Learning Structural Classification Rules for Web-page Categorization

Heiner Stuckenschmidt[1], **Jens Hartmann**[2] **and Frank van Harmelen**[1]
[1] AI Department, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands
e-mail: {heiner,frankh}@cs.vu.nl
[2] Center for Computing Technologies, University of Bremen
Universitaetsallee 21-23, 28359 Bremen, Germany
e-mail: jhart@tzi.de

Abstract

Content-related metadata plays an important role in the effort of developing intelligent web applications. One of the most established form of providing content-related metadata is the assignment of web-pages to content categories. We describe the Spectacle system for classifying individual web pages on the basis of their syntactic structure. This classification requires the specification of classification rules associating common page structures with predefined classes. In this paper, we propose an approach for the automatic acquisition of these classification rules using techniques from inductive logic programming and describe experiments in applying the approach to an existing web-based information system.

Introduction

Metadata plays an important role in the effort of developing intelligent web applications. This data may cover very different aspects of information: technical data about storage facilities and access methods coexist with content descriptions and information about intended uses, suitability and data quality. In this paper we focus on a specific type of metadata, namely metadata related to the contents of a web-page. A common approach to capture this kind of metadata is so-called web-page categorization (Pierre 2001). Here, web pages as a whole are assigned to a set of classes representing a certain topic area the page belongs to. In order to apply this approach there has to be a set of classes to be used as targets for the classification task.

A problem that remains is the classification itself which can be a tremendous effort considering the size of normal web-sites or even the web itself. There is a need for automatic or semi-automatic support for the classification process that has already been observed by others. Jenkins and others, for example, use text mining technology in order to generate RDF models describing the contents of web-pages (Jenkins *et al.* 1999). It has been argued that web page classification can be significantly improved by using additional information like other kinds of metadata (Pierre 2001) or linguistic features (Basili, Moschitti, & Pazienza 2001). We follow an approach that exploits another kind of additional information namely the syntactic structure of a web page. This can be done because it has been shown that it is possible to identify syntactic commonalty between web-pages information about similar topics (Craven *et al.* 2000) and to learn wrappers for automatically extracting information from web pages (Freitag & Kushmerick 2000).

In this paper, we present an approach for the automatic acquisition of content-related metadata. The approach is based on structural classification rules that are learned from examples. We exemplify the approach in the domain of environmental information to be classified according to different subject areas. In section 2 we introduce the web-based information system BUISY which serves as an application domain. Section 3 describes the Spectacle system for classifying web-pages according to structural classification rules and its application to the BUISY system. An approach for automatically generating classification for the Spectacle system is motivated and presented in section 4. We summarize with a discussion of achievements and open problems.

The Application Domain

The advent of web-based information systems came with an attractive solution to the problem of providing integrated access to environmental information according to the duties and needs of modern environmental protection. Many information systems were set up either on the Internet in order to provide access to environmental information for everybody, or in intranets to support monitoring, assessment and exchange of information within an organization. One of the most recent developments in Germany is BUISY, an environmental information system for the city of Bremen that has been developed by the Center for Computing Technologies of the University of Bremen in cooperation with the public authorities. The development of the system was aimed at providing unified access to the information existing in the different organizational units for internal use as well as for

the publication of approved information on the internet.

Abbildung 1: The main areas of the BUISY system

Meta-data plays an important role in the BUISY system. It controls the access to individual web pages. Each page in the BUISY system holds a set of meta-data annotations reflecting its contents and status (Vögele, Stuckenschmidt, & Visser 2000). The current version of BUISY supports a set of meta tags annotating information about the data-object's type, author, dates of creation- and expiration, as well as relevant keywords and the topic area of the page. The "Status" meta-tag indicates whether the data-object is part of the Internet or the Intranet section of BUISY.

```
<meta name="Status" content="Freigegeben"/>
<meta name="Typ" content="Publikation"/>
<meta name="Author" content="TJV"/>
<meta name="Date" content="10-04-1999"/>
<meta name="Expires" content="31-12-2010"/>
<meta name="Keywords" content="Wasser, Algen"/>
<meta name="Bereich" content="Wasser"/>
```

At the moment, this meta-data is used to provide an intelligent search facility for publications of the administration concerned with environmental protection. The user selects a document type and a topic area. Based on the input, a list of available publications is generated.

The Spectacle Approach

We have developed an approach to solve the problems of completeness, consistency and accessibility of metadata as described above. This is done on the basis of rules which must hold for the information found in the Web site, both the actual information and the metadata (and possibly their relationship). This means that besides providing Web site contents and metadata, an information providers also formulate classification rules (also called: integrity constraints) which should hold on this information. An inference engine then applies these integrity constraints to identify the places in the Web site which violate these constraints. This approach has been implemented in the Spectacle Workbench, developed by the Dutch company AIdministrator (`http://www.aidministrator.nl`). We describe the different steps of our approach. Formulating and applying classification criteria and integrity constraints is done in a three step process (van Harmelen & van der Meer 1999).

Constructing a Web-site Model

The first step in our approach to content-based verification of web-pages is to define a model of the contents of the web-site. Such a model identifies classes of objects on our web-site, and defines subclass relationships between these classes. For example, pages can be about water, soil, air, energy, etc. Each of these types of pages can again be subdivided into new subclasses: water-pages can be about waste-water, drinking water, river-drainage, etc. This leads to a hierarchy of pages which is based on page-contents, such as the example shown in Figure 2.

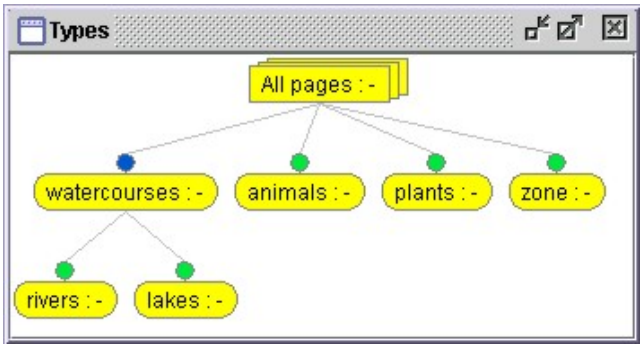

Abbildung 2: An Example Classification Tree

A subtle point to emphasize is that the objects in this ontology are objects on the web-site, and not objects in the real-world described by the web-site. For example, the elements in the class riversäre not (denotations of) different rivers in a specific region, but they are *web-pages* (in this case: web-pages talking about rivers). As a result, any properties we can express for these objects are properties of the *pages on the web-site*, as desired for our classification purposes.

Defining Syntactic Criteria for Classes

The first step only defines the classes of our ontology, but does not tell us which instances belong to which class. In the second step, the user defines rules that determine which Web pages will be members of which class. In this section, we will briefly illustrate these rules by means of three examples.

Figure 3 specifies that a rule is about "watercourses" if the keyword "Gewaesser" appears in the meta-information of the web-page. The rule succeeds if for example the following code appears in the web-page:

```
<META NAME="Keywords" CONTENT="Gewaesser, Bericht">
```

In the typical case, a page belongs to a class if the rule defined for that class succeeds for the page.

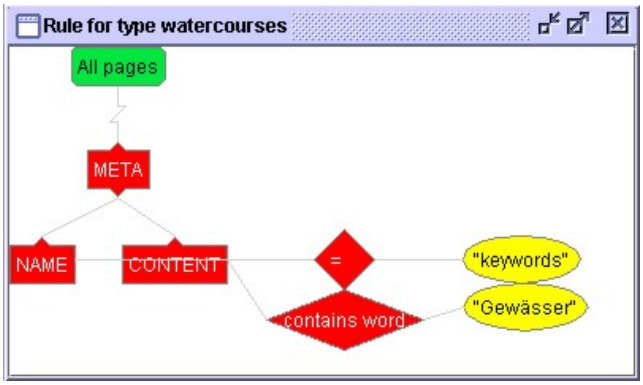

Abbildung 3: Example of a Classification Rule Using Metadata

Classifying Individual Pages

While the human user of Spectacle performs the previous steps, the next step is automatic. The definition of the hierarchy in the first step and the rules in the second step allow the Spectacle inference engine to automatically classify each page in the class hierarchy. Note that classes may overlap (a single page may belong to multiple classes) and may not be exhaustive (pages do not need to belong to any sub-type.

The rule format has been defined in such a way as to provide sufficient expressive power while still making it possible to perform such classification inference on large numbers of pages (many thousands in human-acceptable response time).

Learning Spectacle Rules

We conducted experiments in using Spectacle to classify the web-pages in the BUISY system. In the course of this case study eight groups of AI students with some experience in knowledge representation and knowledge-based systems independently specified classification rules reflecting the conceptual structure of the system. This corresponds to the second step in the process described above. It turned out that the creation of structural classification rules is still a difficult and time-consuming task which requires some knowledge about the information to be annotated. In order to avoid the effort of analyzing the whole web-site we are currently developing an approach for automatically learning page structure from examples and partial specifications and encode them in terms of Spectacle rules.

Requirements

For this to be possible, a trade-off must be struck between the expressiveness of these rules (to allow the information providers to express powerful constraints) and the efficiency with which the rules can be tested on specific Web-pages by the inference engine, and the learnability of these rules. Following a suggestion in (Rousset 1997), we have chosen the following general logical form for our rules and constraints also known as *positive database update constraints* in the database area:

$$\forall \vec{x}[\exists \vec{y} \bigwedge_i P_i(x_k, y_l)] \implies [\exists \vec{z} \bigwedge_j Q_j(x_k, z_m)] \quad (1)$$

or equivalently

$$\forall \vec{x}, \vec{y}[\bigwedge_i P_i(x_k, y_l)] \implies [\exists \vec{z} \bigwedge_j Q_j(x_k, z_m)] \quad (2)$$

where the $\vec{x}$, $\vec{y}$ and $\vec{z}$ are sets of variables, and each of the P_i and Q_j are binary predicates.

Furthermore, the classification rules used in Spectacle are always *hierarchical* (i.e. recursion free), and also free from negation.

This class of formulae is less expressive than full first order logic over the predicates P_i and Q_j (because of the limited nesting of the quantifiers), but is more expressive than Horn Logic (because of the existential quantifier in the right-hand side of the implication).

If we further restrict the rule format, and we drop the existential quantification in the right-hand side, we are left with a very restricted form of logic programs: *hierarchical normal programs over binary predicates*. This of course directly suggests the use of ILP techniques for learning such classification rules.

Inductive Logic Programming

Acquiring Spectacle rules requires an approach able to learn first order formulas of the form described above. Inductive Logic programming (Muggleton 1999) is such an approach. It learns a general hypothesis H from a set of given examples E and possibly available background knowledge B. The set of examples is normally decided into a set of positive E^+ and negative examples E^-, so that

$$E = E^+ \cup E^-.$$

B, E^+, E^- and H are logic programs. The goal is to generate a consistent hypothesis, that

$$B \wedge H \models E.$$

The main operations to generate such a hypothesis are *specialization* and *generalization* which is called a **top-down** respectively a **bottom-up approach**. ILP Systems can also be discerned between **single-predicate** and **multiple-predicate** learners. All

examples given to a single-predicate learner are instances of one predicate P. Instead of a multiple-predicate learner which examples are instances of more than one predicate. Another distinction can be made how the examples are given to the learner. Given all examples at once, it is called **batch learning**. At **incremental learning** an example is given one by one and the theory covers the actual set of examples. If there exits a possibly interaction between the teacher and the learner while learning is in progress, it is called an **interactive learner** otherwise it is a **none-interactive learner**.

The ILP System we used is Progol (Muggleton 1995), a top-down, multiple-predicate non-interactive batch learner which has been successfully used in many real world applications (Finn *et al.* 1998; King *et al.* 1996; Muggleton, King, & Sternberg 1992; Srinivasan *et al.* 1996). Progol uses inverse entailment to generate only the most specific hypothesis. An A* like algorithm is used to search through the hypothesis space. To restrict the hypothesis space (*bias*), the teacher defines first-oder expressions called *mode declarations* which define predicate and function symbols. Thereby types and parameters has to be defined, too.

Learning Results

We conducted an experiment in learning classification rules in order to assign pages of the BUISY system to one of the eight main areas of the system depicted in figure 1. We restricted the task to the use of meta tags in the documents. We can use knowledge about the position of these tags in the document in order to focus the learning process making it more efficient. Using this approach classes are discriminated by the values of the content attribute present in the different predefined meta-tags. For example Progol generates following hypothesis for the given training set of the class *Luft*:

```
document(A) :- metatag(A,bereich,luft).
```

The hypothesis states that the following tag structure has to appear on a web page belonging to the area:

```
<META NAME="bereich" CONTENT="luft">
```

Similar results were generated for the other areas. Admittedly this result is not very surprising as the meta tag explicitly assigns a page to a content area. The real benefit of the learning approach, however, is its ability to find classification criteria that are not obvious. In order to discover such unexpected patterns as well, we defined a learning strategy on top of the Progol system. Once a valid hypothesis is found, it is stored in a separate solution file. Then all occurrences of the defining structures are deleted from the training data and the learner is run on the data again.

Assessment

In order to assess the quality of our learning approach, we determine the accuracy of the learned rules in terms of the ratio of correctly classified pages. We use the following notation to refer to classification results:

P(A): right positive (pages from the class covered by the rule)

$\neg P(A)$: wrong negative (pages from the class not covered by the rule)

$\neg P(\neg A)$: right negative (pages not from the class that are not covered by the rule)

$P(\neg A)$: wrong positive (pages not from the class that are covered by the rule)

In our experiments, we used a ratio of 1:2 between positive and negatives examples (for each positive example there are two negative ones) Using these definitions, we use the following definition of accuracy:

$$Accuracy = \frac{P(A) + \neg P(\neg A)}{P(A) + \neg P(A) + \neg P(\neg A) + P(\neg A)} * 100$$

The accuracy is determined by splitting the set of all web pages into a training-set and a test-set, where 70% of all pages belong to the training and 30% to the test set. Below we give accuracy measures for our experiments based on this ratio.

Classification by Metadata

We conducted an experiment in learning classification rules in order to assign pages of the BUISY systems to one of the eight main areas of the system depicted in figure 1. We restricted the task to the meta-tag goal predicate described above. Using this approach classes are discriminated by the values of the content attribute present in the different predefined meta-tags. Figure 1 summarizes the results that are close to 100% accuracy as expected.

metatag(I, N, C)					
Class	*P(A)*	$\neg P(A)$	$\neg P(\neg A)$	$P(\neg A)$	*Acc.*
Abfall	13	0	26	0	100
Boden	11	0	22	0	100
Luft	28	0	56	0	100
Natur	20	1	42	0	98,41
Wasser	50	8	114	2	94,25
Total					98,53

Tabelle 1: Classification Results based on meta-tags

The real benefit of the learning approach, however, is its ability to find classification criteria that are not obvious. In order to discover such unexpected patterns as well, we defined a learning strategy on top of the Progol system. Once a valid hypothesis is found, it is stored in a separate solution file. Then all occurrences of the defining structures are deleted from the training data and the learner is run on the data again. This process is repeated until no more valid hypotheses are found. As a result of this strategy we get alternative definitions of the different classes.

Classification by other Criteria

In the course of our experiments it turned out, that we already get good classification results when analyzing the titles of pages in the BUISY system. Despite the fact that our learning approach is based on purely syntactical string matching, we reached an accuracy of almost 90% on pages of the BUISY system. The results are summarized in figure 2.

doctitle(D, T)					
Class	$P(A)$	$\neg P(A)$	$\neg P(\neg A)$	$P(\neg A)$	*Acc.*
Abfall	3	10	24	2	74,36
Boden	8	3	22	0	90,91
Luft	24	4	56	0	95,24
Natur	16	5	42	0	92,06
Wasser	50	8	114	2	94,25
Total					89,4

Tabelle 2: Classification Results based on Page Titles

Beside the analysis of page titles there are many other options like analyzing the URL of a page or external links to other pages as well as to email addresses. In the case of the BUISY system, these alternative criteria where not of great use, however, in other systems comparable results were achieved with these criteria.

Discussion

We presented an approach for automatically acquiring structural classification rules for classifying web pages. The approach can be used to enhance web-based information systems with content-related metadata in terms of an assignment of pages to certain topics. We argued that Inductive Logic Programming is the method of choice for implementing the learning process, because the Spectacle system we use for classification relies on first order rules with an expressiveness close to Prolog, which is the language learned by the Progol system we use. We discussed the pre-processing steps needed to apply our approach to semi-structured information and showed some learning results. Finally we discussed the need to explicitly handle noisy data which is very likely to occur on the web. We briefly sketched the approach we use to cope with noise. However, the problem of noisy and incorrect data is still an open problem which needs further investigation. This concerns the pre-processing part as well as the learning process itself. Our further work will be concerned with the implementation of a learning strategy on top of Progol that handles noise using the already mentioned approach of splitting the training set in combination with a rough estimate of the probability of generated hypotheses.

References

Basili, R.; Moschitti, A.; and Pazienza, M. T. 2001. Nlp-driven ir: Evaluating performances over a text classification task. In Nebel, B., ed., *Proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI-01.*

Brickley, D.; Guha, R.; and Layman, A. 1998. Resource description framework (rdf) schema specification. Working draft, W3C. http://www.w3c.org/TR/WD-rdf-schema.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence* 118(1-2):69–113.

Finn, P.; Muggleton, S.; Page, D.; and Srinivasan, A. 1998. Pharmacophore discovery using the Inductive Logic Programming system Progol. *Machine Learning* 30:241–271.

Freitag, D., and Kushmerick, N. 2000. Boosted wrapper induction. In *Proceedings of AAAI-00*, 577–583.

Jenkins, C.; Jackson, M.; Burdon, P.; and Wallis, J. 1999. Automatic rdf metadata generation for resource discovery. *Computer Networks* 31:1305–1320.

King, R.; Muggleton, S.; Srinivasan, A.; and Sternberg, M. 1996. *Proceedings of the National Academy of Sciences* 93:438–442.

Muggleton, S.; King, R.; and Sternberg, M. 1992. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering* 5(7):647–657.

Muggleton, S. 1995. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming* 13(3-4):245–286.

Muggleton, S. 1999. Inductive logic programming: issues, results and the LLL challenge. *Artificial Intelligence* 114(1–2):283–296.

Pierre, J. M. 2001. On the automated classification of web sites. *LInkping Electronic Articles in Computer and Information Science* 6.

Rousset, M.-C. 1997. Verifying the world wide web: a position statement. In van Harmelen, F., and J. van Thienen., eds., *Pro-ceedings of the Fourth European Symposium on the Validation and Verification of Knowledge Based Systems (EUROVAV97).*

Srinivasan, A.; Muggleton, S.; King, R.; and Sternberg, M. 1996. Theories for mutagenicity: a study of first-order and feature based induction. *Artificial Intelligence* 85(1,2):277–299.

van Harmelen, F., and van der Meer, J. 1999. Webmaster: Knowledge-based verification of web-pages. In Ali, M., and Imam, I., eds., *Proceedings of IEA/AEI99*, LNAI. Springer Verlang.

Vögele, T.; Stuckenschmidt, H.; and Visser, U. 2000. Buisy - using brokered data objects for environmental information systems. In Tochtermann, K., and Riekert, W.-F., eds., *Hypermedia im Umweltschutz.* Marburg: Metropolis Verlag.