Towards Web Information Extraction using Extraction Ontologies and (Indirectly) Domain Ontologies*

Martin Labský
Univ. of Economics, Prague
Czech Republic
labsky@vse.cz

Marek Nekvasil
Univ. of Economics, Prague
Czech Republic
nekvasim@vse.cz

Vojtěch Svátek
Univ. of Economics, Prague
Czech Republic
svatek@vse.cz

ABSTRACT
Extraction ontologies allow to swiftly proceed from initial domain modelling to running a functional prototype of a web information extraction application. We investigate the possibility of semi-automatically deriving extraction ontologies from third-party domain ontologies.

Categories and Subject Descriptors
H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

1. INTRODUCTION
Most approaches to *web information extraction* (WIE) deliver extracted information as somewhat weakly semantically structured from the knowledge engineering viewpoint; secondary mapping to ontologies is typically needed, which makes the process complicated and possibly error-prone. Approaches based on *extraction ontologies* (EO) [1], in turn, push ontologies more towards the actual extraction process through defining the concepts the instances of which are to be extracted in the sense of various attributes, their allowed values and higher level (e.g. cardinality or mutual dependency) constraints. EO are assumed to be hand-crafted based on observation of a sample of resources. They allow for rapid start of the extraction process, as even a very simple EO is likely to cover a sensible part of target data and generate meaningful feedback for its own redesign. However, to make maximal use of available data/knowledge and avoid overfitting to a few data resources examined by the designer, the whole process must not neglect pre-existing domain ontologies, labelled data and HTML formatting regularities. This is the rationale of our WIE tool under development called *Ex*, which combines richly-structured extraction ontologies with inductive and wrapper-based techniques [2]. Here we investigate the reuse of domain ontologies; the structure of EOs will however be explained first.

*(Produces the permission block, copyright information and page numbering). For use with ACM_PROC_ARTICLE-SP.CLS V2.6SP. Supported by ACM.

2. EX(TRACTION) ONTOLOGY CONTENT
EOs in *Ex* are designed so as to extract occurrences of *attributes* (such as 'age' or 'surname'), i.e. standalone named entities or values, and occurrences of whole *instances* of *classes* (such as 'person') as groups of attributes that 'belong together'.

Mandatory information to be specified for each *attribute* is: name, data type and dimensionality (e.g. 2 for computer monitor resolution like 800x600). Further extraction knowledge related to attribute *value* includes: textual value patterns; for numeric types: min/max values, numeric value distribution and units of measure; min/max value length in tokens or length distribution. Extraction knowledge about attribute *context* includes textual context patterns and formatting constraints. *Nesting* of attributes is allowed, their *course* can be specified, and external resources of *named entities* can be referenced. Additional constraints (such as numerical comparisons) can be specified via JavaScript[1]. Finally, *HTML formatting constraints* may be provided.

Each *class definition* enumerates the list of attributes, and for each attribute, a *cardinality* range. Extraction knowledge for class *content* consists of: apriori probability of each attribute being included as part of a class instance (as opposed to standalone occurrence), and class content patterns (such as attribute ordering). Extraction knowledge for class *context* again consists of textual and HTML formatting patterns.

All types of extraction knowledge yield pieces of *evidence* indicating the presence of a certain attribute or class instance. Every piece of evidence may be equipped with two *probability estimates*: precision and recall; they can be estimated from data or set manually.

[1]ECMAScript, see `http://www.mozilla.org/rhino`.

An alpha version of *Ex* is publicly available[2]. It has so far been tested in three domains; details on the first two are in [2], and the third is reviewed in the next section. In the MedIEQ project[3] we attempt to automatically evaluate medical website quality criteria (such as info on responsible medical professional) so as to ease accreditation by specialised agencies. In another application, in cooperation with one of largest Czech web portals, we extract information about products sold or described online; an excerpt of the EO for computer monitor descriptions is below. Finally, in the domain of weather forecasts we investigated the possibility to assist the ontology engineer in reusing existing *domain ontologies* in order to develop the extraction one/s.

```
<class id="Monitor">
 <attribute id="name" card="1" eng="0.6">
  <value>
   <pattern recall="0.3" prec="0.95">
    LCD <att ref="manuf"/> <ALPHANUM/>{1,2} </pattern>
  </value>
  <context>
   <pattern recall="0.25" prec="0.5">
    (model|monitor) name :? $ </pattern>
  </context>
 </attribute>
```

3. REUSE OF DOMAIN ONTOLOGIES

Due to slightly different modelling principles, *transformation* of domain ontologies (DOs) to EOs is needed. In order to transform a DO expressed in OWL into an EO we should (possibly repeatadly for multiple classes): (1) choose the *core* class C and add it to the EO; (2) create its attributes in the EO from structures of the DO; (3) formulate ontological constraints (concerning e.g. data type or cardinality) over attributes, based on constraints over properties from the DO or based on known instances; (4) formulate additional extraction knowledge as described in section 2. Examples of non-deterministic *transformation rules* for step (2) are:

- A *datatype property* may directly yield an attribute.
- A datatype property D of some C_1, together with a *chain of object properties* $(O_1, O_2, ..., O_n)$, where O_1 is object property of C, O_n is object property of C_1, and for every k, $1 \leq k \leq n-1$, there is a class having both O_k and O_{k+1} as its properties, may yield an attribute.
- A *set of mutually disjoint subclasses* of C may yield an attribute.

Similar heuristics may help suggest which class/es in the DO may become the *core* class in the EO, i.e. assist in step (1), or, more realistically, to check if a certain DO is suitable for transformation to EO, i.e. if the core class/es suggested correspond to the class whose instances are the target of the extraction task. Our start-up set of *selection rules* was:

1. Class with instances asserted in the DO should not become core class in the EO.
2. Classes that appear more often in the domain than in the range of object properties are candidates for core class/es.
3. Classes that appear in the filler of a minimum cardinality restriction are less likely to become the core class.
4. Classes that appear in the end of (especially, more than 2) chains of object properties are candidates for core class/es.

Table 1: Core class selection rule matches

Ontology name / Rule no.	1	2	3	4
weather-ont (from Semwebcentral)	1	4	-	-
WeatherConcepts (from LSDIS)	-	9	-	6
weather-ont3 (from AgentCities)	7	38	7	11

In the weather domain, the *selection rules* seemed to perform reasonably on our three DOs. Their classes `WeatherObservation`, `WeatherReport` and (by name coincidence) `WeatherReport` again, which were pointed to by the 'positive' rules (no.2 and 4), indeed look like meaningful concepts for which instances could be extracted. In contrast, e.g. rule no.1 (a 'negative' one) prevented the `Precipitation` class from *weather-ont* from being indicated as core class; rule no.3 acted the same for the `WeatherEvent` class in *weather-ont3*. Table 1 lists the number of activation of each rule for each ontology. In addition, *transformation rules* seemed, by first judgement, to suggest a sensible and inspiring, though by far not complete, skeleton of an extraction ontology. Testing this ontology on real weather forecast records is however needed for proper assessment.

The research was partially supported by the EC under contract FP6-027026, Knowledge Space of Semantic Inference for Automatic Annotation and Retrieval of Multimedia Content - K-Space. The medical website application is carried out in the context of the EC-funded (DG-SANCO) project MedIEQ.

4. REFERENCES

[1] D. W. Embley, C. Tao, and S. W. Liddle. Automatically extracting ontologically specified data from HTML tables of unknown structure. In *Proc ER '02*, 322–337, 2002. Springer-Verlag.

[2] M. Labský, V. Svátek, M. Nekvasil and D. Rak. Web Information Extraction using Extraction Ontologies. Submitted paper.

[2] `http://eso.vse.cz/~labsky/ex`
[3] `http://www.medieq.org`