Ontology Population from Web Product Information

Damir Vandic
vandic@ese.eur.nl

Lennart J. Nederstigt
nederstigt@appophetweb.nl

Steven S. Aanen
aanen@appophetweb.nl

Flavius Frasincar
frasincar@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, the Netherlands

ABSTRACT

With the vast amount of information available on the Web, there is an increasing need to structure Web data in order to make it accessible to both users and machines. E-commerce is one of the areas in which growing data congestion on the Web has serious consequences. This paper proposes a framework that is capable of populating a product ontology using tabular product information from Web shops. By formalizing product information in this way, better product comparison or recommendation applications could be built. Our approach employs both lexical and syntactic matching for mapping properties and instantiating values. The performed evaluation shows that instantiating consumer electronics from Best Buy and Newegg.com results in an F_1 score of approximately 77%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation languages*

Keywords

Ontology population, product information, key matching

1. INTRODUCTION

Nowadays, e-commerce has become very popular among consumers. According to a recent report from Forrester Research, e-commerce spending in the United States will hit approximately $262 billion this year. This is a 13.4% increase compared to last year, in which the e-commerce spending was estimated to be $231 billion. At the same time, we see the Web doubling in size roughly every five years. To keep up with this growth, several developments based on the ideas of the Semantic Web have been adopted for large-scale use. One of these developments is the Semantic Web vocabulary schema.org.

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577322.

In this paper, we propose a framework that accommodates the recent Semantic Web developments in the e-commerce domain. More specifically, we focus on knowledge extraction from product pages on the Web. Because a lot of research effort has already been invested in the actual extraction of (tabular) Web site data [4], we do not focus on this topic and assume that the crawled raw data can be effectively obtained. The proposed framework is capable of large-scale ontology population with product information in the e-commerce domain using raw tabular product data. The ontology-driven framework aims to create a structured knowledge base of product information. In order to achieve this goal, user-defined annotations for lexical and syntactic matching are employed, which facilitate the two main tasks of our framework, i.e., property mapping and value instantiation. For our knowledge base, we propose the OntoProduct ontology, which defines detailed properties of 24 consumer electronic product classes and is compatible with the well-known GoodRelations ontology for e-commerce [5].

2. FRAMEWORK

There is a lack of detailed ontologies for consumer electronic products. The only (re-)usable upper-ontology is the *Consumer Electronics Ontology* (CEO) [3], but although this ontology includes subclass-of relationships between product classes, product attributes are not available. Therefore, we propose the *OntoProduct* ontology. OntoProduct is fully compatible with the GoodRelations ontology, even though GoodRelations is relatively high-level and does not describe actual product classes and their features.

In total, OntoProduct, as an extended version of CEO, including new properties, product classes, and relations, contains 24 product classes and 270 distinct product properties from the consumer electronics domain. Such an ontology allows for the instantiation of product information with a relatively high-level of detail. Furthermore, OntoProduct requires, through the usage of the Units of Measurement Ontology (MUO) [2], units of measurements to be linked to quantitative values. The reason for this is that in e-commerce many product features are quantitative and use a unit of measurement (e.g., the weight of a product can be given in pound or in kilogram).

We do not focus on the required preprocessing steps, such as HTML table extraction [4] and product duplicate detection [1, 6]. Instead, the presence of product information on the collected Web pages in the form of key-value pairs is assumed. Collecting this data is often trivial, as many Web

Table 1: Property Matching and Value Instantiation results for the optimal parameters.

Process	Precision	Recall	Accuracy	F_1 score
Property Matching	96.95%	93.27%	94.80%	95.07%
Value Instantiation	77.12%	76.09%	62.07%	76.60%

shops already offer product information in tabular form, ordered as key-value pairs. Our framework uses this *raw product data* for instantiating the individual products and their features into the OntoProduct ontology.

We distinguish three important processes for obtaining a populated ontology from raw input data. First, the type of product that is being instantiated is obtained in the *Classification* process. The classes are predefined in the ontology and determine the possible properties of the product. Most Web stores nowadays have some kind of class or category data of each product available. Therefore, this step is considered as optional in this paper.

The subsequent *Property Matching* step is dependent on the classification result (a product class linked to the raw product), the raw product, the sets of ontology properties and classes, and a similarity threshold for the property matching. The goal of this step is to map each raw product key to an ontology property, as preparation for the subsequent *Value Instantiation* step. To achieve this goal, a lexical (for keys) and regular expression (for values) matching score between each key-value pair from the raw product and each ontology property is computed.

Once the class of the raw product has been determined, and its key-value pairs have been mapped to ontology properties, in the last step, the values from the raw product description can be instantiated. Also, a product individual within the proper class is created and correctly associated to each property-value pair. The last processing step consists of a collection of parsers, content spotters, and instantiation tools.

3. EVALUATION

The raw product data was obtained from two different Web sources, i.e., Best Buy and Newegg.com. Each process in the framework is evaluated separately using a golden standard, under the assumption that the product class of a product description is known.

For the evaluation of the framework we use a slightly modified binary classification scheme, as it is not a typical binary classifier problem. For the Property Matching process, a *true positive* (TP) indicates that the framework has mapped a key-value pair to the correct ontology property. Unlike regular binary classification, where a *false positive* (FP) would represent the case that the framework mapped a property that should not have been mapped, in our case, it could also mean that the algorithm mapped the key-value pair to an incorrect property instead. A *true negative* (TN) is a key-value pair that has correctly not been mapped, whereas a *false negative* (FN) represents the case when a key-value pair that should have been mapped to a property, is not mapped. For the evaluation of the Value Instantiation process, we adopt a similar scheme, where we compare the complete set of triples of a product from the golden standard with the generated triples from the instantiation process.

Table 1 shows the results of the Property Matching process and the Value Instantiation process. Our approach achieves a solid performance for the two processes. Although some raw product keys in the test set were not present in the training set, many key-value pairs were still matched with ontology properties. In practice, this means that a semi-automatic approach would only require training the algorithm with a few products from each product class in order to achieve satisfactory performance on Property Matching for all the products in a Web shop.

After analyzing the results in more detail, we found that the regular expressions, in conjunction with the lexical representations, are often capable of correctly mapping key-value pairs to properties in the ontology. For example, the key 'Product Dimensions' is correctly mapped to `ceo:hasWidth`, `ceo:hasHeight`, and `ceo:hasDepth`, demonstrating the usefulness of regular expressions in this context.

Acknowledgments

Damir Vandic is supported by an NWO Mosaic scholarship for project 017.007.142: *Semantic Web Enhanced Product Search (SWEPS)*. Frederik Hogenboom is supported by the NWO Physical Sciences Free Competition project 612.001.009: *Financial Events Recognition in News for Algorithmic Trading (FERNAT)* and the Dutch national program COMMIT.

4. REFERENCES

[1] S. Aanen, L. Nederstigt, D. Vandic, and F. Frasincar. SCHEMA - An Algorithm for Automated Product Taxonomy Mapping in E-commerce. In *9th Extended Semantic Web Conference (ESWC 2012)*, pages 300–314. Springer, 2012.

[2] D. Berrueta and L. Polo. MUO — An Ontology to Represent Units of Measurement in RDF. `http://goo.gl/SwJyq`, 2009.

[3] CEO. Consumer Electronics Ontology — An Ontology for Consumer Electronics Products and Services. `http://goo.gl/vWFpP`, 2009.

[4] C. H. Chang, M. Kayed, R. Girgis, and K. F. Shaalan. A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, 2006.

[5] M. Hepp. GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In *16th International Conference on Knowledge Engineering (EKAW 2008)*, pages 329–346. Springer, 2008.

[6] L. Nederstigt, S. Aanen, D. Vandic, and F. Frasincar. An Automatic Approach for Mapping Product Taxonomies in E-commerce Systems. In *24th International Conference on Advanced Information Systems Engineering (CAiSE 2012)*, pages 334–349. Springer, 2012.