Towards Federated Product Information Search and Comparison Backed by Ontologies

Maximilian Walther

TU Dresden, Department of Computer Science,
Helmholtzstr. 10, 01062 Dresden, Germany
maximilian.walther@tu-dresden.de

Abstract. Product information search has become one of the most important application areas of the Internet. Especially considering pricy technical products, consumers tend to carry out intensive research activities previous to the actual acquisition for creating an all-embracing view on the product of interest. Federated search backed by ontology-based product information representation shows great promise for easing this research process. The topic of the thesis is the development of a comprehensive technique for localizing, extracting, integrating and comparing product information in an automatic way, as well as adopting ontology learning techniques for extending the product information model. Concurrently, intuitive interfaces should allow the user to improve the information quality while only having layman knowledge.

Keywords: Federated Ranking, Information Extraction, Ontology Mapping, Ontology Learning, Product Information Search.

1 Research Problem

As the WWW has become today's most important source of product information, consumers avail themselves of the given possibilities by carrying out product information search on the Internet. Unfortunately, moving the process of information collection into the Internet results in the absence of client counseling which forces consumers to gather product information on their own. Finding and consolidating this information is an ambiguous challenge, as it is distributed all over the Web, thus obliging the consumer to locate and evaluate sources, extract relevant information and integrate it. Additional problems emerge if the user does not prefer a special product in advance or is even missing basic knowledge about the product's domain.

Online product information sources may be divided in vendors, producers and third parties. Vendors or online malls are widely used by consumers as starting points for collecting product information. Producers offer detailed product information on special product websites that mostly include images and semi-structured product details as well as describing texts. Third parties include all sources that do not fit into the first two categories, e.g. information created by average Internet users ("user-generated content") on product forums, social communities, etc.

Most sources hold assets and drawbacks considering information quality. For instance, producer websites provide correct, fresh and verifiable information, but use advertising text for promotion purposes. Lexica like Wikipedia contain goal-oriented and fresh information, but are not immune to biased product characterizations.

As per description there are a lot of criteria for information sources to be called ideal. Table 1 presents all conditions that an ideal information source should fulfill. As no current product information source on the Internet might comply with these criteria, a combination of different product information sources could satisfy them.

Table 1. Requirements for an ideal product information source.

- Completeness	All available information is included.
- Correctness	All included information is correct.
- Freshness	All included information is up-to-date.
- Neutrality	The information is not biased.
- Goal Orientation	All included information is relevant.
- Comparison	Information from similar products is available.
- Verification	Information is backed by corresponding references.

The research work to be done in this Ph.D. aims at enabling federated search using vendor sources and automatically extending the retrieved information with details from semi-structured and unstructured sources. Thus, mechanisms for automating the source retrieval, information extraction and information mapping are to be developed.

Additionally, users may experience a strong boost in product information search if dynamic comparisons of products based on their features would be enabled. The central requirement for offering product comparisons is the use of a unique product terminology that may be modeled as an ontology. Techniques of ontology learning from semi-structured sources are to be developed for automatically extending the terminology. Finally, the user should always have the possibility to intervene for assuring a high quality of the ontology's model (TBox) and instances (ABox).

2 Related Work

As pointed out in the last section, the current work treats several research areas. Considering the automatic localization of product information pages, such as producers' product pages, no research work is known to the author.

However, in the field of information extraction from semi-structured sources a bulk of approaches has been presented in the last decades. Considering information extraction from vendor sites, Shopbots [1] were the first step towards integration of multiple vendors in a federated product search using screen scraping. In [2] Lerman et al. focus more on the general aspect of information extraction from semi-structured sources and offer means for extracting information automatically with only very general assumptions about the structure of the list or table to be analyzed. In [3] Cohen et al. present a similar system. It focuses on learning wrappers for extracting

information from tables and lists. Liu et al. [4] describe a different technique for the same domain, handling shortcomings of [2] and [3]. The developed approach is based on the assumption that so-called data records appear in the same area of an HTML page and are surrounded by similar tags having a common parent tag. In [5] TextRunner is developed, a facts-based search engine using the principles of Open Information Extraction. Sources treated by [5] do not need to be of a special structure. In [6] Wong and Lam present algorithms for feature mining especially applicable for extracting product information from vendor sites. Their evaluation proves the algorithms' feasibility in comparison to other systems.

The mentioned approaches offer powerful means to access semi-structured sources in a very general way. Especially [2], [3], [4] and [6] show admissible results regarding the information extraction from semi-structured sources. This research work focuses on the presentation and comparison of product information and thus on the one hand needs highly reliable extraction results, while the determined information source types on the other hand allow the adaptation of the extraction procedure especially to product information. Thus, algorithms utilizing special peculiarities of the product information domain might show substantial advancements in comparison to the general approaches. Examples for such peculiarities comprise the typical presentation of product information as key-value-pairs, the inclusion of product names or attributes in specification page links or the usage of identical templates for presenting one producer's products.

In the field of ontology mapping and ontology learning, a number of approaches has already been presented as well. Originally, the term ontology learning was introduced by Maedche and Staab [7]. They divide the ontology learning process in five steps, namely import, extraction, pruning, refinement and evaluation. Hearst [8] already published concepts for ontology learning in 1992 without actually using the term itself. In [8] the ontology learning process is accomplished by using so-called Hearst-patterns which consist of English phrases describing special concept relations and variables to be replaced by named entities. E.g., the pattern "NP_0 such as {NP_1, NP_2 …, (and | or)} NP_n" would enable an algorithm to detect sub-concepts of known ontology concepts. In [9] Sanchéz describes a comprehensive approach for creating and extending domain-focused ontologies. The approach is mainly based on Hearst-patterns and Web scale statistics. Patterns may identify new concepts and relations which are examined using public Search Engines. Although creating ontologies for special domains, the approach is applicable for every knowledge domain. Finally, Cimiano [10] presents additional approaches for learning ontologies from textual data and methodologies for the evaluation of learned ontologies.

The presented research works ([7], [8], [9], [10]) offer means to create and extend ontologies from textual, i.e. unstructured sources. So far, few works on exploiting characteristics of semi-structured sources for extending ontologies have been presented. This research work is focusing especially on product information which is mostly presented in a semi-structured way. Thus, specifically adapted algorithms are to be developed that return better results than the application of the general algorithms presented above, as they aim on utilizing peculiarities of product information sources.

3 Contributions

As already mentioned, the aim of the Ph.D. is to develop algorithms for locating, extracting, mapping and comparing product information as well as automatically updating the product information model for meliorating the results of the information retrieval algorithms. Before being able to locate and process product information, some kind of bootstrapping is required that provides a starting point to the follow-up algorithms. Thus, a crawler was created that queries vendors like Amazon for random electronic products. Afterwards, product and producer name are extracted from the vendor's page (e.g. "D60" and "Nikon"), as this kind of information is expected to reside on every vendor's product pages.

For providing a broad overview, the complete workflow is sketched in the following, focusing on the entire procedure rather than algorithmic details.

3.1 Product Information Source Localization

When provided with a product title and producer name, the source localization algorithm is able to retrieve the website of the product on the producer's pages.

First of all, potential producer pages are retrieved using a meta-search engine that operates on a number of rated search engines. It consolidated all search engines' results and finds out frequently returned and highly ranked pages. Based on additional conditions (e.g. domain on blacklist?), the producer's website is identified and its domain is calculated (e.g. "nikon.com"). The meta-search engine is queried again to find potential product pages, this time restricting the results to the producer's domain. Results are ordered again. Then the best-rated HTML page is chosen and, if it is not already the product's specification page (e.g. "http://www.nikon.com/d60"), all links are extracted from it. Using pattern matching for the link texts, found link URLs, etc., the links are ranked and the specification page of the product is found (e.g. "http://www.nikon.com/d60/specs"). The verification of the discovered site is done e.g. by a content keyword check and URL comparisons with known product URLs.

3.2 Product Information Extraction

When provided with a product page (Fig. 1), the following algorithm is able to extract product specifications. On the left side of Fig. 2 the general algorithm is presented.

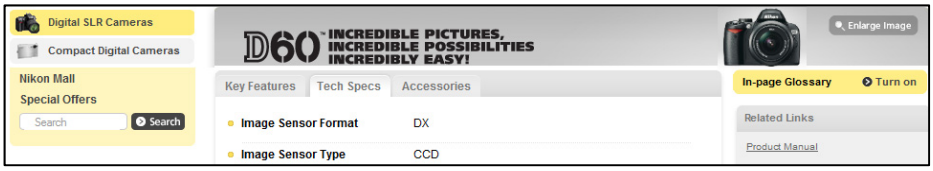

Fig. 1. Nikon's presentation page for the D60 (Source: nikonusa.com).

The algorithm starts by retrieving the page of interest. If no wrapper exists or the existing wrapper is not valid anymore, the algorithm looks for given key examples (product property names contributed by a system user, e.g. “effective pixels”). If key examples are given, the supervised information extraction algorithm may be used. If no examples are given, the unsupervised algorithm on the right side of Fig. 2 is adopted. The unsupervised algorithm creates text clusters, consisting of text nodes from the web page’s DOM tree being located under the same parent elements and residing on the same level of the tree (e.g. “effective pixels, optical zoom, …” or “overview, D60 specs, …”). Then, all known key phrases from the underlying ontology are retrieved and the cluster containing the biggest amount of them is chosen as the product information cluster. The ontology contains information about different product domains including product classes and properties with respective synonyms.

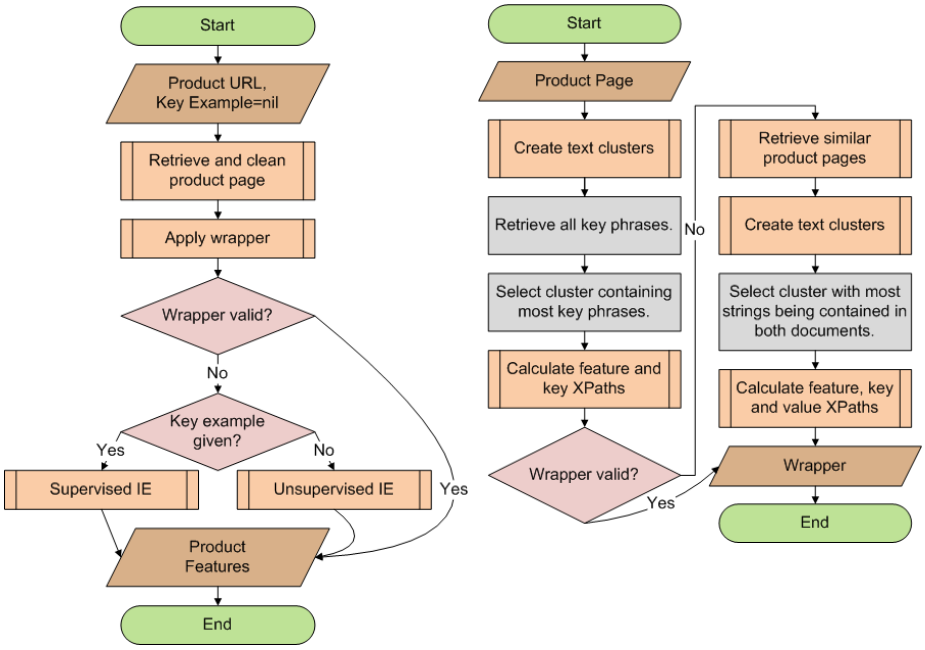

Fig. 2. General information extraction (left) and unsupervised information extraction (right).

If no key phrase could be detected, the algorithm searches for additional product information pages on the producer’s domain having a similar structure (e.g. “http://www.nikon.com/s550/specs”). By comparing text clusters of both pages, the algorithm is able to find out the product information cluster by selecting the cluster that includes the biggest amount of equal terms in both pages. Finally the wrapper is created using XPath expressions and the information can be extracted.

3.3 Product Information Mapping

Extracted product information may be very heterogeneous. Compared to information from other producers selling similar products, the information may vary in the attribute names as well as value structure, type, unit and keywords. The mapping algorithm tries to resolve these inconsistencies like pictured on the left side of Fig. 3.

After storing the extracted information, the product has to be categorized for being able to map its properties. Then the mapping can be executed (Fig. 3, right side).

The mapping algorithm compares each extracted property with all properties of the product category. It uses similarity measures for generating an overall property similarity which are based on the key (e.g. Levenshtein), value structure (e.g. vector), value type (e.g. integer), value unit (e.g. pixels) and value keywords (e.g. "MMC").

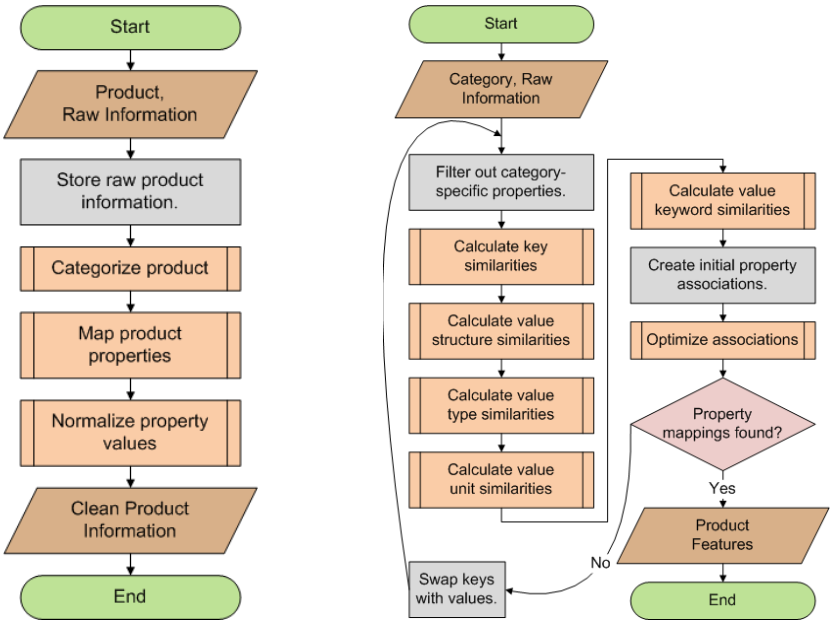

Fig. 3. General information mapping (left) and details on mapping properties (right).

Initial property associations are created (e.g. "effective pixels" » "resolution") which are optimized in several following cycles. Finally a list of mapped product features in a consistent terminology is created. In a last step the property values and their units are normalized (e.g. "9MP" » "9.000.000 Pixels").

3.4 Ontology Learning

The preceding chapters described the process of gathering clean product information. Only those cases were covered, where all product information could be mapped to known concepts and attributes. This chapter shows what happens if the mapping fails.

As every product feature is valuable, no extracted information should be discarded. Thus, the whole feature including its value is saved to a candidates list which is processed by the algorithm in Fig. 4. For frequently found property candidates the system starts an inverse Web search using the property values as search strings and looks for noun phrases (potential property names) residing next to this value. If found noun phrases may be mapped to a known property in many cases, the candidate can be added as synonym candidate. After some follow-up analyses, a new synonym may be added to the ontology's TBox. If no synonym relationship could be detected, and the candidate has been found on a wide set of pages, it is considered to be a new property. Additional analyses decide about the final utilization of each candidate.

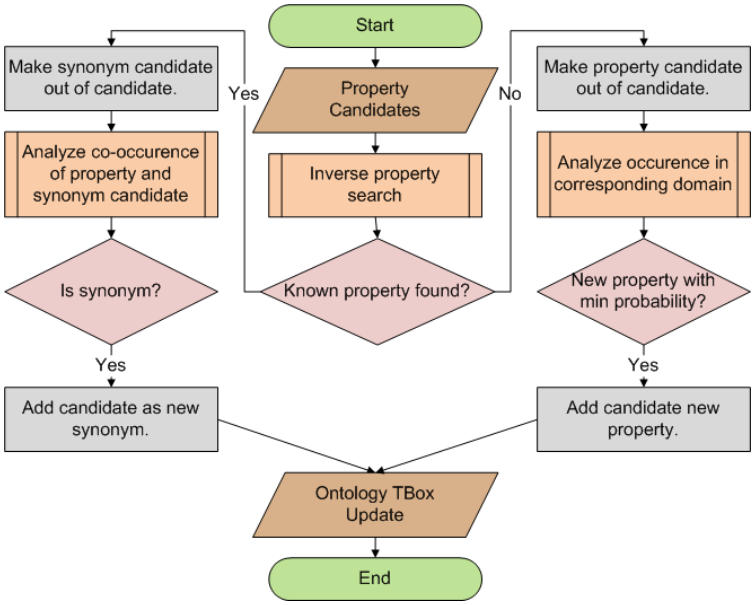

Fig. 4. Classifying and exploiting property candidates.

The described procedure sketches roughly which steps need to be taken for collecting valuable product information in a highly unsupervised way. Due to the limitation of this paper, the user interaction for maintaining a high quality TBox and ABox as well as comparing products based on the extended model are not described.

4 Evaluation

For evaluating the algorithms, a product ontology (that is, an ontology representing information of certain product domains especially adapted to consumers' demands) needs to be developed as well as gold standards that collect a number of product websites, extracted information from these sites and mapped information. The algorithms then would be used to retrieve and map the information automatically for

calculating the success rate of each component. It is planned to evaluate the system in several cycles, each time modifying algorithmic details as well as included thresholds. Additionally, the added value of the ontology learning component should be proven this way, as every evaluation cycle generates better results due to the growing TBox.

5 Work Plan

As presented in the sections above, concrete ideas of the complete process for localizing, extracting and mapping product information already were designed which have been implemented as well. Ongoing evaluations prove the concepts' efficiency.

The ontology learning components are in a very early stage yet. Following the examination of additional related work as well as state of the art technologies, concrete algorithms will be developed for enabling the ontology learning process. However, a basic product ontology has already been designed which allows the mapping of product information especially for the digital camera domain. The ontology will be extended manually for additional domains to build the basis for evaluating the ontology learning component.

After finishing the implementation, cyclic evaluations will be used to extend all existing components and configure included thresholds.

6 References

1. Fasli, M.: Shopbots: A Syntactic Present, a Semantic Future. IEEE Internet Computing 10(6): 69-75 (2006)
2. Lerman, K., Knoblock, C., Minton, S.: Automatic Data Extraction from Lists and Tables in Web Sources. International Joint Conference on Artificial Intelligence, Seattle, USA (2001)
3. Cohen, W.W., Hurst, M., Jensen, L.S.: A Flexible Learning System for Wrapping Tables and Lists in HTML Documents. International World Wide Web Conference, Honolulu, Hawaii, USA (2002)
4. Liu, B., Grossman, R., Zhai, Y.: Mining Data Records in Web Pages. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, (2003)
5. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open Information Extraction from the Web. International Joint Conference on Artificial Intelligence, Hyderabad, India (2007)
6. Wong, T.-L., Wong, W.L.: An Unsupervised Method for Joint Information Extraction and Feature Mining Across Different Web Sites. Data & Knowledge Engineering 68(1): 107-125 (2009)
7. Maedche, A., Staab, S., Ontology Learning for the Semantic Web. In: IEEE Intelligent Systems 16 (2), 72-79 (2001)
8. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. 14th International Conference on Computational Linguistics, Nantes, France (1992)
9. Sanchez, M.: Domain Ontology Learning from the Web. VDM, Saarbrücken (2008)
10. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, New York, USA (2006)