Chapter 7

Conclusions

7.1 Summary

This work presented OntOAIr, an ontology learning method that is robust enough to construct valid lightweight ontologies from multiple collections of documents with minimal human intervention. This method is based on four main tasks: harvesting, representation, clustering and formalization.

The harvesting task obtains the documents from the collections; it depends on external factors such as network overhead and availability of collections. The request verbs of OAI-PMH protocol were used for harvesting, although other methods and techniques for distributed systems could carry out this task.

The representation task constructs a vectorial representation for each harvested document after a stopword elimination process and the assignment of weights to keywords. The OntOAIR method uses the TF*IDF method, but other methods such as Jaccard index can also be used.

The clustering task applies the FIHC algorithm to produce a tree of labeled clusters. Based on experiments with Tales collection, we proposed an adaptation of this exclusive hierarchical algorithm which removes the child pruning process and the sibling merging process from the original version proposed by [16]. The selection of values for the input parameters (*support of an itemset*, *global support* and *cluster support*) strongly influences the accuracy of the clustering; however, experimental evaluations (using the F measure and the entropy) have shown that the FIHC algorithm is as at least as good as UPGMA algorithm and bisecting *k-means* algorithm.

The formalization task transforms the tree of clusters into a lightweight ontology. The XML, RDF and OWL languages were used to explore different alternatives to formalize the ontologies constructed by the OntOAIr method. The XML language produces simple, readable and reusable ontologies, but their semantics is only accessible to humans, not to machines. The RDF language enables the definition of the vocabulary and the specification of properties, but the expressivity to represent relationships is limited. The OWL language offers RDF compatibility, computational completeness and decidability. However, the OWL model is more complex than the XML Schema and the RDF schema.

In comparison with other ontology learning methods, the OntOAIR method has the following advantages: (1) uses general domain documents, (2) produces a hierarchy of labeled clusters, (3) constructs ontologies in linear time with respect to the number of documents, and (4) supports scalable information retrieval models (similarity functions only involve queries and cluster labels). The main disadvantages are the following: (1) the determination of appropriate values for input parameters, (2) the assignment of keywords weights do not take into account the structure of documents, and (3) the

representation of concepts of a single term (the current version of the method does not manage phrases).

Ontologies of records are an information retrieval model on their own because they support indexing and retrieval. The indexing process is based on two tasks: (1) the selection of informative elements of each record, and (2) the organization of the records in data structures that enable the search. The retrieval process has been implemented through the keyword-based retrieval model and the ontology-based exploration model. These models assume that a record that does not match any term in the query is not relevant.

An algorithm to establish similarity between queries and groups of records has been developed for the keyword-based retrieval model. We have conducted some small scale experimentation and, as a result, we have demonstrated that the effectiveness of the keyword-based retrieval model is similar to that of the vector spaces model. Averaged values show a recall of 87% and a precision of 71%. However, further experimentation and larger document sets are needed to test and improve our method.

The agent-based prototypical system called OntoSIR was implemented to validate the potential of the OntOAIr method and the feasibility of the retrieval model and exploration model. OntoSIR can be regarded as a searching service for the Open Archives Initiative community. In addition to OAI-compliant data providers, OntoSIR can use other document collections. The main disadvantage of this system is its asynchronous operation when ontologies are not previously constructed.

The ontologies constructed by the OntOAIr method constitute a data model able

to provide an unambiguous and shared terminology represented in schemas and models that human and software agents can both understand and use. Further, the use of XML, RDF and OWL languages make ontologies accessible for Semantic Web technologies such as RDQL[1] , RQL[2] , or OWL-QL[3] .

Partial results of this work have been published in different forums. For instance, [44] and [45] described the design of the agents in the OntoSIR system; and [46] its architecture. The representation of hierarchies of documents in markup languages was first proposed in [48] and [43]. The keyword-based information retrieval was presented in [47]; while in [41] an analysis to support maintenance of the ontologies is done. [42] described the evaluation of the OntOAIr method.

7.2 Future work

The OntOAIr method can be used to support manual construction of ontologies, to cluster the responses of search engines, or as a basis to support reasoning in Semantic Web contexts. However, testing the OntOAIr method with large collections is the first of our planned tasks for the immediate future. Then, open research lines to be considered are:

- Adding inference mechanisms that search through the ontologies and deduce results in an organized manner

[1] RDF Data Query Language
[2] RDF Query Language
[3] OWL Query Language

- Defining tasks to support the maintenance of ontologies

Inference is a useful tool to complete missing information. In OntoSIR, it can be used for implicit query expansion or to automatically maintain the consistency of the ontologies of records.

Ontologies are rarely static, thus we propose two tasks for their maintenance: First, the inclusion of a set of records in a previously constructed ontology without compromising the accuracy of the clustering or the effectiveness of the retrieval, and second, the management of versioning, which should help to keep track of the evolution of ontologies.

The construction of ontologies of records is a step towards the formal encoding of the content of document collections. We expect the reuse of ontologies of records by other applications and ontologies to support Semantic Web tasks such as knowledge acquisition, knowledge management and similarity-based retrieval.