Construction of domain ontologies from tables

Aleksander Pivk, Matjaž Gams
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia
aleksander.pivk@ijs.si, matjaz.gams@ijs.si

Abstract

Turning the current Web into a Semantic Web requires automatic approaches for document annotation, since manual approaches will not scale in general. The focus of the paper is on automatic transformation of arbitrary table-like structures into knowledge models, i.e. frames, and on the extension for merging of frames into domain ontologies. The presented work is based on Hurst’s table model and consists of a methodology, an accompanying implementation named TARTAR, and a thorough evaluation. The evaluation showed over 80% success rate of automatic transformation of tables into semantic representations and 100% accuracy in the task of query answering over the table contents.

1 Introduction

Most information on the Web is presented in the form of semi-structured or unstructured documents, encoded as a mixture of loosely structured natural language text and template units. Tabular structures are one of the most popular and commonly used structures when it comes to presenting, visualizing or comparing data. The lack of metadata, which would precisely annotate the structure and semantics of documents, and ambiguity of natural language makes automatic computer processing very complex [1]. The Semantic Web aims to overcome this bottleneck. It relies on annotating resources such as documents by means of ontologies, enabling better and faster discovery of relevant information. Most efforts have been so far devoted to automatic generation of ontologies from arbitrary text, where quite poor results have been obtained. The research concentrates on analysis of tabular structures aiming to exploit their partial structure and cognitive modeling habits of humans. Understanding of table contents requires table-structure comprehension and semantic interpretation, which exceed the complexity of corresponding linguistic tasks.

The central contribution of the paper is twofold: (a) presents a novel method for automatic generation of knowledge models such as frames from arbitrary tabular structures, e.g. found on the Web and encoded in HTML; (b) presents a novel method for creation of more complex knowledge models such as ontologies, by merging generated frames. The outcome of applying these methods is threefold: a knowledge frame for each incoming table, a domain ontology created from all generated frames, and a knowledge base, all encoded in an F-Logic representation language. These structures are directly applicable for different scenarios, i.e. query answering (see Figure 1).

2 Methodological description of the approach

The most comprehensive and complete model for the analysis and transformation of tables, found in literature, is Hurst’s [2], which was also adopted in the thesis. The model analyzes tables along graphical, physical, structural, functional, and semantic dimensions. Our approach stepwise instantiates the last four dimensions.

In the first step, corresponding to the physical dimension, a table is extracted, cleaned, canonicalized, and transformed into a regular matrix form.

In the second step, the goal is to detect the table structure. This is a very complex task, since there is a huge number of table layout variations. The three most important sub-tasks are: (a) determine table reading orientation(s), which is discovered by measuring the distance...
among cells, and hence among rows and columns; (b) dismember a table into logical units and further into individual regions, in a way that regions consist of only attribute or instance cells; (c) resolve the table type, which must belong to one of five pre-defined types. For these purposes several techniques, heuristics, and measures are used, i.e. token type hierarchy, value similarity (regression), value features (character/numeric ratio, mean, variance, standard deviation), data frames, and string patterns.

In the third step, the functional table model (FTM) is constructed. FTM is represented as a directed acyclic graph, which rearranges table regions in a way to exhibit a global path for each individual cell. After finalizing the FTM construction also its recapitulation is carried out, with a goal to minimize the model.

Finally, in the fourth step we deal with the discovery of semantic labels for table regions, where WordNet [7] lexical ontology and GoogleSets service are employed. These semantic labels serve as annotations of FTM nodes and are later also used within outgoing formal structures.

At the end of table analysis and transformation, a frame is generated out of an FTM. The frame makes explicit the meaning of cell contents, the functional dimension of the table, which is comparable to the relational schema, and the meaning of the table based on its structure. Figure 2 depicts a simple table-to-frame transformation, and a corresponding part of a knowledge base.

In general, our method is not bound to a single document type nor domain, hence it can be applied to any table layout description (i.e. html, excel, pdf, text, etc.). This indicates that there are no domain-specific operations and no specific knowledge incorporated in the approach. A domain is chosen by a supervisor in order to ensure the semantic similarity of involved tables, making it possible to create ontologies. Note that only the first transformation step (physical dimension) must be specialized for the incoming document encoding, the remaining steps are independent. This makes the approach very flexible and easily extensible.

The transformation method is implemented within a system named TARTAR (Transforming ARbitrary Tables into fRames), consisting of 15,000 lines of code written in Java programming language. Its sources are freely available at http://dis.ijs.si/sandi/work/TARTAR/. The system presents a component of a multi-agent system OntoGeMS (Ontology Generation Multi-agent System), which is applicable also for automatic construction of domain-dependent sets, consisting of proper Web tabular structures only.

3 Domain Ontology Construction

Tables with different structure that belong to the same domain, all transformed into frames, could be merged into a single domain ontology. In this way, a generalized representation that semantically and structurally describes captured tables would be created. This would most importantly provide an immediate way to generate annotated pages for the exploitation of Semantic Web aware computer agents, improve scalability, and reduce complexity. Figure 3 depicts an initial idea, where as an input serves a set of same-domain tables, that are transformed into frames, and finally merged into an ontology.

![Figure 3: A process of ontology creation by merging different frames.](image)

Our approach to frame matching is multifaceted, which means that we use all evidence at our disposal to determine how to match concepts. In using this evidence we look not only for direct matches as is common in most schema matching techniques, but also indirect matches. The most relevant matching techniques are the following:

- **Label Matching**: this technique depends on WordNet features such as synonyms, word senses, and hypernyms/hyponyms. Besides WordNet also string distance metrics, such as Levenshtein edit-distance,
fast heuristic string comparators, token-based metrics, and hybrid methods are used. These modified measures are particulary useful when name matching is obscured by shortened mnemonic names, abbreviations, and acronyms, which are often found in table headers.

- **Value Similarity**: matching of different object sets is based on value characteristics such as alphanumeric features including length, alpha/numeric ratio, space/nonspace ratio and numeric features such as mean and variance. Gaussian value matching and regression matching allow us to match imprecise but highly correlated value sets, such as price values, or insurance rates.

- **Expected Values**: using constant value recognizers in data frames ensure that finding and matching expected values in value sets provides significant leverage in schema matching. Being able to recognize values such as distances, dates, times, currencies, percentages, etc., helps us match object sets. Data frames recognizers also help distinguish labels from values in tables, decompose or compose value strings for matching, and determine whether value sets are unions or subsets of other value sets.

- **Constraints**: include keys in the table (as well as non-keys), functional relationships, one-to-one correspondences, subset/superset relationships, and optional and mandatory constraints involving unknown and null values. Constraints can be also derived from typed hierarchies and recurrent patterns.

- **Structure**: a matching algorithm was developed that is based on structural context, where features such as proximity, node importance measured by in/out-degree, and neighbor similarity help match object sets.

Once the mappings among frames have been discovered, the merging process begins. Sometimes the match is such that two frames are directly fused by simply merging corresponding nodes and edges. Often, however, merging induces conflicts that must be resolved, where several different approaches for conflict resolution are used. Conflict resolution will not be discussed in detail in this paper due to paper length limits.

Initially, we look for the frames that exhibit the largest possible overlap (as measured by the number inter-frame mappings) with respect to the size of considered frames to be merged. Thereafter frames that overlap the most with the growing ontology are selected individually for merging. Finally, after all frames are incorporated, the outcome is a domain ontology, where the concepts are arranged into a directed acyclic graph, with the arcs representing relations among concepts and also the types of relations. Eventually, a knowledge base is created by formalizing the table contents according to the newly generated formal structure.

<table>
<thead>
<tr>
<th>Accommodation</th>
<th>Location</th>
<th>Type</th>
<th>Period</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Millennium</td>
<td>British</td>
<td>Apartment</td>
<td>Winter</td>
<td>1.26</td>
</tr>
<tr>
<td>Ski Resort</td>
<td>Columbia, Canada</td>
<td>Single Room</td>
<td>Spring</td>
<td>1.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Summer</td>
<td>1.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Autumn</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Winter</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Other</td>
<td>0.90</td>
</tr>
</tbody>
</table>

| Logi Health  | Bangalore | Suite    | Winter | 1.16 |
| Resort       | India     |          | Autumn | 1.06 |
|              |          | Other    | Winter | 1.00 |
|              |          |          | Autumn | 0.70 |

![Figure 4](image.png)

Figure 4: A simple example of the ontology creation process.

Figure 4 illustrates a simple example of the ontology creation process by merging two same-domain tables, where second table is shown in Figure 2. Both tables deal with hotel price information but the naming convention and contents are different. The matching of the frame concepts and object sets are based on several techniques, i.e. Label Matching and Structure help determine the concept matching, Value Similarity and Expected Values apply to price, room type, and period value matching. The domain ontology is presented as a frame, but its concepts are further classified, where for the representation purposes F-Logic notation is used. For example, concept 'Hotel' and 'Accommodation', according to the WordNet, share a common hyponym 'Structure'; values 'Double/Single Room', 'Apartment', and 'Suite' become concepts and are hierarchically organized under a 'RoomType' concept; the same thing happens to season periods. Also note that concepts 'Price' and 'Cost' share the same synset in WordNet, which is in the ontology described by an equivalent operator (≡) among them. In this way a common ontology covering both frames is generated, which enables ontology population, automatic query answering, and annotation.

In some cases, when there is no overlap among tables, or only few table are in the processing loop, a possible deficiency of the final domain ontology is that it could be too general or have only little (or none) common concepts and hence be useless for further exploitation. This is even more probable if we only depend on limited number of (semantic discovery) resources and tech-
techniques. The lack of the specific ontology is possible from the following aspects: its size (could grow fast by adding new frames), poor connectedness (due to unsuccessful merging process), and hence sparse instance values. We suspect that using (a kind of) a domain specific ontology could enhance the resulting ontology, but this is yet to be verified.

4 Empirical results

The empirical evaluation is performed from the following perspectives: efficiency, usability, and applicability.

The efficiency $E$ of the approach is measured according to the portion of correctly transformed tables belonging to two domains, tourist ($E_t$) and geopolitical ($E_g$), also enabling to prove the approach domain independence [5]. The efficiency reached 84.52% ($E = \frac{1}{2}(E_t + E_g) = \frac{1}{2}(\frac{289}{369} + \frac{313}{345})$).

The usability of the approach is shown in terms of the agreement among system-generated frames against manually annotated ones. The analysis showed that the system is getting appropriate formal representations from a structural and semantic point of view in almost 75% of cases and totally identical representations in more than 50% of cases [4].

Approach applicability is shown from two views. By querying the content of tables, formalized according to the domain ontology, and encoded in the knowledge base, it is shown that returned answers are true and complete in all cases [3]. The query execution is enabled by an inference engine OntoBroker [6].

The strict evaluation of the frame merging process is yet to be verified.

5 Conclusion

Most efforts in bridging the gap between the current Web and the Semantic Web are devoted to automatic generation of ontologies from unstructured text, where the results are still quite poor. By limiting the problem domain to tabular structures, we show that the automatic transformation into Semantic Web is feasible and gives much better results than from text, even though some operations exceed the complexity of corresponding linguistic tasks.

In the paper we present novel methods for automatic transformation of arbitrary table-like structures into frames and their merging into more complex, formalized, knowledge models, i.e. ontologies. The frame construction method is based on Hurst’s cognitive table model, which covers five aspects of table analysis, where four of them are stepwise instantiated and implemented within TARTAR system. The approach is based on heuristic algorithms and methods since Hurst’s model is inherently heuristic. The merging method incorporates several techniques for matching concepts and object sets. The evaluation of the approach shows that the method provides good results in term of efficiency and usability and is thus applicable in practice.

There is one interesting future research direction: use of several machine learning techniques in different levels of our approach, i.e. to eliminate (some) heuristics, or to enable better scalability in a merging process.

Acknowledgment

The research was supported by Slovenian Ministry of Education, Science and Sport, and by Marie Curie Fellowship of the European Community program ’Host Training Sites’, mediated by FZI and AIFB at University of Karlsruhe, Germany. Thanks to all our colleagues for participating in the evaluation of the system as well as to the reviewers for useful comments on the paper.

References