

Helping Courseware Authors to Build Ontologies: the Case of TM4L

Darina DICHEVA and Christo DICHEV
*Winston-Salem State University,
601 M.L.K. Jr. Dr., Winston Salem, N.C. 27110, USA*

Abstract. The authors of topic map-based learning resources face major difficulties in constructing the underlying ontologies. In this paper we propose two approaches to address this problem. The first one is aimed at automatic construction of a “draft” topic map for the authors to start with. It is based on a set of heuristics for extracting semantic information from HTML documents and transforming it into a topic map format. The second one is aimed at providing help to authors during the topic map creating process by mining the Wikipedia knowledge base. It suggests “standard” names for the new topics (paired with URIs), along with lists of related topics in the considered domain. The proposed approaches are implemented in the educational topic maps editor TM4L.

Introduction

The web is changing dramatically the way teachers teach and students learn. For almost any topic now there are online resources that are more current and diverse in perspective than the printed media. For example, computer science master students would find about 70% of their reading on the web. Despite the abundance of online information though, many students fail to get the best out of it: even Google’s smart keyword search provides hundreds of hits to browse. In addition, the web information is not always accurate, credible or reliable. For e-learning this implies a need of repositories, complimentary to web, with trusted information, organized appropriately for easy access by learners. Ontology-driven learning repositories, supported by intuitive interface and efficient navigation and search capability for easy exploration, come as a promising solution.

Ontology-driven learning repositories need authoring tools that support both the ontology creation and semantic annotation of resources. Currently available general purpose ontology languages and tools though tend to impose high entrance barriers for end users. They typically assume that authors are able to conceptualize domains of interest in terms of concepts, concept types, and relationships between concepts. However, courseware authors are largely used to organizing learning content in terms of modules, chapters and subchapters (i.e. of “concept containers”), instead of concepts [1]. The task of providing a complete and systematic list of all concepts that should be covered by a learning unit is not trivial. Additional “ontological challenges” include using widely agreed names for the concepts and inter-relating them with meaningful relationships. So, authors do need assistance in conceptualizing subject domains.

Topic Maps for e-Learning (TM4L) [2] is an environment for creation and use of topic-focused, ontology-driven learning repositories, based on the Semantic Web

technology Topic Maps (TM) [3]. In the Topic Map paradigm, an ontology is an accurate description of the essential entities and relations in the modeled domain and can be represented as a set of topics linked by associations.

Compared to the manual topic map authoring offered by TM4L, an automatic extraction of topical structures seems very appealing. However, at present there is no technology available to provide automatic topic, relationship, and resource extraction in conceptually clear, intuitive, accurate, and scalable fashion. Indeed, fully automatic topic map construction is not feasible yet, especially in the education domain, where many subject and resource specific decisions must be made to adequately specify a learning collection. Thus our goal was to extend TM4L functionality, so as to support authors with limited knowledge of IT and ontologies as much as possible.

The proposed support consists of automatic extraction of topic map constructs (concepts and relationships) from specified web pages and using them to build a “draft” topic map for the author to start with. We have analyzed and compared different text extraction approaches (including machine learning and natural language processing) and have chosen to extract topics and relationships by crawling a website and using the semantic information that can be extracted from the HTML markup. Our idea was inspired from a study on the feasibility of using three HTML tags as proxies for semantic content: link text, heading, and comment [4]. In that study, human experts have analyzed a large number of web pages and have found that 61% of the link text was “helpful” or “very helpful” in indicating semantic content. This seemed quite encouraging since the web pages have been arbitrarily chosen.

We designed and implemented a plug-in for TM4L including two options for automatic extraction of topics and relationships: from a website specified by the author and from the Wikipedia and Wikibooks websites.

1. Topic Map Object Extraction from a Website

This aspect of our support for TM authors was motivated by the fact that there is a significant amount of semi-structured information on the web. HTML documents, for instance, are structured for rendering purposes but their structure can also be used for extracting some semantic information. For example, list items can be transformed into members of “whole-part” relationships; items marked up as “bold” or “italic” can be considered semantically important, etc. Thus our goal was to find out what semantic information can be extracted from the HTML markup of web pages and included in the intended ‘draft’ topic map. Since there are no formal rules for extracting semantic information, heuristic approaches are practically feasible. From this viewpoint we can set some heuristics for HTML tables, such as “A row represents a topic, a column represents an occurrence”. Such observations, combined with experiments with various types of information in HTML format led us to propose the following *heuristic rules*.

1.1. Defining ‘Page’ Topics

A ‘draft’ topic map consists of topics and relationships. These objects are extracted by crawling a specified web site. We differentiate between two types of topics: topics that reflect the web site topology and topics extracted from the text of the web pages.

Rule 1: For each web page visited by the crawler, a new topic is created in the topic map. We call these topics “page” topics.

Rule 2: All the topics created in the process of parsing a specific web page are sub-topics of the “page” topic for that page.

Rule 3: Naming of “page” topics: The theme of interest, provided by the user, is used as a name of the “page” topic, corresponding to the entry page for crawling the site; all other “page” topics are named using the text in the corresponding “anchors”.

1.2. Information Extraction from Heading Tags, List Element Tags, and Table Tags

Rule 4: Heading represents a topic that is more general than the topics extracted from the text below it (if any).

Rule 5: The topics extracted from headings of different levels can be organized in a taxonomy reflecting headings’ level (1, 2, etc.).

Rule 6: Heading tags on a referenced (through an ‘anchor’ element) web page are considered as structurally related to the “page” topic of the referencing page. Thus the result of parsing heading tags will consist of a set of topics named by the text enclosed in the heading elements and connected to the “page” topic of the referencing page.

Rule 7: The topics extracted from list item tags are more detailed (sub-topics) of the topics contained in the list tags. The list-item relationship between any two topics is modeled by a “child-parent”-type relationship. Table of contents expressed as a bulleted list in HTML has an isomorphic image in terms of a “whole-part” tree in TM.

Rule 8: The topics extracted from the cells of a table column are related, since they can be considered as values of the same attribute (represented by the column header).

Rule 9: The topics extracted from the cells of one column in a table are subtopics of the topic corresponding to the column header.

The difficulties in extracting relations from text are largely recognized, but we can try capturing at least the relevancy of topics that appear in the same HTML elements:

Rule 10: Group the topics extracted from the same HTML element together, since this grouping indicates some kind of relatedness of the topics.

2. Topic Map Object Extraction from the Wikipedia and Wikibooks Websites

Since names are the primary mechanism for denoting subjects in human/machine discourse, we need sharable topic names. However, deciding, from one side, what concepts (topics) to include in an ontology and, from another, which are the widely agreed names for the selected concepts is a big challenge for the topic maps authors. We address this dual challenge by proposing the use of Wikipedia and Wikibooks as information sources in the automatic creation of draft topic maps. Manual topic extraction is not feasible since it is not easy from a given Wikipedia article to locate all topics related to it.

2.1. Why Wikipedia?

The underlying structure of a concept-based learning repository usually can not be derived from a single textbook or course syllabus. Besides the fact that textbooks and courses are frequently changed, their structures are often inconsistent. In addition, their titles and section names are generally arbitrary. If we aim at a reusable model, it should be founded on a more stable structure [1]. We can increase the level of consensus on a domain description by using Wikipedia as a provider of concepts (a vocabulary) for

defining topics or expressing queries (see Fig 1). So, the idea is to use the Wikipedia/Wikibooks corpus as a rich source of common sense conceptualization. This comes also in support of the consideration that ontologies are not just formal representations of a domain, but much more community contracts [5] about such formal representations. Wikipedia is popular as a reference and its concepts can be expected to have commitment by a wide audience. The authors of [5] have shown that the URIs of Wikipedia entries are surprisingly reliable identifiers for ontology concepts. The English version of Wikipedia contains more than 850,000 entries, which means it holds unique identifiers for 850,000 concepts. Because many people are contributing to Wikipedia, we believe that the terminology and naming convention there are more “standard” (agreed upon) than in other sources. Thus, Wikipedia can be used as a source of reliable and sharable concept names.

2.2. How to use it?

The TM4L extraction tool uses Wikipedia in the following way:

- As a source for proposing concepts relevant to a particular subject.
- As a source of “standard” concept (topic) names.
- As a source of URIs to be used as Public Subject Identifiers (PSI) [3] for concepts.

The duality of our Wikipedia use as a source of topics and identifiers reflects the human/computer dichotomy: topics are for the benefit of human users; PSIs are for the machines that need to ascertain whether two pieces of information are about the same subject. Technically, for a given sequence of keywords, e.g., “Operating Systems”, a topic is selected from Wikipedia and Wikibooks that is the best match to the intended concept along with a set of related to it topics (see Fig. 1). Both sites are searched separately and the resulting topic collections are merged. In general, Wikibooks provides a better “table of contents” than Wikipedia (more useful for conceptual structuring of a course) however some topics are not covered there, since its content is more limited. The combined search in both sites provides a better coverage.

3. Implementation and Evaluation

3.1. Implementation issues

The TM4L extraction plug-in is written in Java and consists of Extraction Engine, Topic Map Builder, and User Interface. It uses WebSphinx and Java API for XML Processing (JAXP) that supports Document Object Model (DOM) API to process XHTML documents.

The heuristic rules proposed in Section 1 are used for parsing a specified unknown website. For choosing the Wikipedia/Wikibooks web pages to be processed, we use the Wikipedia search engine that returns Wikipedia links ordered by their relevance to the specified topic. The problem of using directly the standard URLs of Wikipedia articles is that an article corresponding to a required topic may not exist or a different type of addressing may be used. Our solution helps also in the case of misspelled words: the search engine returns the links in order corresponding to their relevance score and the highest ranked is taken for processing. For the Wikipedia/Wikibooks article search we

identified mark-up patterns that are applied consistently across all articles. For example, the majority of articles provide a table of content (TOC) to support better page navigation. Although, Wikipedia and Wikibooks apply a variety of page structuring patterns, we identified rules defining a TOC. When the algorithm cannot identify a TOC, it performs a search for section titles and links.

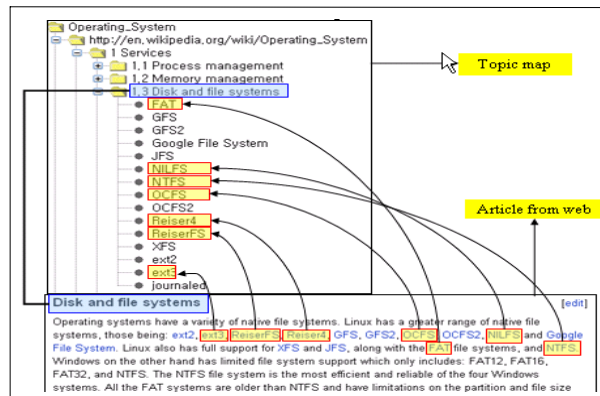


Fig 1. Topic hierarchy extraction from Wikipedia

The output of the extractor in both cases (from a specified unknown web page or Wikipedia) is a hierarchical structure of topics (Fig. 2). In Wikipedia, the level of nesting for a particular topic corresponds to its location in the page section-subsection structure. For example, “FAT” was found in “Disk and file systems”, which in turn is located in “Services” (see Fig. 1). The user can then select some of the proposed topics to be merged with his or her topic map.

3.2. Measuring the Effectiveness of the Extraction

To test our topic extraction algorithm we applied it to 35 web pages with different patterns of topic groupings. To illustrate the evaluation process we present here the evaluation results of applying the algorithm to two representative web pages: a topic specific page - Wikipedia Prolog (<http://en.wikipedia.org/wiki/Prolog>) and a general information page of the Math Forum Internet Mathematics Library (<http://mathforum.org/library/toc.html>). The assessment of the performance of the crawler and the proposed rules are based on the standard measures from Information Retrieval, *Recall* and *Precision*. For our task recall is interpreted as the number of relevant topics returned in a particular grouping, divided by the total number of relevant topics in that grouping. Precision is interpreted as the returned relevant topics, divided by the total number of topics returned by the crawler using the proposed heuristics. The evaluation results in terms of recall and precision are shown in Table 1 in Table 2.

A similar evaluation was performed for the information extraction from Wikipedia. Five instructors were asked to create topic maps with TM4L, each with 10 topics, based on their course syllabus. For each topic map, they had to report the number of all topics returned from Wikipedia, the number of relevant topics among them, and the total number of relevant topics in Wikipedia. A topic was considered relevant if the corresponding concept was covered in the course. The precision and recall values were then computed. (The tables are not presented here due to lack of space.)

Table 1. Recall values for the two websites

Web page	HTML Element	# Relevant Topics Returned	Total # Relevant Topics	Recall Value
Wikipedia/Prolog	Lists	50	50	1
	Tables	12	12	1
	anchors	24	25	0.96
Math Forum	Lists	134	134	1
	Tables	10	22	0.45
	anchors	134	156	0.86

Table 2. Precision values for the two websites

Web page	# Returned Relevant Topics	Total # Returned Topics	Precision Value
Wikipedia/Prolog	1055	1863	0.57
Math Forum	278	380	0.73

3.3. Discussion

Our preliminary assessments demonstrated an acceptable accuracy for the rules. Interpreting the topic extracted from an anchor element as the root for all topics extracted from the referenced page, generally resulted in a natural hierarchical structuring of the topics in the extracted tree and thus in a reasonable level of precision and recall. The text extracted from the anchor elements in most cases was appropriate for naming the “page topics”. Heading tags allowed the extraction of topics related indeed by a “child-parent” relationship to the root topic. In a similar way, concepts extracted from headings were linked by a taxonomic relation to the concepts extracted from the subordinate sections. Topic extraction from HTML lists produced the best results in terms of precision and recall. The hierarchical topic structure captured from lists (see Fig. 2) reflects the nested nature of the corresponding list structure thus preserving the intended relationship between the list items in the document.

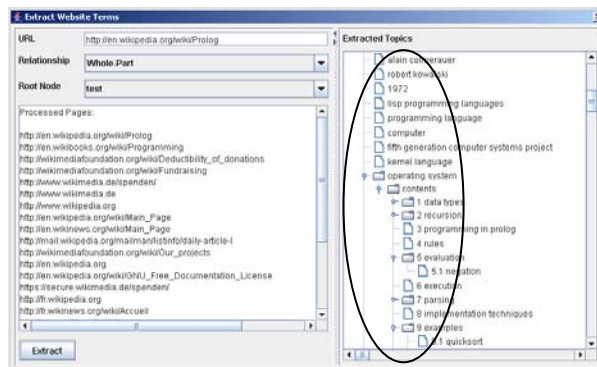


Figure 2. Results screen with list tags extraction

The evaluation results are encouraging since they were obtained without considering the impact of the user factor. Since the user selects the web site for

crawling, it is reasonable to expect that he or she will choose pages containing some sort of classification scheme. In such cases the recall and precision of the extracted topic structures will be comparable (if not better) to our results. We found out that our recall and precision results were not significantly different from the manually processed results reported in [4]. This is in favor of our algorithm since it extracts automatically a larger set of concepts while preserving the same rate of relevance. The precision value could be improved further by suggesting more heuristics.

4. Related Work and Conclusion

A vast number of methods have been proposed for text extraction, including text mining and NLP approaches. The novelty of the unstructured or semi-structured and highly noisy hypertext data, however, in many cases diminishes the efficiency of these classical methods. Among the most used are the clustering (unsupervised learning) and classification (supervised learning) methods. A recognized complication with clustering in the hypertext domain is the lack of agreement about what the different people expect a clustering algorithm should output for a given data set. This is partly a result of the difficulty to guess what their implicitly used similarity measures are, due to the typically very large number of attributes. The classic supervised learning methods (such as [6]) are not suitable for our goal, since they require initial training with a corpus of documents that are previously labeled with topics. For the same reason, the classical NLP approaches are also unsuitable. Language analysis should be specialized in a specific topical area. For example, [7] describes an NLP approach for extracting text to build a table of contents of pages on the web for specific concepts. The approach searches for definitions and subtopics of a given topic and preprocesses the extracted data using predefined verbs.

The OntoLT approach [8], used in a plug-in for the ontology editor Protégé, proposes automatic extraction of concepts and relations from annotated text collections. The annotation allows for automatic extraction of linguistic entities according to predefined conditions and using them to construct a shallow ontology of domain concepts and relations. This approach is not suitable for us, since it requires preliminary annotation of the text collection. Fortuna et al. describe a system for a semi-automatic ontology extraction from a document collection, using text mining algorithms including Latent Semantic Indexing (LSI), K-Means clustering, and keyword extraction to build their tool, which suggests to the user possible new topics [9]. When the user selects a topic (defined as a set of related documents), the system suggests subtopics of that topic, by applying LSI to the documents from the selected topic. While their goal of ontology extraction is close to ours, they do use the created so far ontology, while we aim at an initial creation of a topic map.

The idea of exploiting Wikipedia semantics has been in the focus of several studies in the last years. The authors of [10] discuss semantic relationships in Wikipedia and the use of link types for search and reasoning. Recent research has also shown that Wikipedia can be successfully employed for NLP tasks, e.g. question answering [11] or text classification [12]. However, we are not aware of any research for using Wikipedia semantics in educational systems.

In the educational computing field, Kay and colleagues have been working on extracting ontologies from dictionaries, see for example [13], Henze and colleagues

focus in [14] on metadata generation from documents using context, while the authors of [15] utilize resource description formats for generating hypermedia structures.

The main difference between the above approaches and our approach is in the intended users of the extracted information: in our case it is intended for *human consumption*. The novelty of our approach is the collaborative building of the ontology by the author and the “helper agent”. The agent extracts on demand topical structures that the author evaluates and possibly includes in the ontology. In this scenario the user interface and the presentation of the extracted topical structure are of equal importance compared to the accuracy of concept extraction. All this sets a new research perspective in the area of intelligent educational systems.

Various approaches, as described here, are applicable to the TM authoring support. At one pole are shallow techniques for extracting rather weak structural information while at the other extreme are deep statistical methods or knowledge driven parsers able to build rich semantic structures. We have shown that the proposed heuristics enable extracting structured information from HTML documents with usable degree of accuracy. We also succeeded to make use of Wikipedia without deep language understanding. This was made possible by applying standard techniques to match the structures with relevant properties. Empirical evaluation confirmed the value of encyclopedic knowledge for indexing of topics. The heuristics based on syntax alone, however, are not enough to capture the mass of structured information on the web. We are exploring the use of domain knowledge, through ontologies, for better topic extraction algorithms.

References

- [1] Dicheva D., Dichev C.: Authoring Educational Topic Maps: Can We Make It Easier? 5th IEEE Intl Conf on Advanced Learning Technologies, July 5-8, 2005, Kaohsiung, Taiwan, (2005) 216-219
- [2] Dicheva, D. & Dichev, C.: TM4L: Creating and Browsing Educational Topic Maps, British Journal of Educational Technology - BJET (2006) 37(3): 391-404
- [3] ISO/IEC 13250:2000 Topic Maps, www.y12.doe.gov/sgml/sc34/document/0129.pdf
- [4] Hodgson, J.: Do HTML Tags Flag Semantic Content?. IEEE Internet Computing, 5 (2001) 25
- [5] Hepp M., D. Bachlechner, and K. Siorpaes. Harvesting Wiki Consensus - Using Wikipedia Entries as Ontology Elements. 1st Workshop Semantic Wiki 2006 at ESWC 2006, Budva, Montenegro (2006)
- [6] Vargas-Vera, M., Motta, E., Domingue, J., Shum, S., Lanzoni, M.: Knowledge Extraction by using an Ontology-based Annotation Tool: kmi.open.ac.uk/projects/akt/publication-pdf/vargas-saw.pdf, 2000.
- [7] Liu, B., Chin, C.W. and Ng, H. T.: Mining Topic-Specific Concepts and Definitions on the Web. Proc of WWW 2003, www2003.org/cdrom/papers/refereed/p646/p646-liu-XHTML/p646-liu.html (2003)
- [8] Buitelaar, P., Olejnik, D. and Sintek, M.: A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. 1st European Semantic Web Symp.. <http://dfki.de/~paulb/esws04.pdf> (2004)
- [9] Fortuna, B., Mladinic, D., Grobelnik, M: System for Semi-Automatic Ontology Construction. Proc. of 3rd European Semantic Web Symp.. <http://www.eswc2006.org/demo-papers/FD18-Fortuna.pdf> (2006)
- [10] Krotzsch M, Vrandečić D., and Volkel M. Wikipedia and the Semantic Web -The Missing Links. In Proc. of Wikimania 2005, First International Wikimedia Conference, Wikimedia Foundation, 2005.
- [11] Ahn D, Jijkoun V, Mishne G, Müller K, Rijke M, Schlobach S. Using Wikipedia at the TREC QA Track. In Proceedings of TREC 2004.
- [12] Gabrilovich E., Markovitch S. Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. Proc of AAAI 06, pp. 1301-1306, Boston, MA 2006.
- [13] Apted, T., Kay, J., Lum, A. Supporting metadata creation with an ontology built from an extensible dictionary. 3rd Int. Conf. on Adaptive Hypermedia and Web-based Systems, Springer, (2004) 4-13.
- [14] Henze, N., Dolog, P., & Nejdil, W., “Reasoning and Ontologies for Personalized E-Learning in the Semantic Web,” Educational Technology & Society, 7 (4), (2004) 82-97
- [15] Cardinales, K., Meire, M., Duval, E.: Automated Metadata Generation: the Simple Indexing Interface, Int. WWW Conference, Chiba, Japan, (2005)