic Web.

First of all, the main problem consists of the format the most information is available in. Since the very beginning of the Internet era, the most documents have been presented in the form of HTML documents (over 8 billion documents indexed by Google™ so far). Even thought there have been introduced some more sophisticated formats for data representation since then (including XHTML, XML, RDF [7], OWL [5], OWL-S etc.), the major part still stays in HTML. Before considering the advantages of the new generation of Internet, the Semantic Web, we should keep in mind the enormous amount of knowledge presented in a "traditional" way. To be able to use the data later in the semantic oriented languages, there is a need for the transmission of the old presentation format into the new ones. Finding and developing of such a mechanism should certainly be a subject of the further study [4].

Another important problem concerning the idea of Semantic Web is the absence of personalized access to the information. To gain the flexibility in information retrieval and adaptive behavior in the interaction with users, Web applications should be able to reason about users' goals. This customization would markedly reduce the time needed for finding the relevant information, which, especially during the last years, paradoxically increases [8]. The major challenge is to develop a system which personalizes the service to different users and handles each specific request for data individually. The problems that need to be solved to create such a model of personalized access to the Semantic Web are: identification of user's interest, mechanism for expressing how information should be interpreted based on its context and mechanisms for user preferences [6].

Our research effort there will be design of a *model for semantic classification of HTML documents by set representation* and design of *algorithm providing the personalized access to the data*. We suppose the design of the semantic classification and the personalization of the access to such classified data will be an extension of some of the models described below.

STATE OF ART

In the following, we describe a method of finding the semantics in plain text called *Latent Semantic Indexing* and its possible extensions and three chosen approaches of personalized access to the data. Creating of a connection between these techniques and improving them will be the subject of our future research.

Latent Semantic Indexing

Latent Semantic Indexing (LSI) is called a method based on co-occurence of words across the set of documents [11]. The main idea is to find special words in the document, i.e. those ones which are frequently used in the examined document but rather rare in the most others. These words are most likely to characterize the document's content.

Natural language uses many words which do not carry any semantic meaning – conjunctions, prepositions, auxiliary verbs etc. In fact, these words are the most common ones in written documents[1] too. The first step in the process of LSI is *culling* all these words from a document. After discarding all the articles, prepositions, conjunctions, common verbs (see, do, be...), pronouns and common adjectives (big, small...), we can apply a filter of often used words without any special meaning in the current language, called *stop list*. Also such words which appear in one document only or, on the contrary, appear in all the documents, are not interesting either.

After this first step, there are only those expressions left, which potentially can have some semantic meaning (*content words*). The second step is called *stemming*. The purpose is to prune the set of different expression with the same semantic meaning (inflected verbs, plural nouns,...) to reduce the total amount of relevant words. For example, all the words "inform", "informatics", "information" or "informed" can be reduced to the stem "inform", because they carry almost the same semantic meaning and it does not seem useful to have them listed separately. Counting the occurrences of all the stems in the given document results in an n-dimensional vector ($V_D(n)$) specific for the document D, where n is equal to total count of different stems in the whole document set S.

Having such a vector $V_D(n)$ for every examined document D, we can create a term space $S_n = \sum_S V_D(n)$ containing all the vectors (documents). This space has often many thousands dimensions ($car(S)$). To be able to reason about similarity between documents and discard noise, it is necessary to reduce the amount of dimensions of the term space. The method of such reducing is called singular value decomposition (see [10]). It consists in mapping $\mathrm{M}(S_n, S_m)$ of an n-dimensional space into an m-dimensional one (m < n) by choosing the most appropriate axes and directions of the projection. After such mapping, the size of the term space is significantly reduced without any striking information loss. Even more interesting feature of the singular value decomposition method is that the documents,,which already were similar before this projection, get even closer to each other while those ones, which were different, stay far away from each other in the term space.

After reducing the count of term space's dimensions, we are able to identify the similarity between documents. The closer the vectors of two documents are the more these documents are similar. The last step of LSI algorithm is finding aggregated "clusters" of vectors in the term space. The found sets represent semantically similar documents and can be classified.

[1] For the top 50 most frequent English words see *http://javelina.cet.middlebury.edu/lsa/top_words.htm*

Personalization techniques

Concerning data access personalization there are several independent research groups that seem to be the most perspective ones [6]. We will shortly discuss the approach of the *Torino group* (adaptive behaviour based on action and reaction [3]), the *Malta group* (HyperContext framework for adaptive navigation [9]) and the *Hannover group* (open-corpus adaptive systems [2]).

Adaptation by Reasoning

When searching for a concrete information, the goal of the user and the interaction occurring with the user play a fundamental role. Based on this observation, the Torino group comes with an idea of exploiting reasoning techniques to obtain adaptation. First, the application should identify the goal of the user, so that the whole interaction with the system can be aimed at achieving this goal, no matter whether the user is human or anything else (device, agent, software...).

For adaptivity in the interaction with users and gaining flexibility in problem solution, Web applications should be able to reason about the resource descriptions and users' current goals. Imagine the following example: a student must learn about the Semantic Web. He has access to a repository of educational resources, which does not contain any material annotated as "Semantic Web". However, its information system has a machine-interpretable description of what "Semantic Web" is – the conjunction of keywords: "knowledge representation" and "XML-based languages". Then the system, in case it is able to make inferences over a knowledge-based representation, might answer to the student's query by returning links to documents that explain either something about knowledge representation or about XML-based courses. This result often closely corresponds with the user's expectations, in our example the student obtains the requested data about the Semantic Web.

Furthermore, more complicated forms of reasoning might take place, such as *planning*, which is understood as an automatic construction of a solution, consisting of a sequence of actions that makes a system pass from an initial state to a state of interest (in the previous example it would be a reading sequence for some case where the order of reading documents matters). For planning, an action-based interpretation of the resources is useful. In a similar way, for reading a document some background knowledge might be necessary. In this case we will consider it as the precondition to the action's effect.

The possibility of reasoning about domain knowledge is the starting point of two perspective trends: the *recommendation systems* and the *Web services*. In the first case, the system must be able to foresee the sequence of proposed actions and help the user to achieve his/her goal faster. The Torino group already developed a tutoring system for the cooperation between a teacher and students [3]. The other case, the Semantic Web services, is the idea of cooperating devices scattered over the internet, able to combine results and request in order to solve more sophisticated tasks. This approach is supported in OWL-S language, which, besides the possibility of ontology modeling, enables to see data as objects [5].

HyperContext

HyperContext [9] is a framework for adaptive and adaptable hypertext. A hypertext is a collection of documents and links. An adaptive hypertext is extended by user model, observations and an adaptive component. Each document has zero, one, or more parents. A parent is a document that contains at least one link pointing at it, the document at the destination of the link is the child. Each way of accessing a document is called a *document context*. If a document contains multiple links to the same destination, each link provides a separate document context to the child. Each document contains information related to one or more *concepts*. A visited document contains zero, one, or more concepts in which

the visitor/user is interested. This is all that is observed from the user interaction. An *interpretation* identifies the concepts in a visited document that are relevant to the context in which the document was accessed.

HyperContext [9] is based on identification of user's short-term interest, creating a user model and guiding him to requested information. When the user starts browsing the hyperspace (looking for the information), an empty user model is created. As he moves through the hyperspace, the path of visited documents and links is being created. The technique assumes that the documents visited on the path of traversal are at least partially similar to the user's goal. This is an advantage in comparison to the adaptation by reasoning technique described above, because HyperContext can provide adaptive support even in environments that are not yet semantically described.

After identifying the user's goal, it is possible to recommend him next steps to achieve this goal faster (highlighting links, "see also" links or links even not reachable from the current location etc.).

There is a disadvantage of the HyperContext approach - HyperContext assumes that links already exist between documents in hyperspace. With the existence of these links, it is possible to discover context blocks in the documents. However, if these links do not exist, there is a lack of information about connection between various documents. Extending this method by using Latent Semantic Indexing described above could potentially help to create connections between similar documents explicitly.

Adaptive Functionality in Educational Hypermedia

A logical definition of adaptive educational hypermedia consists of description which kind of processing information is needed from the hypermedia system (the *document space*), the runtime information which is required (*observations*) and the *user model*. *Adaptive functionality* can be described by means of these three components. The aim of such logical definition is to provide a language for describing adaptive functionality, to allow comparison of adaptive functionality in different contexts and systems.

An Adaptive Educational Hypermedia System (ADHS) is a quadruple

$$ADHS = (DOCS, UM, OBS, AC)$$

where

- *DOCS* (Document Space): a finite set of first order logic (FOL) sentences with constants for describing documents and predicates for defining relations between them.
- *UM* (User Model): a finite set of FOL sentences with constants for describing individual users and their characteristics.
- *OBS* (Observations): a finite set of FOL sentences with constants for describing observations and predicates for relating users, documents / topics, and observations.
- *AC* (Adaptation Component): a finite set of FOL sentences with rules for describing adaptive functionality.

Document Space describes the resources belonging to the hypermedia system as well as information associated to these resources. This associated information might be annotations (e.g. metadata attributes); domain graphs that model the document structure or knowledge graphs that describe the knowledge contained in the document collections (e.g. domain ontologies). *User Model* stores and describes information, knowledge, preferences etc. about an individual user (it might share some models with Document space). *Observations* are used for updating the user model. Observations describe the runtime behavior of the system concerning user interactions is contained (whether a user has visited a document, or visited document for some amount of time, etc). Finally, *Adaptation Component* contains the rules for adaptive functionality, rules for adaptive

functionality (e.g. sorting the links leading to further documents according to their usefulness for a concrete user), etc.

All the above described personalization techniques are still problematic [6] in some aspects. In the first case (adaptation by reasoning), there is no proposal of describing the reasoning behavior in Semantic Web yet (even thought OWL-S is the closest). The HyperContext approach is limited by the lack of prior knowledge about the individual user as he starts searching for the information. Yet another problem concerning HyperContext – developing a mechanism of correct interpretation of the document dependent on context of accessing the document is a hard challenge, too. The third proposal, formal description of adaptive hypermedia, is already useful for educational systems. Finding a way to use this approach more generally, especially in field of Semantic Web should be subject of future research as well.

CONCLUSIONS AND FUTURE WORK

Despite of the major volume of information on the Internet presented in HTML format, the recent Semantic Web technologies are based on formats XML, RDF ([7]) or OWL ([5]) and furthermore, the personal preferences of concrete users when accessing information are not yet supported. Our intent is to suggest a *model for semantic searching of HTML documents*, which takes the *context of searched information* and *preferences of the concrete user* into account. The input format of HTML has been chosen because of the prevailing volume of information presented on the internet in this way, and despite of the expected complications, which are most probably going to occur during our intent of the semantic processing of the documents.

The algorithms and techniques described above are promising starting points in the research. Our aim is to develop such a model that, based on analysis of set of different HTML documents, will be capable to identify the *subject matter* of every single one. The model will extend the LSI (Latent Semantic Indexing – [11]), i.e. the frequency of used special words. More clues will be gained from the structure and styles of the document as they were suggested by its author (headings, division into chapters, paragraphs and sections, highlighted keywords, document's title, hyperlinks to other documents…). After such a classification and analysis of thematically similar documents, the next step will be their clustering into hierarchical sets dependently on common topics (keywords) a establishing of connections of different weight between these sets.

During the search itself, the set that matches the query closest will be preferred while sorting the results. Another advantage of this document classification is the possibility to search within the "neighbour" sets, i.e. topic-close. This feature will be certainly appreciated by those users, who are often not able to formulate their search queries exactly.

Besides the thematic similarity of the documents and the search query, the criteria when choosing the most relevant results will be the *preferences of the concrete user*. First of all, the structure and content of his favourite and recently visited web pages will be taken into account. The analysis of the content and style of these pages will provide information about the user's personal preferences and the searching mechanism will be able to prefer the most acceptable pages. The next criteria the user can influence the searching mechanism with, is the application of "filters" – e.g. setting of the technical level or structure of requested document (many examples or theoretical analysis). The application of such a personalization has a great potential in itself (e.g. the user provides an example document about Java programming language and wishes to find documents just as good explaining C#). For this personalization, there will be used and extended some of the already mentioned algorithms.

A successful semantic classification of HTML documents can create a unified structure in the recent rather chaotic distraction of HTML documents on the web and build

a parallel branch of semantic access to data next to the coming formats like RDF or OWL. Such a classification is a starting point for the future transformation of the documents to these new perspective formats. Semantic searching with support of personal preferences can reduce the time needed to find the relevant information in acceptable form, and will be the base for further more sophisticated search algorithms.

REFERENCES

[1] Alesso, H., Craig F. Developing Semantic Web Services. A. K. Peters, Ltd., 2004.
[2] Dolog, P., Henze, N., Nejdl W. Logic-Based Open Hypermedia for the Semantic Web. Hannover (Germany), 2003.
[3] Gena, C., Orfino, M. Engineering the Adaptive Web. Torino (Italy), 2004.
[4] Guha, R.V., McCool, R. Building the Semantic Web. http://tap.stanford.edu/
[5] Harmelen, F., McGuinness, D. L. OWL Web Ontology Language. 2004. http://www.w3.org/TR/owl-features/
[6] Henze, N. Personalization Functionality for the Semantic Web. 2004. http://rewerse.net/deliverables/a3-d1.pdf
[7] Miller, E., Swick, R. Resource Description Framework. http://www.w3.org/RDF/
[8] Search Engine Strategies. http://www.searchenginestrategies.biz/
[9] The HyperContext Framework for Adaptive Hypertext. http://portal.acm.org/ft_gateway.cfm?id=513346&type=pdf
[10] Will, T. Introduction to Singular Value Decomposition. http://www.uwlax.edu/faculty/will/svd/
[11] Yu, C., Cuadrado, J., Ceglowski, M., Payne, J.S. Latent Semantic Indexing. http://javelina.cet.middlebury.edu/lsa/out/lsa_definition.htm

ABOUT THE AUTHORS

Svatopluk Fronk, Ivan Jelínek, Department of Computer Science and Engineering, FEE CTU Prague, Phone: +420 737 569 731, E-mail: fronks1@fel.cvut.cz.