FAETON: Form Analysis and Extraction Tool for ONtology construction

Rafael Berlanga\(^1\), Ernesto Jiménez-Ruiz\(^1\), Victoria Nebot\(^1\), y Ismael Sanz\(^2\)

\(^1\) Departament de Llenguatges i Sistemes Informàtics
\(^2\) Departament d’Enginyeria i Ciència de la Computació
Universitat Jaume I, Castelló, Spain

Resumen. This paper presents a method for semi-automatically building tailored application ontologies from a set of data acquisition forms. Such ontologies are intended to facilitate the integration of very heterogeneous data generation processes and linking with external sources useful to enrich the generated data. The resulting tool is being applied to the medical domain where well-known knowledge and linguistic resources are publicly available. Preliminary results are shown in this paper, and demonstrate that the approach can perform effectively. However, further experiments should be performed to demonstrate their applicability for hard integration issues.

1 Introduction

The Semantic Web is aimed at facilitating the automatic processing of Web contents and services. Nowadays there are many methods to migrate current Web contents to this semantic space. Most of them have concerned with the semantic annotation of Web pages by using Information Extraction techniques (see [12] for a review). Other works like [1] have concerned with analysing data records published in the Web (e.g. book information) to build proper ontological instances that facilitate their integration.

In this work we focus on the migration of data acquisition forms to the semantic space. More specifically, we deal with data acquisition forms coming from disparate workflows (usually from different organisational units), which generate data to be processed by Decision Support Systems. The high heterogeneity of these forms in both contents and structures makes it crucial to enrich them somehow by means of external knowledge resources (e.g. thesauri like WordNet and UMLS). In our case, we propose to build tailored application ontologies so that forms and data can be processed in an effective way. Moreover, the annotations provided by these ontologies are also useful for linking forms and data to existing external sources like bibliographic repositories and publicly available databases also annotated with the same resources.

Our application scenario for validating the proposed approach is a biomedical application developed in the Health-e-Child (HeC) project\(^3\). Data acquisition processes are medical protocols that have been specified within hospital departments. Thus, forms present very different formats and contents depending on the clinician specialities (e.g. Rheumatology, Cardiology, Oncology, etc.) and what they consider relevant for characterizing patients. Generated data is usually expressed in XML and stored, after anonymisation, in a distributed repository in order to be shared by other clinicians of the project. Among the tasks performed over this repository we highlight that of decision support, which consists on applying data mining algorithms and visualisation tools over a set of selected patients to find new evidences for proper diagnosis, follow-up and treatments.

In this scenario, application ontologies are essential to perform the following tasks: (1) to homogenise vocabulary and terminology used in the data forms, (2) to discover data forms

\(^3\) http://www.health-e-child.org
providing similar information, (3) to organise data fields according to the concepts provided by the ontology, and (4) to link forms and data to well-established external sources like PubMed and UMLS.

The main differences of our approach from other works in the literature mainly reside in the nature of the elements to be semantically annotated. Contrary to works based on Information Extraction techniques, we do not deal with plain text sources. Usually, forms contain many small text sections (labels) consisting of only a few words. Unlike wrapping approaches, we have no regularities to exploit as we deal with the data forms (schemas) instead of the data they can generate (instances), other than the implicit conventions of form layout [15, 3]. In other words, we face a more heterogeneous scenario where patterns are scarce.

The rest of the paper is organised as follows. Section 2 is dedicated to the extraction of structural properties of the forms. In Section 3 we present the semantic annotation of the forms as well as how related knowledge is imported from external domain ontologies. Section 4 is dedicated to the preliminary experiments. Finally we give some conclusions and present future work.

2 Extracting Structural Properties from Forms

The first techniques for the automatic acquisition of knowledge from data acquisition forms were developed in the context of the “paperless office” vision, and focused on for the processing of paper-based documents; a survey can be found in [13]. Most of this effort was focused in low-level techniques such as image segmentation and optical character recognition (document analysis), followed by a phase of so-called document understanding, which involved deriving a logical structure for the document.

In contrast with the approaches for paper-based documents we have just outlined, we will be dealing with electronic forms. Obviously, this greatly reduces the difficulty of document analysis, but it does not eliminate the need for it. Consider, for example, the case of forms produced by popular database tools such as Microsoft Access, or many HTML forms in Intranets, which provide little structural metainformation. The designers of these forms typically assume they are designed to be interpreted only by humans, and therefore they rely on the same visual conventions used by their paper-based counterparts.

2.1 Form analysis

In order to perform the analysis of an electronic form, we assume that we have access to the following basic information about it:

- The basic components in a form: labels (i.e. simple text) and standard controls such as text entries and combo boxes.
- Visual and geometric features of the basic components: font size, colour, absolute positions and rectangular areas are all useful and generally available.
- Type information associated to the controls, including the data type (e.g. a combo box may be restricted to a list of acceptable inputs) and, if possible, the data source to which a control is linked.

It is important to note that additional metainformation may be available depending on the original format of the form. For instance, Microsoft Access forms provide richer data than HTML forms, such as the data source of each control, which are exploited by FAETON.

This metainformation is processed during the analysis phase, whose goal is to obtain a logical representation of the form. In our case, this representation will contain the following related models:
Logical model The logical model is the main result of this phase, and contains the inferred hierarchy of elements in the original form. According to the natural groupings found in forms, the model is structured as a multi-level structure: forms are composed of sections, which in turn may contain labels, control groups, and independent controls. This model also contains basic horizontal relationships between controls and their corresponding labels.

Layout model Even in a poorly designed form, there is a limited number of visual styles (combination of font, colour, alignment, and so on) present. It is possible to automatically find a relationship between these styles (i.e. finding out which are more salient), which convey structural information. The layout model contains these descriptions.

Link model The interpretation of many forms does not proceed linearly; rather, it is directed by indications such as “if the answer to such question is yes, then proceed to that other question”. The correct extraction of these relationships is crucial, since they contain information about the workflow that underlies the form.

Figure 1 shows a fragment of data acquisition form processed by our approach. The form has two sections, entitled “Brain MRI” and “Spinal MRI”. Control names are drawn in grey (e.g. MRINORM, PRECON1, etc.) The rest of strings are labels. These labels can be associated to controls (e.g. “Pre-contrast Intensity” to PRECON1), control values (e.g. Normal and Abnormal) and titles (e.g. “Brain MRI”).

![Diagram](image)

**Fig. 1.** Prototype showing the segmentation produced for the form *Diagnosis.*

2.2 Algorithm for extracting structural properties

In order to obtain the models described above, we follow the following steps:

1. First, we obtain an inventory of visual styles (layout model). A ranking of the visual relevance of the styles is computed using an entropy function which can be parametrized by the user.
2. Next, a segmentation is performed using a variant of a X-Y tree decomposition. This technique, originally proposed in [7], consists of successively splitting a page by horizontal and vertical cuts on white spacing; the end result being a segmentation of the page into hierarchically related boxes.

3. The information obtained by the segmentation and the layout analysis is used to build the hierarchical structural information, using techniques adapted from [14]. These techniques analyse visual influence areas (which allow us to find which labels and controls fall under a higher-level heading) and positional relations (adjacency and nesting) to find out relations between labels and controls, and in particular, which controls are organized as a grid, described by row and column headers. Alternative approaches can be found in [15, 3, 9, 10].

4. Labels are then analysed using simple techniques to find out workflow relations.

5. Finally, every control is analysed to extract type information. A particularly useful feature is the list of possible values.

The models obtained by this process are encoded using an XML-based internal representation, and form the basis for the semantic enrichment described in the remainder of this paper.

3 Bringing Knowledge to Forms

The main aim of the present work is to enrich data acquisition forms with both knowledge extracted from the forms and existing knowledge in external sources. For this purpose, the first step to be done consists of identifying known concepts from the different elements of the forms, mainly short descriptions associated to form controls. This task is known as the semantic annotation of the form components.

In our scenario, we can take profit from several annotation tools for biomedical texts that have emerged during the last years. Most of them use the Unified Medical Language System (UMLS) since it provides a rich lexicon with a wide coverage over the biomedical domain. Additionally, it contains a potential taxonomy and ontology over the covered concepts. Several works have used successfully UMLS to perform semantic annotation [6, 2]. However, it is worth mentioning that managing this resource is not trivial, as it contains around 2 million of strings associated to 1 million of concepts.

3.1 Annotating Forms

Once forms have been analysed and segmented as described in Section 2, each form control is associated to a text segment describing it (i.e. its labels) and an optional list of values (usually a list of single words). Other interesting textual sections that deserve analysis are form titles and section titles as they provide useful contextual information. All these form elements are subject to semantic annotation.

As previously mentioned, works about semantic annotation are mainly focused on free-text sources. In our case, annotations must be done over a set of very short text sections extracted from the forms. A series of differences can be identified between annotating free-text and data forms. Firstly, there are a lot of concepts from the thesaurus that should be avoided when annotating free text. These concepts are related to highly frequent and ambiguous words such as sex, name and date, as well as words indicating broad categories such as disease, sign, etc. However, when annotating forms these concepts are very relevant for integration issues. Second, semantic annotation of texts use to be oriented toward statistical

4 http://www.nlm.nih.gov/research/umls/
analysis of concepts to find new knowledge from the literature. In this way, approximate annotations are enough to capture interesting correlations. However, forms require very precise annotations for the controls as any error can produce a disastrous effect when processing their associated data.

Another relevant issue about semantic annotation is the way annotations are generated. In MetaMap [2] a series of techniques based on Natural Language Processing are applied to find the noun phrases in texts that fit the best to existing concept descriptions (i.e. lexicon). As a result, MetaMap returns a list of concepts ordered by their similarity w.r.t. the identified noun phrase. Instead, dictionary-based approaches like MWT (Multi-Word Tagger) [11] return the concepts whose descriptions match exactly with a segment of the text.

3.2 Expressing Semantic Annotations

As earlier mentioned, the annotation of form controls will be quite useful in order to classify and define groups of forms and controls. Therefore, it is necessary to include the hierarchical relationships involved by the found annotations in order to enable such a classification. For this purpose, we build a simple ontology module that allocates the annotations and their relationships as follows:

- Each control of the forms will be represented by a uniquely identified concept $A_i$. The UMLS concepts associated to this control, $CUI_1, \ldots, CUI_n$, characterize it by means the following axiom:

\[ A_i \sqsubseteq \exists \text{hasUMLS}.CUI_1 \sqcap \cdots \exists \text{hasUMLS}.CUI_n \]

- Each UMLS concept $CUI_i$ used in the annotations will be related to its semantic types $ST_1, \ldots, ST_k$, with the following axioms:

\[ CUI_i \sqsubseteq ST_1 \cdots CUI_i \sqsubseteq ST_k \]

- Finally, by using a fragment generator for UMLS [8] we add all the is-a relationships involved by the identified concepts. These axioms have the form $CUI_i \sqsubseteq CUI_j$.

3.3 External Domain Ontologies

The previous subsection has presented the link between the protocol attributes and the UMLS sources: with the Metathesaurus by means of the annotation CUIs and with the Semantic Network by means the inherent semantic types of the CUIs. However, this linking may not be enough for complex classifications, since the UMLS semantic network provides an upper level granularity. Domain ontologies will become a key point since they will represent the bridge between the UMLS lexicon and the UMLS semantic network.

The use of medical ontologies like Galen and NCI poses new challenges in this scenario: (1) selection of the proper ontology or ontologies that better characterise the form contents, (2) extraction of the appropriate ontology fragments from the selected ontologies.

In current works about ontology segmentation and modularisation (e.g. [4, 5]) it is usual to define an input set of concepts and properties to express the requirements of the user with respect to the desired module contents. This input set is also referred as input signature as it contains the symbols of the target ontology that will guide the extraction process.

In our application, input signatures are generated from the semantic annotations associated to controls. In order to get the proper symbols of the target ontology we need a

---

mapping between UMLS concepts and the ontology symbols. In our preliminary experiments we have used the NCI ontology because it already provides these mappings in the OWL file. Future work we will be focused on automatically linking other medical ontologies like Galen to UMLS.

3.4 Building the Application Ontology

The main goal of FAETON is to generate an application ontology for semantically describing in an homogeneous way a set of heterogeneous data acquisition forms. This application ontology comprises three layers (see Figure 2), namely:

- **Level 1**: Top level knowledge represented by means the UMLS Semantic Network.
- **Level 2**: The ontology fragments covering the relevant domain knowledge for the annotated protocol attributes.
- **Level 3**: The knowledge coming from the forms. This consists of the annotated UMLS concepts as explained in Section 3.2, and the structural relationships between controls derived from the forms.

![Fig. 2. General FAETON Architecture for the medical scenario](image)

Elements of Level 1 and 2 and the UMLS annotations have been treated in the previous sections. They are included in the application ontology as imported modules. Table 1 shows the possible axioms that can be established from the structural information extracted by FAETON.

4 Prototype and Preliminary Experiments

We have produced a prototype of FAETON, which currently features a full implementation of form segmentation and a selection of semantic enrichment techniques. The prototype has been developed in Python using the GTK+ libraries for the visualisation and analysis of forms. In this section we show the preliminary results obtained with this prototype over a set of forms representing medical protocols.

---

6 NCI Thesaurus properties:
Axiom Pattern | Conditions
--- | ---
\(E_{\text{base}} \sqsubseteq \exists \text{has}_E \text{form}\) | \(E_{\text{base}}\) is the underlying entity on which forms are focused (e.g. Patient). The entity \(E_{\text{form}}\) is identified from the form’s title.
\(E_{\text{form}} \sqsubseteq \exists \text{has}_E \text{section}\) | \(E_{\text{section}}\) is the concept associated to a section of the form represented by \(E_{\text{form}}\).
\(E_x \sqsubseteq \exists \text{hasControl}_E \text{control}\) | \(E_x\) represents the concept of the closest structure to the control represented by \(E_{\text{control}}\), which can be either a section or a form.
\(E_{\text{control}} \sqsubseteq \exists \text{hasM}_E \text{control} \cdot \text{DT}_{\text{control}}\) | \(\text{DT}_{\text{control}}\) is the numeric data type associated to the control represented by \(E_{\text{control}}\).
\(E_{\text{control}} \sqsubseteq \exists \text{has}_E \text{control} \cdot \{E_{v_1} \ldots E_{v_k}\}\) | The control represented by \(E_{\text{control}}\) has associated a set of values \(\{v_1 \ldots v_k\}\) which are represented by the entities \(E_{v_1} \ldots E_{v_k}\).
\(E_{\text{row}_i} \sqsubseteq E_{\text{grid}_{i,1}} \sqcap \ldots \sqcap E_{\text{grid}_{i,n}}\) | Given a grid structure, each entity represented at each row is related to the entities represented by the controls placed in that row. These controls take its concepts from their corresponding header labels.

Tabla 1. Axiom patterns expressing the structural semantics of a form. \(E_x\) represents the concept associated to the form element \(x\).

Combining structural information and semantic annotations. The accuracy of the final application ontology will depend on both the quality of the semantic annotations and the quality of the inferred structures from forms. If controls are incorrectly tagged, then the generated axioms will produce wrong classifications. Whereas if the form’s structure contains errors (e.g. elements incorrectly related) the system will produce both wrong annotations and wrong axioms. Additionally, controls without annotations will be useless for integration and classification issues. In order to refine both the extracted annotations and the structures, the system can assess them to detect flaws and to give experts clues to fix them up. Two kind of flaw detections are currently applied: (1) Identify form elements whose annotations differ from the majority of its nearest neighbours in the form, and (2) associate temporarily each untagged control to a new concept placed under the NCA concept of its neighbours in the form.

Evaluation of Annotations. In the preliminary experiments we have evaluated two methods for annotating the data acquisition forms: MetaMap [2] and MWT [11]. Table 2 shows the coverage achieved by the tested semantic annotators for seven protocols within the Brain Tumours domain in the HeC project. We have manually evaluated the results by considering the following measures: right annotations/incomplete or wrong annotations/Non-tagged controls. Comparing the two annotators, it can be concluded that MWT produces slightly better results in precision, and MetaMap slightly better results in recall due to its flexibility with term variations. Therefore, both annotators could be combined to complement each other (see last column of the table).

To sum up, nearly 90% of the controls have some annotation associated, although some of them are not as precise as required. For example, we have found 7 wrong annotated elements in the section Permanent Sequels of the protocol Follow up Data. This is due to the fact that UMLS does not cover the knowledge needed to express these elements. Thus, further knowledge should be extracted from other sources in order to the able to correctly
annotate these controls. On the other hand, we have found 4 untagged elements that can be placed correctly in the ontology thanks to the their neighbours in the form. Finally, partial annotations should be carefully analysed. Although they contain correct concepts, they do not fully express the semantics of the control in the form. As an example, in the section Patient Status of the Follow-up Data protocol, some of the controls such as Alive with disease and Dead of disease have been homogeneously annotated with diseases concepts, but they not capture the very sense of these controls. Since no concepts exist in UMLS that can cover these types of meanings, these annotations are considered incomplete.

5 Conclusions

This paper has presented a new method for semi-automatically building tailored application ontologies from a set of data acquisition forms. Preliminary results are satisfactory, but some issues remain open as not all the elements of the forms can be semantically annotated due to the limitations of the external resources. Future work should address more complex logical representations for the application ontologies, that is how to combine atomic annotations to provide more powerful representations. Another issue to address in future work is the use of logical queries to perform integration and linking tasks. Finally, new ontological resources need to be included in the whole process in order to cover as much as possible the form contents.

Referencias