Learning Comprehensible Conceptual User Models for User Adaptive Meta Web Search

Martin E. Müller
Institute for Semantic Information Processing
University of Osnabrück, D-49069 Osnabrück, Germany
Martin.Mueller@cl-ki.uni-osnabrueck.de

Abstract

In course of the OySTER project our goal is to induce conceptual user models that allow for a transparent query refinement and information filtering in the domain of WWW meta-search. User models which describe a user's interest with respect to an underlying ontology allow for a manual user model editing process and also pose a well defined problem for a conceptual inductive learning task. OySTER is a research prototype that is currently being developed at the university of Osnabrück.

Introduction

User Modeling and Machine Learning. User models represent assumptions about a user. User modeling systems infer user models from user interaction, store user models and induce new assumptions by reasoning about the models. These models are used within the system in order to adapt to the user. Furthermore, these models shall be accessible to the user — they should be both understandable and manually modifyable. Incorporating machine learning into this framework often leads to intertwined representation formalisms for both user models and information resources, "efficient" representations that are motivated by algorithms and filtering processes which cannot be explained to the user.

Contents. In the first section we will motivate the use of conceptual user models and according learning techniques. We then describe the user model representation formalism that is used within the OySTER[1] project (c.f. [11]). In the third section we demonstrate how different learning task can be carried out on these models. Finally, we discuss our approach and provide a short summary which includes a brief overview of related projects.

In this paper, we try to give a precise though not too formal overview of the interesting user modeling and ontology refinement processes within OySTER. Further technicalities had to be truncated with respect to limited space.

1 Adaptive Information Retrieval from the Web

Efficient User Modeling in Recommender Systems. In course of building recommender systems for the WWW, many projects (c.f. [1, 2, 6, 12]) tackled the problem of personalized document filtering by using n-ary vectors representing the user's interest. This implies an intertwined representation of information resources (namely the representation of a web document as a vector) and user models. Though vectors *represent* a user's interest, they do not explicitly *describe* a user's interest. Accordingly, vector based methods *perform* pretty well but they do not really

[1] http://mir.cl-ki.uni-osnabrueck.de/oyster/

explain the outcome.

The OySTER approach. Within our approach we want to explicitly represent a user's interest that which is lucid to the user himself and thus allow for user maintenance. Furthermore, we discriminate between domain dependent aspects (like terminology) and independent aspects (like representation language). The core ideas behind the intelligent meta search engine OySTER are as follows:

- The user model M_u^* is a conceptual description over concept hierarchies for document contents $\mathcal{C}$ and document types $\mathcal{T}$. It consists of an explicit representation of the user's interest M_u^+ as well as of a representation of concepts the user is not interested in, M_u^-.
- User queries are *refined* with respect to a user model in order to disambiguate query terms from the very beginning.
- Responses are *filtered* on aggregated results with respect to M_u^*.
- We discriminate document *type* and document *category*; thus obtaining sorted conceptual descriptions of document classifications.

Query refinement is achieved by generating a set of queries from the initial search string using keywords that are attached to the categories that occur within the user model M_u^+.

The question of whether a document d matches a user's interest therefore is answered by two steps: (1) What is the document classification? (2) Does this document classification fit in the user model? In other terms: User u is interested in d, if $M_u^* \models \mathbf{A}_{\mathcal{T}\times\mathcal{C}}(d)$, where $\mathbf{A}_{\mathcal{T}\times\mathcal{C}}$ is a classifier for document type and category and M is a user model.Using conceptual user models, one could describe "fitting" as subsumption: a document fits a user model, if the user model subsumes the document's classification (see section 2). Note, that the domain dependent ontology is both used for document classification as well as for user model descriptions. Classification with respect to conceptual descriptions and efficient retrieval in the web using this technique has already been demonstrated by the OntoBroker project, see [4]. The WebKB project, c. f. [3], aims to a large knowledge based system that describes the content of the WWW using similar techniques as OySTER.

The reason for using two hierarchies is as follows: a search query for the phrases **lecture note** and **machine learning** becomes a query for documents *about* "machine learning" which are of the *type* "lecture note". Since there are much less document types than content category descriptions, we can efficiently interpret types as sorts thus obtaining sorted conceptual descriptions.

Our document type hierarchy $\mathcal{T}$ is described in [10]; a few examples are depicted in figure 1.[2] As an ontology for document contents $\mathcal{C}$ we consider a library systematics, the universal decimal classification (see also [10]), handcrafted ontologies or any other classification system (as e.g. provided by the ACM).

For both $\mathcal{T}$ and $\mathcal{C}$, we use existing classification algorithms $\mathbf{A}_{\mathcal{T}}$ and $\mathbf{A}_{\mathcal{C}}$[3]. A simple document type classifier $\mathbf{A}_{\mathcal{T}}$ is currently being developed in course of a master's thesis (c.f. [5]). The choice of $\mathcal{C}$ still depends on pending project proposals.

```
id description
--+-------------------------------------------
20 publication
22 publication.lecture
25 publication.lecture.notes
23 publication.lecture.resource
24 publication.lecture.syllabus
33 publication.researchpaper
34 publication.researchpaper.abstract
35 publication.researchpaper.full
31 publication.researchpaper.full.published
32 publication.researchpaper.full.unpublished
21 publication.selfdescription
```

Figure 1: Document Types, $\mathcal{T}$.

[2] $\mathcal{T}$ contains 32 classes including types such as *homepage*, *student hompage*, *research article* or *linklist*.

[3] Since we concentrate on user modeling, we can abstract from the classifaction system and the classifaction process. Thus, we work with small, partially handcrafted ontologies in the first testing phase.

2 Conceptual User Models

Given an ontology of concepts for describing document contents, a user model containing information about the user's interest can be represented by a set of expressions over a semi–lattice. In more informal terms: one can think of a user model as differently shaded parts of a tree; see figure 2. This visualization also can be used for explaining a user the according user model and even for a manual user model editing process (where the user is allowed to change the model by "point and click" actions on the concept hierarchy). In figure 2, backslashed shaded

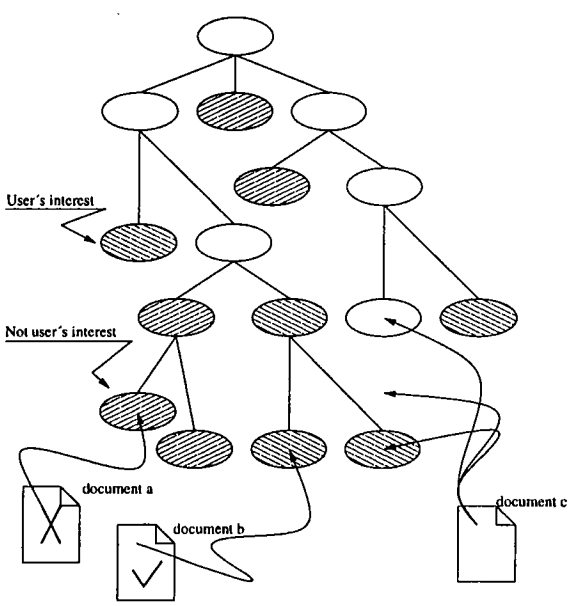

Figure 2: A user model with respect to $\mathcal{C}$

nodes correspond to concepts the user is interested in (M_u^+), while slashed nodes represents concepts the user definitely is not interested in (M_u^-). Note, that interpreting the tree as a subsumption lattice (as we would do in any proper concept hierarchy), interest in a parent node implies interest in child concepts. Note further, that interests can be distributed all over the tree and that interests can be unions, joins and meets (given a proper lattice.This implies, that such a concept hierarchy usually is *not* a lattice, since greatest lower bounds are not always defined (except for artificial $\perp$–elements). More precisely, M_u^+ and M_u^- are sets of expressions which represent different aspects of user interests. Each concept c in this description is order sorted by a document type t. A simple example of an interest aspect could be $\langle$**publication**, $c\rangle$, which is a **publication** about c. More complex expressions would be meets, as e.g. $\langle$**publication**, "machine learning"$\rangle \sqcap$ $\langle$**publication**, "user modeling"$\rangle$ which would be a publication about machine learning in user modeling. Note that $\sqcap$ and $\sqcup$ do *not* distribute over different aspects, since an interest aspect in machine learning and an aspect in underwater robotics does not neccessarily imply an interest in "machine learning in underwater robotics" or even "machine learning or underwater robotics" (unless explicitely stated). This way, we can represent several "independent" user model aspects which can be used for mutual hard feedback generation (see last section). More precisely, $M_u^* \models \mathbf{A}_{\mathcal{T}\times\mathcal{C}}(d)$ thus means: There is a (sub-) concept c of an expression in M_u^+, such that d is classified as a document of category c. A weak definition of interest would be to include those documents which cannot be proven to be not of interest: i.e. $M_u^* \not\models \mathbf{A}_{\mathcal{T}\times\mathcal{C}}(d)$ iff there is no concept c which is subsumed by any expression in M_u^- and d is classifed as being s c. Furthermore, one could expand the representation language by weighed class memberships, such that a for a document d, $\mathbf{A}_{\mathcal{T}\times\mathcal{C}}(d)$ would deliver a list of pairs $\langle\langle t, c\rangle, p\rangle$, where p is the probability of d belonging to the category c and type t.

In our system we shall investigate an approach for order sorted inductive logic programming (ILP, as described in [8]) for inducing user models. Thus, we interpret M_u^* as a set of order sorted Horn clauses with two different predicates **pinterest** and **ninterest** defining M_u^+ and M_u^-, respectively. Subsets of those clauses define aspects: Figure 3 represents two aspects of user u's interests in clauses 1-3. It means, that u is interested in machine learing, user modeling *and* cognitive science *or* user modeling, tutoring systems *and* machine learning of according document types. The user also is interested in personal hompages on diving and group hompepages about submarine robotics. Similar, clauses 4 and 5 describes

```
pinterest(u, 01, D) :-
  a_CxT( D, 13$'machine learning'),      (1.1)
  a_CxT( D, 12$'user modeling'),         (1.2)
  a_CxT( D, 19$'cognitive science').     (1.3)
pinterest(u, 01, D) :-
  a_CxT( D, 13$'user modeling'),         (2.1)
  a_CxT( D, 19$'tutoring systems'),      (2.2)
  a_CxT( D, 12$'machine learning').      (2.3)
pinterest(u, 02, D) :-
  a_CxT( D, 11$'diving'),                (3.1)
  a_CxT( D, 15$'submarine robotics').    (3.2)

ninterest(u, 01, D) :-
  a_CxT(D, 13$'language acquisition'),   (4.1)
  a_CxT(D, 12$'student modeling'),       (4.2)
  a_CxT(D, 19$'cognitive science').      (4.3)
ninterest(u, 01, D) :-
  a_CxT(D, 13$'student modeling'),       (5.1)
  a_CxT(D, 19$'tutoring systems'),       (5.2)
  a_CxT(D, 12$'human_comp interact').    (5.3)
```

Figure 3: `pinterest`: user u's interest

the dual of aspect `01`. Note, that the same clause here could be used for defining negative interest for more than just one single aspect. This will become interesting when looking for negatively labelled data during user model induction as described in the next section. Thus, we will be able to perform content based classification [7] that goes beyond statistical phrase clustering. Given relevance feedback on a sufficiently large sample, user model induction on $\mathcal{T} \times \mathcal{C}$ becomes a supervised learning task.

3 Inducing User Models

Using conceptual user models poses a well defined learning task for inductive logic programming techniques.

Learning User Models. Now, given sets $E_u^+(i)$ and $E_u^-(i)$ of positively and negatively labeled samples for a user u and aspects i, we can define a learning task. Example sets are sets of documents and their respective classifications (i.e. $\langle d, \mathrm{A}_{\mathcal{T}\times\mathcal{C}}(d)\rangle$). Thus, a trivial intial user model would be a set of Horn clauses

$$\texttt{pinterest}(u, i, D)\texttt{:-a\_CxT}(D, dtype\$dcategory)$$

for all *doctype$doccategory* that have been positively labelled. Order sorted generalization then yields a first hypothesis. As an illustration, consider the following example: Let there be a concept $c \in \mathcal{C}$ with four child nodes c_1, c_2, c_3 and c_4. Relevance feedback shows, that user u was interested in docuemtns of categories c_1, c_2 and c_3. Furthermore, u has not negatively labeled any d which was classified as c_4. Therefore,

$$\texttt{pinterest}(u, i, D)\texttt{:-a\_CxT}(D, t\$\texttt{c}).$$

would be a correct hypothesis as delivered by a so-called truncation operator. Here, t is the least upper bound of all types of documents in the sample under consideration. Thus, M_u^* would describe any document which is classified as $c' \sqsubseteq c$ as something that belongs to u's interest. The role of document types within this generalization step is illustrated in [10]. Intra–Construction on aspect `01` of the user model as described in figure 3 then would further lead to another hypothesis:

```
pinterest(u, 01, D) :-
  a_CxT(D,11$'machine learning'+'user modeling'),
  a_CxT(D,16$'new class').
```

Document type `11` is the lub of types `12` and `13`. The meaning of the new class `'new class'` and the join `'machine learning'+'user modeling'` is explained in the next paragraph.

Refining Concept Hierarchies. Given the last example of the preceeding section, we also found that we obviously learned new categories. The concept `'machine learning'+'user modeling'` emerged out of the body literal conjunctions `(1.1),(1.3)` and `(2.1),(2.3)` of the input clauses and thus represents an intersection of the both concepts. The `'new class'` can be canonically defined (just as it is done by fanning operators in logic programming): `'new class'` is `'tutoring systems'` *or* `'cognitive science'` but certainly not `'tutoring systems'` $\sqcup$ `'cognitive science'`. In both cases,

guided user feedback for class definition should be considered. The first newly invented category could be `'ml4um'` which actually is defined as `'machine learning'`$\sqcap$`'user modeling'`. The second concept definition could be used for defining several aspects. With a growing number of user models, the category hierarchy might soon become much too coarse. $\mathcal{C}$ is too coarse, if for two users u and v their respecitve models are "similar" although their labelled samples are "heterogeneous". Another key for coarseness is sudden growth of concept extensions; i.e. the classifiers do not distribute documents nearly uniformly over the classification tree.
Now, imagine some equivalent user model aspects for two different users u and v. Such a case gives rise to two different interpretations:

1. If the labelled data of u and v has a "significantly large" intersection, the examples of the disjoint union could be used for building a group model or for collaborative labelling. We will discuss this issue in the next paragraph.

2. If the labelled data sets are "nearly completely" disjoint, it is quite reasonable, that m_u^i and m_v^j should not be equal.[4]

In the latter case, we can use the example sets for mutual exclusive learning tasks in order to learn new concepts.

Feedback Generation. Obtaining relevance feedback is a crucial drawback in the WWW domain; especially in WWW search applications. Only few users will be provide at least partial feedback about delivered search results. Thus we use different techniques in order to receive labeled training data: On the one hand, we will use a Slider like interface in order to receive hidden explicit feedback on search results (c.f. [1]). But since this will provide only a small amount of feedback, we will have to make use of more sophisticated methods for generating labelled data.

[4]Note, that this case could be a follow-up of the former case, if u and v actually turn out to have different interests. This can easliy be checked by asking u to label data from E_v and vice versa. See next paragraph.

A standard bootstrapping method for obtaining initial feedback data is to scan a user u's homepage. Collecting the user's "hot links" from his linklist (which can be done during a registration procedure) forms a first rough picture of E_u^+.
When learning user models, we focus on aspects i. For each aspect we can generate a set of additional negative examples by taking into account positive examples for disjoint aspects of the same user. This technique yields a strong negative bias on hypotheses. In order to increase the number of positive examples, we will have to make use of other users' examples. By way of collaborative filtering we can increase the set of positive examples by examples that were labelled by other users with "similar" interests. More examples can be extracted by using labeled data from users whose E_v^- coincides with E_u^+. Their negative examples can be interpreted as positive exmples for u. Learning the user's disinterest, i.e. M_u^-, turns out to be the dual case: The positively labelled examples are interpreted as negative feedback and vice versa. Thus we can construct a sequence of enlargements of the labelled data sets.

4 Conclusion and Prospects

Conclusion. We have described a machine learning based approach for user modeling in the web search domain which emphasizes the following topics:

1. user models are transparent to the user,

2. the information filtering process becomes lucid and can be explained,

3. a manual user model adaption by the user himself is possible.

Though ILP is not a prototypical tool for learning in this context, we have motivated that user modeling might profit from this method. Finally, we have drawn a borderline that discriminates classification tasks and user modeling where the interface consists of conceptual descriptions.
But breaking with intertwined representation also reveals the drawback of this approach: The overall performance of the system—i.e. precision of search re-

sults with respect to the user's interest—depends on **both** classification and user modeling. Furthermore it depends on the fact, that inducing user models depends on classification. Thus, we presuppose *perfect* classifiers. Even worse, induction of new concepts presupposes *perfect* learning of new perfect classifiers each time a new concept has been invented.
Future evaluation work will show, if deliberate violation of these assumptions causes worse results in practice.

Open Questions and current work. The approach presented so far also gives rise to several interesting open research questions:

1. How many examples are needed in relation to the coarseness of an initial concept hierarchy in order to be able to induce sufficiently precise user models?
2. What are reasonable conditions for defining "similiar" user models and "heterogenous" samples as introduced in section 3?
3. What are reasonable conditions for defining "significantly large intersections" and "nearly complete disjointness" as introduced in section 3?

Among others, these questions are in focus of current work.

Future work and Prospects. A detailed description of the system architecture can be found in [9]. Depending on pending project proposals, we choose $\mathcal{C}$ and according classifiers. Until then, we will implement first user modeling modules using a small handcrafted ontolgy for a first prototype whose evaluation may help answering the questions mentioned above.

References

[1] BALABANOVIC, M. An interface for learning multi-topic user profiles from implicit feedback. In *AAAI-98 Workshop on Recommender Systems* (Madison, Wisconsin, July 1998).

[2] BILLSUS, D., AND PAZZANI, M. Learning probabilistic user models. In *User Modeling: Sixth International Conference, UM97*, A. Jameson, C. Paris, and C. Tasso, Eds. Springer Wien New York, Vienna, New York, 1997. Workshop Paper.

[3] CRAVEN, M., DIPASQUO, D., FREITAG, D., MCCALLUM, A., MITCHELL, T., NIGAM, K., AND SLATTERY, S. Learning to extract symbolic knowledge from the world wide web. In *Proc. of 15th Nat. Conf. on Artificial Intelligence (AAAI-98)*. AAAI Press, 1998.

[4] FENSEL, D., ERDMANN, M., AND STUDER, R. Ontobroker: How to make the www intelligent,. In *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems, Workshop (KAW98)* (Banff, Canada, 1998).

[5] HEISSING, C. Clustering URLs by Pathname Generalisation. In *LWA 99: Lernen, Wissen und Adaptivität* (Magdeburg, 1999), Universität Magdeburg, pp. 301–306.

[6] JOACHIMS, T., FREITAG, D., AND MITCHELL, T. Webwatcher: A tour guide for the world wide web. In *Proceedings of IJCAI 97* (August 1997).

[7] KRULWICH, B. Learning document category descriptions through the extraction of semantically significant phrases. In *Proc. of the 14th IJCAI, Workshop on Data Engineering for Inductive Learning* (1995).

[8] MÜLLER, M. E. Eine Erweiterung der inversen Resolution auf sortierte Hornklauselmengen. In *Beiträge zum 7. Fachgruppentreffen Maschinelles Lernen* (1995), K. Morik and J. Herrmann, Eds., no. 580 in Forschungsberichte der Universität Dortmund, Fachbereich Informatik.

[9] MÜLLER, M. E. An Intelligent Multi-Agent Architecture for Information Retrieval from the Internet. Tech. rep., Institute for Semantic Information processing, Univ. Osnabrück, 1999. available from `http://mir.cl-ki.uni-osnabrueck.de/~martin/index.publications.html`.

[10] MÜLLER, M. E. Inducing conceptual user models. In *7. GI Workshop "Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen" (ABIS-99), LWA Sammelband, ISBN 3-929757-26-5* (Magdeburg, Germany, 1999).

[11] MÜLLER, M. E. Machine Learning based User Modeling for WWW Search. In *Proc. 7th Int. Conf. on User Modeling (UM-99)* (Banff, 1999). Workshop paper "Machine Learning for User Modeling".

[12] PAZZANI, M., MURAMATSU, J., AND BILLSUS, D. Syskill & Webert: Identifying interesting Web Sites. In *Proc. of the National Conference on Artificial Intelligence* (Portland, OR, 1996).