IDENTIFYING ONTOLOGY COMPONENTS FROM DIGITAL ARCHIVES FOR THE SEMANTIC WEB

JRG Pulido∗ R Herrera† M Aréchiga‡ A Block§ R Acosta¶ S Legrand∥

ABSTRACT

This paper describes an approach for identifying Ontology components by using Self-Organizing Maps (SOM). Our system represents the knowledge contained in a particular domain, any kind of digital archive, by assembling and displaying its ontology components. This novel approach provides a solution to the problem of semi-automatic ontology construction, supports mechanisms that explore domains, and allows knowledge components to be displayed in a browsable manner. Further processing may be carried out on the extracted knowledge to be embedded on the semantic web for software agents to use.

KEY WORDS

Semantic web, ontology learning, self-organizing maps.

1 Introduction

It is known that the web contains several billion of static pages connected by hyperlinks [26, 29]. Reaching them is a gigantic challenge having into account that current search engines only contain a small percentage of the total of documents in the web. Furthermore, this small amount of reachable documents is in an unstructured way, meaning that software agents understand actually nothing about the actual content of them. In other words, these documents can be read but not understood [3]. It would be useful to develop representations of the information contained in digital archives and create intelligent systems supporting interactive searching. In this paper we describe an approach for helping in the semi-automatic construction of ontologies for such web sites. The remainder of this paper is organized as follows. In section 2 some related work is introduced. Our approach is outlined in section 3. Results are presented in section 4 and conclusions and further work in section 5.

2 Related Work

One of the most important challenges that the semantic web poses in dealing with large amounts of on-line knowledge is the mapping of unstructured information, suitable for humans, to formal representation of knowledge [5]. In the next subsections we have a brief look at some work done on Ontologies as well as Semantic Maps.

2.1 Constructing Ontologies

A representation that brings order and structure to a web site can be referred to as an Ontology. Representing knowledge about a domain as an ontology is a challenging process which is difficult to achieve in a consistent and rigorous way. It is easy to lose consistency and to introduce ambiguity and confusion [4]. An important observation in this context is that there is a significant manual effort involved in translating ontologies [27]. Nevertheless, ontologies are a useful form of knowledge representation which may be used to support the design and development of intelligent software applications and expert systems. Web ontologies can take rather different forms to traditional ones. New approaches, including advanced ontology languages have been proposed, such as OIL, DAML, OWL [2, 17, 10, 14, 8]. In [13] the use of the so-called Simple HTML Ontology Extension (SHOE) in a real world internet application is described. This approach allows authors to add semantic content to web pages, relating the context to common ontologies that provide contextual information about the domain. A similar approach is presented in [11]. Most tag-annotated web pages tend to categorize concepts, therefore there is no need for complex inference rules to perform automatic classification. One of the most exciting uses of an ontology, in the context of the semantic web, is to support the development of agent-based systems for web searching [4, 21].

2.2 Semantic Map Systems

An interesting project is presented in [18], where the results of applying the WEBSOM2, a document organization, searching and browsing system, to a set of about 7 million electronic patent abstracts is described. In this case, a document map is presented as
a series of HTML pages facilitating exploration. In [25] a distributed architecture for the extraction of metadata from WWW documents is proposed and is particularly suited for repositories of historical publications. This information extraction system is based on semi-structured data analysis. The system output is a *metadata object* containing a concise representation of the corresponding publication and its components. In that research gatherers have been designed as a combination of a parser, based on a context-free grammar, and a web robot, which navigates the links contained in the basic document type to infer the document structure of the entire site. These meta-data objects can be interchanged with other web agents, then classified and organized.

### 3 Methods

Our software is written in Java, which offers robust, multiplatform, and easy networking functionalities. Being an object-oriented programming language, it also facilitates reuse as well. Speed is not an issue anymore as computer processors are faster and faster. Java and its various APIs are powerful enough for constructing ontology software systems. The idea of combining ontologies and semantic maps has motivated our work. For the semantic web to become a reality, we need to transform the current web into a web where software agents are able to negotiate and carry out trivial tasks for us. Doing this manually, would mean a bottleneck for the semantic web. We need software tools that help us accomplish this enterprise.

Our system consists of two applications: Spade and Grubber [7, 8]. The former pre-processes html pages and creates a document space. The latter is fed with the document space and produces knowledge maps that allow us visualize ontology components contained from a digital archive. They may later be organized as a set of *Entities*, *Relations*, and *Functions*. Problem solvers use this triad for inferring new data from knowledge bases [11, 12, 28, 22].

#### 3.1 The Algorithm

SOM can be viewed as a model of unsupervised learning and an adaptive knowledge representation scheme. Adaptive means that at each iteration a unique sample is taken into account to update the weight vector of a neighbourhood of neurons [17]. Adaptation of the model vectors take place according to the following equation:

$$m_i(t + 1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{1}$$

where $t \in \mathbb{N}$ is the discrete time coordinate, $m_i \in \mathbb{R}^n$ is a node, and $h_{ci}(t)$ is a neighbourhood function. The latter has a central role as it acts as a smoothing kernel defined over the lattice points and defines the stiffness of the surface to be fitted to the data points. This function may be constant for all the cells in the neighbourhood and zero elsewhere. A common neighbourhood kernel that describes a natural mapping and that

---

1. Anything about which something can be said.
2. Interconnections between entities in a universe of discourse (eg part-of).
3. A special type of interrelation (eg is-a).
is used for this purpose can be written in terms of the
Gaussian function:

\[ h_{ci}(t) = \alpha(t) \exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right) \] (2)

where \( r_c, r_i \in \mathbb{R}^2 \) are the locations of the winner and
a neighbouring node on the grid, \( \alpha(t) \) is the learning
rate (0 \( \leq \alpha(t) \leq 1 \)), and \( \sigma(t) \) is the width of the kernel.
Both \( \alpha(t) \) and \( \sigma(t) \) decrease monotonically.

The major steps of our approach are as follows:

a) **Produce a document space** A document space
is created with the individual vector spaces.

b) **Construct the SOM** By using a suitable num-
ber of cells and iterations the map \([24]\) is trained
with the docuspace.

Once the SOM is done, ontology components can be vi-
ualized clustered together. One important difference
between our approach and Kohonen’s is that we do not
use *average context* \([24, 16]\) to create the docuspace.
This helps us reduce the dimensionality of the dataset.
Contextual information is clustered together anyway.
Preliminary results were surprisingly close to our in-
tuitive expectations. After this, some other ontol-
ygy tools such as editors can be used to organize this
knowledge. Finally, it can be embedded into the digital
archive (Fig.1) where it was extracted from by means
of any of the ontology languages that exist.

4 A domain two datasets: The animal
kingdom case

This section presents two experiments that we have
have carried out. First we present, though in a dif-
f erent and enhanced way, some results that have been
published before by other authors. Then in subsection
4.1 the results from applying our system to a bigger
domain, from the same kingdom, are shown. Both sub-
sections describe two maps of the domain in two ways,
the front view showing Attributes, and the transposed
view showing Entities. Both views display knowledge
componentes of the domain clustered together.

4.1 The animal dataset

In \([24, 23]\) the animal dataset is presented by means
of a html page. Our approach uses a 4x4 SOM and
presents the same data by using colored areas. In
our experiment we found that one dominant charac-
teristic amongst the animals is their size, e.g. birds
are small, mammals come in two sizes. On the other
hand, birds of prey and hunting mammals, small ani-
mals with feathers, big animals with hooves, and the
ones with four legs and hair are also clustered together.

This is consistent with earlier tests carried out on the
dataset. Both SOMs are shown in figure\([3]\). It must be
noticed that the vector spaces for *zebra* and *horse*, and
*owl* and *hawk* are equal. The ones for *hen* and *duck*
are approximately equal. Similarly, the vector spaces
for the Attributes *feather* and *two legs*, and *hair* and
*four legs* are equal. That is why some areas overlap
and produce a combination of colorings.

4.2 The zoo dataset

For our second analysis another animal dataset has
been used. This is commonly known as the zoo dataset.
It contains 101 instances belonging to a seven already
identified classes and 16 attributes. It combines 15
boolean attributes and a numerical one. The original
dataset also includes *name* and *class*. We have used
the former as one of the header files and the latter
has been omitted here as our approach is going to find
the classes. Two 10x10 SOMs have been used here for
the analysis. It took just about 30 seconds training
each map, the front and transposed view, on a IBM
netvista computer (1.7GHz, 256Mb RAM). Browsing
the SOMs gives us a clear idea and helps us understand
what the domain is all about.

For instance we can readily identify *bird*\([4]\),
*fish*\([5]\), *insect*\([6]\), and *mammal*\([7]\) within the domain.
*Amphibians* and *reptiles* have not been easy to find
as they overlap other classes. Attributes like *toothed*,
*backbone*, *tail* are shared by *haddock* and *pitviper* for
instance. Other attributes that can be seen clustered
together by our software tool and that, of course,
are not shared by the mentioned instances are *eggs*, preda-
Figure 3. The animal dataset, each cell labelled. **Left.** Attributes describing groups of animals. **Right.** Entities (animals) sharing attributes.

tor, venomous. Sharing the terms *domestic* we find animals such as *chicken*, *calf*, *pussycat*, and so forth. We may consider *backbone* as an important attribute for it determines whether these animals are vertebrate or not (Fig.4).

The experiments we have presented in this section show that Self-Organizing Maps are an efficient software tool to analyse domains. We have reported the use of SOM for other domains [1]. The next step would be to use other ontology tools to organize and embed this knowledge into web pages.

5 Conclusion

An ontology can be used to give a sense of order to unstructured digital sources. It also provides a common vocabulary of *concepts* and *relationships* which may be used to inform a viewer, a search engine, or other software entities such as agents. A common ontology would enable collaborators to work together with a minimal risk of misunderstanding. Principled techniques that allow the ontological engineer to deal with the problems caused by such complexity need to be developed, and the ideas in this paper have shown promise as avenues of investigation. The novelty of our approach is that SOM offer clustering and visualization features not present in other techniques, and as it has been presented helps in the semi-automatic construction of ontologies by identifying components from digital archives. Further research avenues we are working on involve the use of hybrid systems in such a way that by combining clustering techniques with the already trained feature vectors we may refine the classification of the knowledge components from the domain [19, 24]. Must be said that a domain expert is always required in order to obtain a desirable level of accuracy in the ontology. Should that be done manually, then the semantic web will not become a reality in the next couple of decades due to this bottleneck. Ontology learning tools are essential for the realization of the semantic web for the job to be done is quite complex.

References


Figure 4. The zoo dataset, groups labelled. **Left.** Attributes describing groups of animals. **Right.** Entities (animals) sharing attributes.

Figure 5. Two basic taxonomies from the animal kingdom. **Left.** Animal dataset (sec. 4.1), attributes at the bottom are part-of instances. **Right.** Zoo dataset (sec. 4.2), attributes not shown. Note that the former is actually a subset of the latter (vertebrate).


