Motivating Ontology-Driven Information Extraction

Burcu Yildiz[1] and Silvia Miksch[1, 2]

[1] Institute for Software Engineering and Interactive Systems,
Vienna University of Technology, Vienna, Austria
{yildiz,silvia}@ ifs.tuwien.ac.at

[2] Department of Information and Knowledge Engineering,
Danube University Krems, Krems, Austria
silvia.miksch@donau-uni.ac.at

Abstract. Ontologies can provide Information Extraction Systems (IES) with much needed domain and task knowledge. Yet, it has to be analysed to what extent and in which form ontologies can be utilised to enhance the overall performance of an IES. Further, the use of ontologies requires an accurate management of the same. The most important issue with this respect is to keep the ontology up-to-date, because the domain it represents will change inevitably. In this paper we motivate the use of ontologies within IES, especially for automating the extraction-rule generation process, which currently is the main obstacle to portable and scalable IESs.

Keywords: Information Extraction, Ontology-driven Information Extraction, Ontology, Ontology Management

1 Introduction

Information Extraction (IE) is defined as a form of natural language processing in which certain types of information must be recognised and extracted from text (Riloff, 1999). It is an important and popular research field of the current time; for it tries to extract relevant information from the overwhelming amount of data we are facing today. Here, the question of what actually 'relevant information' is comes to ones mind immediately. Unfortunately, it is hard to give a clear answer to this question and it is even harder to communicate the answer to a computer system.

Within the last three decades, the field of IE gained a lot of importance, mainly fostered by the Message Understanding Conferences (MUCs) started in 1987 (Marsh & Perzanowski, 1998), which provided a platform for researchers to present and evaluate their work. One of the main contributions of the MUCs to IE research is the definition of concrete extraction tasks. Starting with the fundamental task of extracting named entities, more complicated tasks were introduced over the years requiring the extraction of several properties of entities (template-element task), the extraction of information related to pre-specified events (scenario template task) and the more complicated task of extracting relations between entities or events (template relation task).

The work presented at the MUCs showed that it is very hard to generate extraction rules that are general enough to extract relevant information from unseen documents, yet specific enough to perform well for the given task specification. Further, rule generation turned out to be an iterative process, where an initial set of rules are applied on the data and according to the results are adopted, until the system yield a reasonable performance. This kind of rule generation, where a knowledge engineer is generating the rules manually is called the *knowledge engineering approach*. It is clear that the generation of extraction-rules by hand represents the main obstacle towards portable and scalable IESs, because for the IES to be applied on a different domain or task often the generation of a whole new set of extraction rules is required. Therefore, the *(semi-) automatic training approach* has been introduced (Kushmerick & Thomas, 2002), where the human intervention is reduced to perform annotations indicating relevant information in a given data corpus, using which the IES can learn extraction patterns. However, this approach requires the annotation of large number of files, so the human intervention cannot be considered to be much less than in the knowledge engineering approach.

For both approaches it can be said that human intervention yields to subjective IESs, because humans often do not agree on the relevance of a part of text. Ontologies, being explicit specifications of conceptualisations (Gruber, 1993), can be used in that context to provide IESs with a machine-readable definition of relevant information by representing the domain knowledge in a formal way. We think, that ontologies can be utilised with both approaches. The knowledge engineer can commit to the ontology, which would guarantee that the extraction rules are tailored to extract the kind of information represented in the ontology, whereas with the second approach, an annotator can commit to the ontology and annotate only parts of text that are relevant from the ontology's point of view.

In this paper, we will motivate the development of ontology-driven IESs, where the ontology is utilised to automate the rule generation process by exploiting all kinds of available knowledge in the ontology. Further, we will motivate the integration of ontology management services to keep the underlying ontology up-to-date. Before doing this, we will first give an overview on existing related work in the field of IE where ontologies are being used in the course of the extraction-rule generation process.

2 Related Work

Embley (Embley, 2004) presents an approach for extracting and structuring information from data-rich unstructured documents using extraction ontologies. With "data-rich" he means data that has a number of identifiable constants such as dates, names, times, and so forth. He proposes the use of the Object-oriented Systems Model (OSM) (Embley et al., 1992) to represent extraction ontologies, because it allows regular expressions as descriptors for constants and context keywords. Both, the generation of the ontology and the generation of the regular expressions are being done manually. The ontology is then parsed to build a database schema and to generate extraction rules for matching constants and keywords. After that, recognisers are invoked which use the extraction rules to identify potential constant data values and context keywords. Finally, the generated database is populated using heuristics to determine which constants populate which records in the database.

For the extraction of relevant information from car advertisements, the presented approach achieved recall ratios in the range of 90% and precision ratios near 98%. For domains with more complex content and where the relevant records (e.g., car advertisements) are not clearly separated from one another, the performance decreases, though.

Aitken (Aitken, 2002) presents an approach to learn information extraction rules from natural language data using Inductive Logic Programming (ILP). He proposes the use of an ontology as a reference to which an annotator can commit to while annotating the data with ontology terms. The supervised induction algorithm then uses the annotations to generate extraction rules.

Dowell and Cafarella (Dowell & Cafarella, 2006) present an automatic and domain-independent ontology-driven IES called OntoSyphon. The system takes an ontology as input and uses its content for specifying web searches to identify possible semantic instances, relations, and taxonomic information. For example, for a concept “Mammal” in an ontology, the system specifies web searches using the phrase patterns introduced by Hearst (Hearst, 1992), such as “mammals such as”, etc. The system then searches the web for occurrences of these phrases and extracts candidate instances.

Maedche, Neumann and Staab (Maedche, Neumann, & Staab, 2002) present a semi-automatic bootstrapping approach that allows a fast generation of ontology-based IESs relying on several basic components: a core IES, an ontology engineering environment, and an inference engine. They start with a shallow IE model that specifies domain-specific lexical knowledge, extraction rules, and an ontology. A domain specific corpus is then processed with the core IES. Based on this processed data, the IE model is extended using different learning approaches. Finally, the human modeler reviews the learning decisions and decides whether to stop the process or not.

However, our focus is on unsupervised and adaptive ontology-driven IESs, that are able to generate extraction rules automatically from a given input ontology and that are also able to react to changes in the domain.

3 Ontology-driven Information Extraction System

We already mentioned that domain knowledge in form of an ontology makes it possible to develop portable IES. But the use of an ontology requires the accurate management of the same, because the domain it represents will change inevitably over time. So, the IES has to provide services that are able to adapt the underlying ontology to the current reality. Therefore, we propose the use of an Ontology Management Module (OMM) to be integrated into an IES (compare Figure 1) that provides the mentioned functionalities.

Further, we aim to develop unsupervised IESs and therefore concentrate on automatic rule generation. For that purpose we propose the use of a Rule Generation Module (RGM) as part of an IES (compare Figure 1), which is able to produce extraction rules from a given ontology automatically.

In the following, we will present the system architecture of such an IES and explain the functionalities of the most important modules in the system.

3.1 System Architecture

In Figure 1, the general architecture of an ontology-driven IES is depicted. The required domain knowledge is captured in an ontology. The required rule-making knowledge and task knowledge are implicitly coded in the RGM.

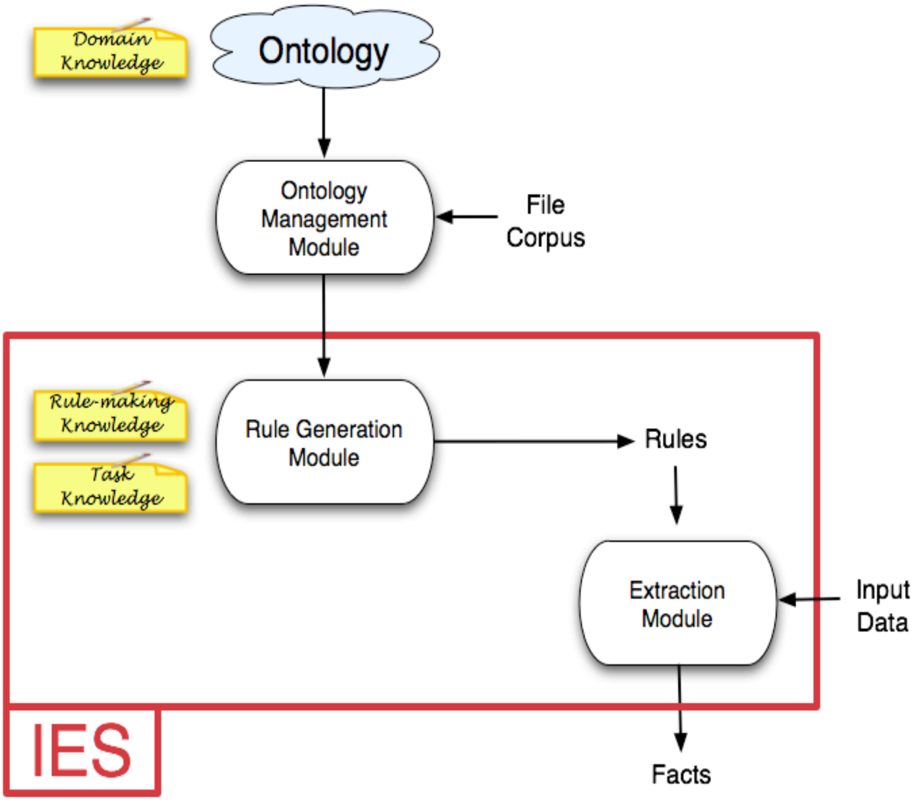

Figure 1: General architecture of an ontology-driven Information Extraction System

3.2 The Ontology Management Module (OMM)

The Ontology Management Module (OMM) is responsible for several tasks concerning the ontology life cycle. To develop such a module, the vast amount of research already done in the fields of Ontology Learning and Ontology Evolution can be exploited.

In the following we will give an overview of the responsibilities of an OMM for enabling portable, scalable and adaptive IESs.

Ontology Learning and Population

To be able to process ontologies in different representation languages and to be more flexible towards up-coming standards, an OMM should be based on an abstract ontology model. Developers may use, for example, the Jena Semantic Web Framework for Java[1], which provides an abstract ontology model that covers all common ontological components (e.g., concepts, instances, properties, etc.). In cases where the system is provided only with a file corpus as input, an ontology model has to be generated using the file corpus. For that purpose several ontology learning algorithms presented by Maedche (Maedche, 2002) can be used.

Ontology Evolution

Change management of ontologies is responsible for keeping the ontology model consistent during processes such as generation or adaptation. During these two processes, components are going to be removed or added to the ontology model. It is essential to decide what to do in cases where a change can cause an inconsistency. Stojanovic (Stojanovic, 2004) proposed the use of so called “evolution strategies” to define the course of action when facing critical changes in advance. A case captured by such an evolution strategy is for example “what to do with orphaned subconcepts?”; where the course of action could be to delete it together with its parent concept or to relate it with the superconcept of its parent.

These strategies might need to be adapted w.r.t. the needs of a particular IES, because we think that changes themselves can be of interest for some IESs too. In such cases, ontology components should

[1] http://jena.sourceforge.net/

not be removed from the ontology, rather their valid times should be changed.

Another functionality that an OMM should provide is data-driven and perhaps also usage-driven change discovery (Stojanovic & Motik, 2002). Data-driven change discovery ensures the detection of changes in the file corpus attached to the OMM, whereas usage-driven change discovery reflects the changes in the users' interests. This might require the adaptation of the defined evolution strategies with additional descriptions for cases like "what to do with a component when all its source documents are being deleted from the file corpus?".

For providing all these functionalities a developer may use the work of Cimiano and Völker (Cimiano & Völker, 2005). They present a framework for data-driven change discovery with several integrated ontology learning approaches. They represent the learned knowledge at a meta-level, using an abstract ontology model, which they call Probabilistic Ontology Model (POM). For each learned component they calculate a value indicating the confidence level of the system, which allows the design of sophisticated visualisations of the POM. The integrated learning approaches in the ontology are able to learn is-a, instance-of, part-whole, and equivalence relations and restrictions on the domain and range of relations. Further, they claim that a particular application that wants to support data-driven change discovery has to meet several requirements. The most important one is to keep track of all changes to the data. Such a system should also allow for defining various change strategies, which specify the degree of influence changes to the data have on the ontology or the POM respectively.

3.3 Rule Generation Module

The rule generation module (RGM) is responsible for automatically generating extraction rules for the IES. It takes an ontology as input and generates extraction rules exploiting all kind of knowledge in the ontology.

Let us assume that someone who has no information about digital cameras is given a set of camera reviews and the job to mark information about several relevant properties of the reviewed cameras. The only domain knowledge that is provided to this person is in form of an ontology that represents only the relevant properties of digital cameras (see Figure 2), whereas it is also assumed that the person knows the semantics of the given ontology, that is, he knows that arrowed lines indicate subclasses and that labeled lines indicate data type properties. How would this person proceed?

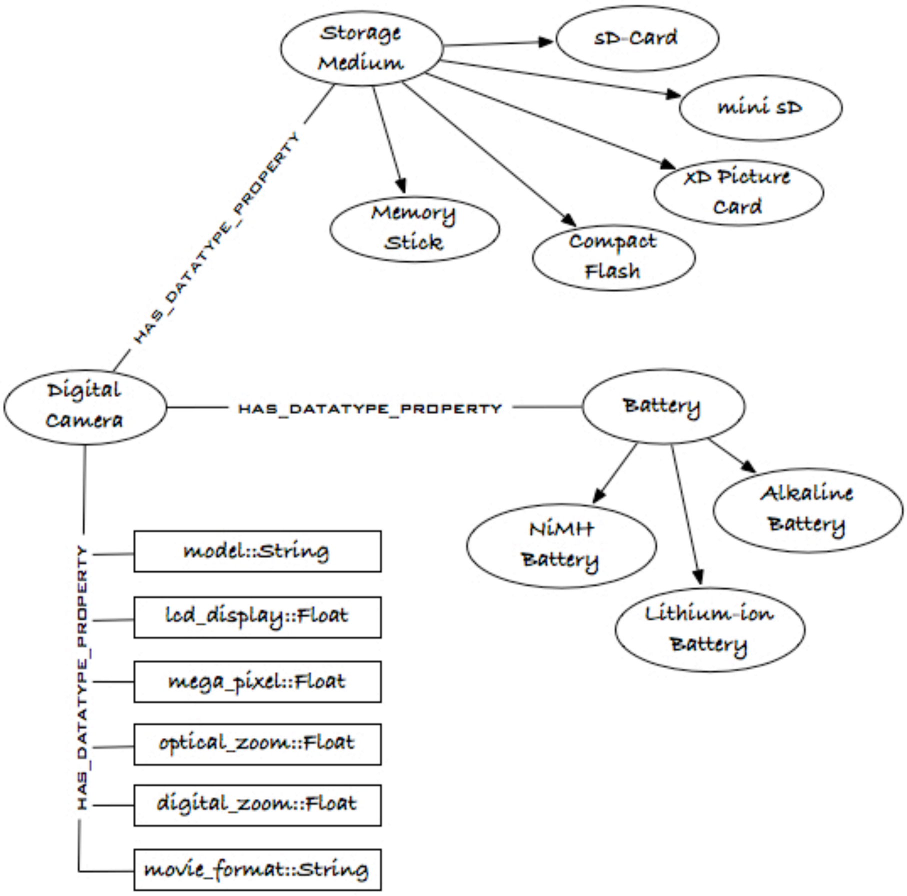

Figure 2: An example ontology of task relevant information about digital cameras

First of all he would look for words that are similar to the concepts names, for example 'storage', 'zoom', etc. Then he would look for words or numbers that fit the constraints of the concepts. For example, a float that appears in conjunction with the word 'optical zoom' would be an indicator for him that he is on the right track.

The following algorithm gives a general outline of one possible way in which the RGM of an IES could proceed:

Algorithm 1:

```
Rule Set R = Ø
Bag-of-words B = Ø
for each concept c in the ontology {
   B ∪ words in the name section of a concept
   B ∪ words that appear in the comment section of a concept

   for each property p of concept c {
      R ∪ regular expression to capture the datatype of p
      B ∪ words in the property´s name and comment section
   }
}

for each word w in B {
   Look for word w in input text
   if (found) {
      apply rules in R to the neighbourhood of w to find
      appropriate values
   }
   else {
      go on with next word
   }
}
```

So, we can say that the more constraints ontology components have the more specific the generated extraction-rules would be, because the constraints enable to narrow the range of possible values for particular properties. Else, sophisticated heuristics would have to be developed that are able to choose amongst different possible values.

4 Conclusion and Future Work

We motivated the use of ontologies in IES to develop ontology-driven IESs, which are unsupervised, portable, scalable, and adaptive. For that purpose we proposed the integration of an Ontology Management Module (OMM) into the system that is able to generate and integrate ontological knowledge and can detect changes in the domain represented by a file corpus. Further we proposed the development of a Rule Generation Module (RGM) as part of an IES, which is able to automatically generate extraction rules from an ontology.

The main field for future work is the field of ontology learning. Better learning approaches are needed, which are able to learn not only basic ontological components (e.g., concepts, instances, relations) but also non-hierarchical relations. The field of change management is also a candidate for future work, because other methods than evolution strategies to prevent inconsistency could be interesting.

Our main interest for future work, however, is to develop well performing rule generation methods.

References

Aitken, J.S. (2002). *Learning Information Extraction Rules: An Inductive Logic Programming Approach.* In Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02). Amsterdam: IOS Press.

Cimiano, P. & Völker, J. (2005). *Text2Onto – A Framework for Ontology Learning and Data-driven Change Discovery.* In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'2005), 227-238.

Dowell, L.K., Cafarella, M.J. (2006). *Ontology-driven Information Extraction with OntoSyphon.* International Semantic Web Conference, 428-44.

Embley, D.W. (2004). *Toward Semantic Understanding: An Approach Based on Information Extraction Ontologies.* In Proceedings of the 15th Australasian Database Conference, 3-12.

Embley, D.W., Kurtz B.D., Woodfield S.N. (2004). *Object-oriented Systems Analysis: A Model-Driven Approach.* Englewood Cliffs, New Jersey: Prentice Hall.

Gruber, T.R. (1993). *A Translation Approach to Portable Ontology Specifications.* Knowledge Acquisition, 5(2), 199-220.

Hearst, M. (1992). *Automatic Acquisition of Hyponyms from Large Text Corpora*. In Proceedings of the 14th International Conference on Computational Linguistics.

Kushmerick, N. & Thomas, B. (2002). *Adaptive Information Extraction: Core Technologies for Information Agents.* In Intelligent Information Agents: The AgentLink Perspective. Berlin/Heidelberg: Springer, 79-103.

Maedche, A. (2002). *Ontology Learning for the Semantic Web.* Massachusetts: Kluwer Academic Publishers.

Maedche, A., Neumann, G., & Staab, S. (2002). *Bootstrapping an Ontology-based Information Extraction System.* Intelligent Exploration of the Web. Heidelberg: Springer.

Marsh, E., Perzanowski, D. (1998). *MUC-7 Evaluation of Information Extraction Technology: Overview of Results.* In Proceedings of the Seventh Message Understanding Conference (MUC-7).

Riloff, E. (2002*). Information Extraction as a Stepping Stone Toward Story Understanding.* Understanding Language Understanding: Computational models of Reading, 435-460.

Stojanovic, L. (2004). *Methods and Tools for Ontology Evolution.* PhD Thesis. University of Karlsruhe, Germany.

Stojanovic, L., Motik, B. (2002). *Ontology Evolution within Ontology Editors.* In Proceedings of the OntoWeb-SIG3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management, 53-62.