

Automatically building domain model in hypermedia applications¹

Hermine Njike, Thierry Artières, Patrick Gallinari, Julien Blanchard, Guillaume Letellier

LIP6, Université Paris 6
8 rue du capitaine Scott, 75015, Paris, France
{Firstname.Lastname}@lip6.fr

Abstract. This paper deals with the automatic building of personalized hypermedia. We build upon ideas developed for educational hypermedia. A standard way to build adaptive educational hypermedia relies on the definition of a domain model and the use of overlay user models. Since much work has been done on learning user models and adapting hypermedia based on such user models, the core problem lies in the automatic definition of a domain model for a static hypermedia. We describe an approach to automatically learn from the hypermedia content such a domain model. This model is a concept hierarchy where concepts are identified by sets of keywords learned from the collection. We propose the use of visualization techniques such as treemaps in order to monitor and analyze efficiently user and domain models.

1. Introduction

Adaptive hypermedia aim at offering personalized hypermedia and websites to a user. It relies on user models that consist in static and dynamic information such as goals, preferences... A domain model may be used that characterizes the whole knowledge accessible in the hypermedia it is used to infer information in the user model [2, 6]. Many works have been done in adaptive hypermedia that one can distinguish according to the nature of the task. Maybe the most well defined problem concerns educational hypermedia and tutorial systems [5, 6, 7]. Although building such systems is still difficult, the task is indeed well identified; in such systems domain models are often manually designed and defined as a set or a graph of the concepts being discussed in the hypermedia. Overlay user models share the same representation as domain models and are used to represent a user knowledge and/or interest in the concept space [5, 6, 9]. These user models are vectors of attributes (e.g. interest) one for each concept in the domain model. These are updated from user navigation logs according to the domain model, a popular way to make inference in these models is to use Bayesian Nets [5, 8, 18] since these models allow taking into account relationship between concepts, inferring and propagating information in nodes. A more difficult task that has been less studied up to now, concerns adaptive systems for any single

¹ This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

website or hypermedia [1, 13, 18]; the domain model, that is the set of topics or concepts, is often wider and much less explicit, so that the task is much harder. In [18] a very simple approach has been proposed where domain model is derived straightforwardly from the website structure, the UM 2001 conference website. Concepts of the domain model correspond to pages in the site (with nodes corresponding to *Paper Submission*, *Call For Papers*, etc) and are organized as a hierarchy derived from the website structure. Unfortunately, viewing the domain model as a clone of the site structure may fail when the structure is weak or not so much related with the underlying concepts discussed in the pages of the hypermedia.

We are interested in this paper in developing techniques allowing the automatic building of personalized hypermedia. To do this, we believe that one can take advantage of works done in the educational hypermedia field concerning the learning of overlay user models and the personalization of hypermedia based on overlay user models. In this context, the core problem for the automatic conception of a personalized hypermedia lies in the automatic learning of a relevant domain model. This is not however an easy task. This model involves high level concepts that cannot be easily inferred automatically. We present an approach that allows learning automatically a concept hierarchy from a corpus of documents (e.g. pages of a website). Concepts in this hierarchy are organized according to a generalization / specialization relation and document subsets may be associated to each concept. Such a representation of the hypermedia thematic content allows defining relevant overlay user models. Also, visualization of this representation using treemaps provides an alternative view of the hypermedia by reorganizing its content according to the learned hierarchy. This may be used for easy monitoring of user models or for users to browse this new interface. We first describe our approach, then we discuss how this approach may be used in the context of user modeling for learning automatically domain models.

2. Discovering concepts from a collection of pages

We study now how to learn automatically concept hierarchies from collections of documents (e.g. the pages of a website) according to a generalization/specialization relation. Our aim is to build generic tools to extract simple semantic relations between corpus elements, which can be used to build domain and user models. Our method starts by automatically learning concepts from a corpus, and then learns generalization/specialization relations between these concepts.

Several approaches were developed in information retrieval for the generation of hierarchies. Clustering techniques have often been used to create document hierarchies. However, there is no semantic relation between the nodes at different levels in these hierarchies. As a consequence, these works are practically useless regarding our goal. Recently new types of hierarchies which are automatically built from corpora have been proposed [10, 11, 16]. These are term hierarchies built from generalization/specialization relations automatically discovered between terms in a corpus. Once this term hierarchy is built, it is possible "to project" documents on it, thus producing a document hierarchy. We propose to extend these approaches to the discovery of a concept hierarchy where concepts, which are discovered from the corpus, are represented as sets of keywords and not by single terms. Such a representation allows for

a richer description than single terms, thus better reflects the different ideas which appear in documents. We detail in the following the main steps of the procedure. For clarity of presentation, we consider in the following that a hypermedia is decomposed in units (e.g. pages of a website), that we will call documents. The corpus, a set of documents, is first preprocessed and segmented into homogeneous paragraphs. The segmentation task consists in identifying, in each document, homogeneous text regions or frontiers corresponding to topic shifts between such regions. Next, all these paragraphs are clustered in order to determine groups of paragraphs related to a similar topic. Each discovered topic is considered then to be a concept of the collection. A by-product of this step is that each cluster (i.e. a concept) is represented as a set of words. Based on a set of concepts, a document may be classified according to the concepts it addresses. Finally, specialization/generalization links are discovered between concepts using a subsumption measure between concepts.

2.1. Pre-processing and document representation

The system's input is a set of documents. All documents are preprocessed as usual in information retrieval tasks; non informative words are removed, all remaining words are lemmatized. Let $V = \{w_j\}_{j \in \{1, \dots, M\}}$ be the vocabulary of M lemmatized words, $D = \{d_i\}_{i \in \{1, \dots, N\}}$ be the set of documents in the collection (after preprocessing), and $P = \{p_k\}_{k \in \{1, \dots, L\}}$ the set of paragraphs of documents in D . Representations of documents and paragraphs are M dimensional vectors. A document d_i is represented as a vector of weighted frequencies (tfidf) for terms in V , $d_i = (tf_i(w_1)idf(w_1), \dots, tf_i(w_M)idf(w_M))$ where $tf_i(w_j)$ is the frequency of term j in D_i and $idf(w_j) = \log(N/df(w_j))$, where $df(w_j)$ is the number of documents in D containing term w_j . Similarly, a paragraph p_k is represented as a vector: $p_k = (tf_k(w_1)ipf(w_1), \dots, tf_k(w_M)ipf(w_M))$ where $tf_k(w_j)$ is the frequency of term j in p_k and $ipf(w_j) = \log(L/dp(w_j))$, with $dp(w_j)$ the number of paragraphs containing w_j . The similarity measure between two entities (documents or paragraphs) is the classical cosine between their vector representations used in information retrieval.

2.2. Segmentation step

The segmentation task consists in identifying in a document (i.e. a page), homogeneous text regions or frontiers corresponding to topic shifts between such regions. We used the technique proposed in [15]. This method proceeds by decomposing texts into segments and topics, a segment being a bloc of contiguous text about one subject and a topic being a set of such segments. Here is the sketch of the algorithm, which starts at the paragraph level (Paragraphs are the basic text unit) since authors generally expose one point of view per paragraph. For each document, repeat until convergence:

- Compute similarities between all paragraphs in a document and keep those higher than a given threshold.
- Build a similarity graph and extract triangles. A triangle is a set of three paragraphs with strong similarities, i.e. susceptible to represent a coherent topic.

- For each triangle, build its vector representation which is the average of the three vectors representing the paragraphs of the triangle.
- Merge the triangles whose similarity is higher than a given threshold.

This procedure is used for any document in D .

2.3. Clustering topics

Once each document is decomposed into a set of topics, we cluster these topics in order to identify a representative set of concepts for the corpus:

- Build a graph based on similarities between topics identified above using the method by Salton (1996) (i.e. there is an edge between two topics if the similarity is higher than a given threshold).
- Compute the connected components of this graph. For each component, keep only nodes which are connected to at least 75% of the other nodes of the component.
- A component with at least $\beta\%$ of its documents (β has been fixed around 90% in our experiments) in a second component will be merged with the latter.

At last, each remaining component is considered as a concept of the corpus. The concept representation is a set of most significant keywords (e.g. with highest *tfidf* measures). From now on we will identify “concepts” and their sets of keywords.

2.4. Inferring « generalization/specialization » relations between concepts

One main idea of our method is to infer generalization/specialization relations between concepts that are identified by sets of keywords. Quite generally, there exists a “generalization/specialization” relation between entities $C1$ and $C2$ if $C2$ evokes a specificity of $C1$, or is about specific themes of $C1$. For example $C1 = \text{sport}$ and $C2 = \text{football}$. Most document hierarchies make use of simple concept representations where a concept is identified with a single keyword. For such a representation, Sanderson (1999) proposed a method for automatically inferring term hierarchies by learning a generalization/ specialization relation between terms; it is based on term subsumption. The idea is that some terms which occur frequently in a collection give significant information about the concepts discussed in the corpus. These terms may define a subject in a general way, whereas others which co-occur with these general terms and are less frequent explain some aspects of the subject. The subsumption measure characterizes a relation of generality/specificity between two terms and is based on asymmetrical terms co-occurrences. It is defined as follows: Term x subsumes (i.e. is more general than) term y if $P(x/y) > th$ and $P(y/x) < P(x/y)$, where th is a threshold. This means x subsumes y if documents in which y occurs are a subset or nearly a subset of the documents in which x occurs. The second rule ($P(y/x) < P(x/y)$) ensures that if both terms occur together more than $t\%$ of the time, the most frequent term will be chosen as the more general. Probabilities $P(x/y)$ may be approximated through counting $P(x/y) = n(x,y) / n(y)$ where $n(x,y)$ is the number of documents that contain terms x and y , and $n(y)$ is the number of documents that contain term y .

Now recall that a result of the previous step is that each concept is identified by a set of keywords. We extended the term subsumption measure described above to concept subsumption, where each concept is represented by a set of keywords. The method consists in computing conditional probabilities $P(Ci/Cj)$, the probability that a docu-

ment discussing of concept C_i discusses also of concept C_j . Estimating such probabilities for any pair of concepts allows applying the subsumption definition directly to the concepts. Once the relations of “generalization / specialization” are detected on pairs of concepts, we apply transitivity to build the concept hierarchy.

The main problem is to compute probabilities $P(C_i/C_j)$. It could be estimated with $P(C_i/C_j) = n(C_i, C_j)/n(C_i)$ where $n(C_i, C_j)$ stands for the number of documents dealing with concepts C_i and C_j and $n(C_i)$ stands for the number of documents dealing with concept C_i . This estimation is rather poor. Another way is to approximate posterior probabilities $P(C/d)$, that a document d discusses concept C or not, which is not easy. At this point, the result of the document segmentation step could be used to assign concepts to the documents. If a paragraph in document d belongs to concept C then $P(C/d)$ is non zero. It could be set to a real value, by measuring e.g. the importance of the paragraph in the document. However, this provides a crude estimation of $P(C/d)$. In our system, we propose to estimate $P(C/d)$ via an Estimation / Maximization (EM) algorithm. This algorithm iteratively computes probabilities $P(t/C)$ for all concept C and vocabulary term t , through maximizing the likelihood of the document collection. Assuming a naïve Bayes model for documents, it allows computing $P(d/C)$ and therefore $P(C/d)$ via Bayes rule. It aims at maximizing training data log likelihood:

$$\text{Log}(L) = \text{Log}(P(D/\Theta)) = \sum_d \sum_{t \in d} \log(\sum_C P(t/C, \Theta) P(C/\Theta))$$

where Θ stands for the model parameters. The EM algorithm alternates estimation of hidden variables $P(C/d)$ and reestimation of model parameter $P(t/C)$. At each step, documents are considered to discuss of concept C if $P(C/d)$ is over a threshold.

Note that with this definition, a concept may have several parents: This corresponds to different meanings of this concept and reflect its polysemia. Also, an important remark concerning this subsumption measure between concepts is that it is suitable in domains where terms are often repeated. If this was not the case, the co-occurrence estimations would not be robust enough to be relevant. However, one could reduce the sensitivity of the technique to corpus variability by using linguistic resources like WordNet to take into account synonymy.

3. Discovering domain model in hypermedia

We applied our approach to the discovery of a domain model, i.e. a concept hierarchy, of a collection of documents, which is a part of the www.looksmart.com site hierarchies. One interest of this corpus is that we can compare, after learning, the discovered hierarchy and the manually designed one. Quantitative evaluation criteria may be defined for estimating generalization / specialization expressiveness of a hierarchy [12], we will mainly show visually our results here since it is more related to our goal. The corpus consists in about 100 documents and 7000 terms about artificial intelligence and is a homogeneous set of documents. This collection has been manually organized in hierarchies of themes. We extracted a heterogeneous sub-hierarchy from this site with documents about different topics. We ran the method on the flat corpus, without any use of the hierarchical information. Compared to the initial Looksmart hierarchy with five categories, the hierarchy derived by our algorithm is much larger and deeper. Most of the original categories are refined by our algorithm. For example, many sub-categories emerge from the original “Knowledge Representation” category (see Figure 1): ontologies, building ontologies, KDD... and most of the emerging

categories are themselves specialized. In the same way, “Philosophy-Morality” is subdivided in many categories like AI definition, Method and stakes, risks ... It is clear that such a result could not have been obtained using single keyword concepts.

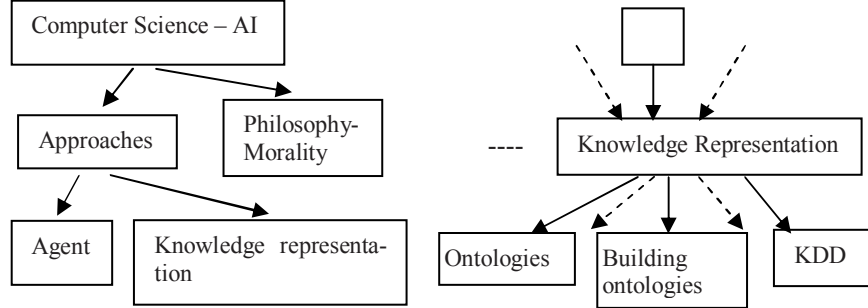


Fig. 1. Sub-hierarchy of the LookSmart corpus used in our experiments (left) and part of the deeper hierarchy discovered using our approach (right).

An interesting feature of this hierarchical organization is that it allows using visualization tools. We considered the use of Treemaps that have been introduced by [17]. The idea of treemaps is to display a tree-like structure in a 2D space where each node is represented by a rectangle whose size or color is determined by a value, it could be the user interest in a concept in our case. Fig. 2 shows a Treemap representing the Looksmart domain model. Each concept is shown as a rectangle with different colour, the hierarchy is shown through inclusion of rectangles. Set of keywords associated to intermediate concepts are also shown.

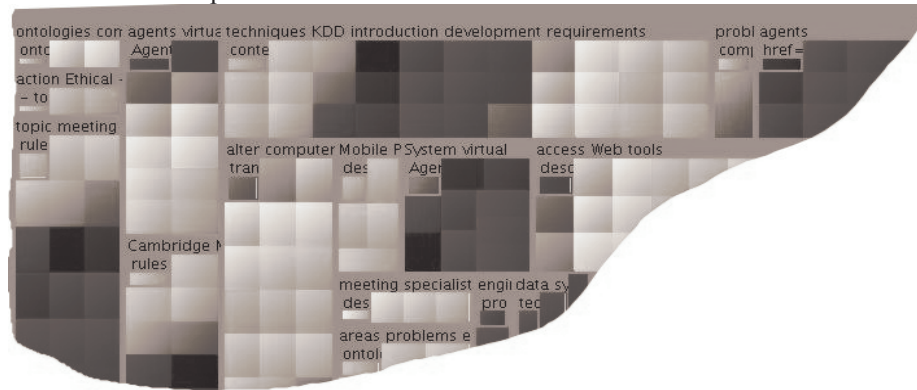


Fig. 2. Part of the Treemap for the Looksmart corpus. Each colored rectangle stands for a concept. Sets of words associated to intermediate concepts only are shown.

4. Using discovered domain model for user modeling

Our method may be applied to build domain models for a hypermedia or website. Once a domain model is learned, user models may be defined as overlay models,

sharing the same representation as the domain model. Standard techniques may then be used to learn and update these user models, including Bayesian Nets as proposed in [5, 7]. We realized an experiment by running the method on the collection of the pages of the website of a French museum. Fig. 3 shows the resulting domain model as a treemap, with french keywords (paleontology, sea, press, biodiversity etc).

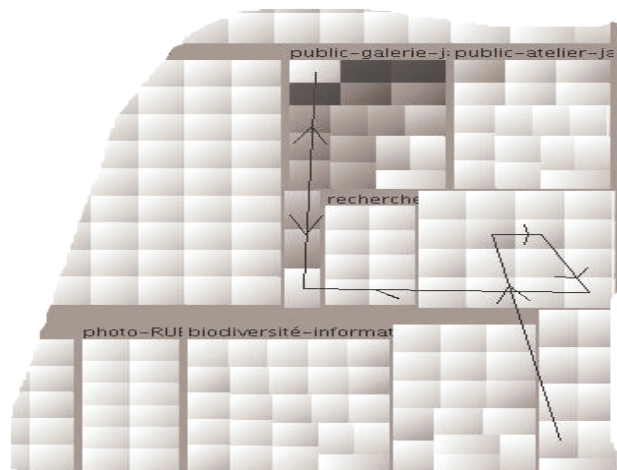


Fig. 3. Part of the Treemap for the website of a French museum where a navigation path has been drawn. The color of rectangles is a function of the similarity between the concepts of the 3 last visited pages and the concepts in the domain model.

To show how such a user model may be used, we have shown a navigation path of a particular user on this treemap, and have defined the colour of a concept (a rectangle) to be a function of the thematic similarity of concepts with the three last pages visited by the user (computed through cosine measure). As may be seen, concepts that are close to the pages recently visited by the user stand close to the current concept. Other information could be visualized. Indeed, treemaps allows redefining easily the rectangles colour and size. Hence, one can browse and investigate a user model by assigning a *knowledge* or *interest* information to the size or colour of the rectangles. This kind of visualization allows having global and synthetic information about a user.

5. Conclusion

We described an approach to automatically learn a domain model from a corpus of hypermedia documents. This approach may be used for instance on the collection of pages of a web site in order to automatically learn a adequate domain model. Based on such a domain model, one can define user models as overlay models. The interest of this approach lies in existing works showing how to learn such user models from logs, and how to perform hypermedia adaptation based on such user models. We also show how efficient visualization techniques such as treemaps may be used to visualize and analyze synthetically both the domain and the user models.

Acknowledgment: The authors would like to thank J.D. Fekete from LRI (Université Paris Sud, France) for helpful discussion about visualization tools and treemaps.

6. References

1. Alfonseca E., Rodriguez P., Modelling users' interests and needs for an adaptive online information system, UM 2003.
2. Brusilovsky P. (1996), Adaptive Hypermedia, an attempt to analyse and generalize, In Multimedia, Hypermedia, and Virtual Reality. Lecture Notes in Computer Science.
3. Brusilovsky P., Adaptive Hypermedia, User Modeling and User-Adapted Interaction, 2001.
4. Cleary C., Bareiss R., 1996, Practical methods for automatically generating typed links. Hypertext '96. Washington DC, USA.
5. Da Silva P., Van Durm V, Duwal E., Olivie H., concepts and documents for adaptive educational hypermedia: a model and a prototype, 2nd workshop on Adaptive Hypertext and Hypermedia, 1998, Pittsburgh, USA.
6. De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Aha! The adaptive hypermedia architecture, HT'03, United Kingdom.
7. Henze N., Nedjl W., Student modeling in an active learning environment using bayesian networks, UM 1999.
8. Herder E., Van Dijk B., Personalized adaptation to device characteristics, International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2002.
9. Kavcic A., The role of user models in adaptive hypermedia systems, Electrotechnical Conference, 2000. MELECON 2000.
10. Krishna K., Krishnapuram R., A Clustering Algorithm for Asymmetrically Related Data with Applications to Text Mining. International Conference on Information and Knowledge Management, 2001, Atlanta, Georgia, USA. pp.571-573.
11. Lawrie D., Croft B., Rosenberg A., 2001, Finding Topic Words for Hierarchical Summarization. Proceedings of the 24th annual international ACM SIGIR conference. New Orleans, Louisiana, USA.
12. Njike H. Gallinari P., Learning generalization/specialization relations between concepts – application for automatic building thematic document hierarchies, RIAO, 2003.
13. Rich E. (1979), User Modeling via Stereotypes, Cognitive Science, 3(4), pp. 329-354.
14. Rojas, Pelechano, Fons Navigational properties and user attributes for modelling adaptive web applications, Engineering the Adaptive Web (EAW'04) Workshop, AH'2004, Eindhoven, The Netherlands.
15. Salton G., Singhal A., Buckley C., Mitra M., 1996, Automatic Text Decomposition Using Text Segments and Text Themes. Hypertext 1996. pp. 53-65
16. Sanderson M., Croft B., 1999, Deriving concept hierarchies from text. In Proceedings ACM SIGIR Conference '99. pp.206-213.
17. Schneiderman B., Tree visualization with tree-maps: 2-d space-filling approach, ACM Transactions on Graphics, Vol. 11, No. 1, January 1992.
18. Schwarzkopf E., An adaptive Web site for the UM2001 conference, Proceedings of the UM2001 Workshop on Machine Learning for User Modeling.
19. Zhu T., Greiner R., Häußl G., Learning a model of a web user's interests, UM'03, pp 65-75.