Discovering Multi Terms and Co-hyponymy from XHTML Documents with XTREEM

Marko Brunzel and Myra Spiliopoulou

Otto-von-Guericke-University Magdeburg
{forename.name}@iti.cs.uni-magdeburg.de

Abstract. The Semantic Web needs ontologies as an integral component. Current methods for learning and enhancing ontologies, need to be further improved to overcome the knowledge acquisition bottleneck. The identification of concepts and relations with only minimal user interaction is still a challenging objective. Current approaches performed to extract semantics often use association rules or clustering upon regular flat text. In this paper we describe an approach on extracting semantics from Web Document collections which takes advantage of the semi structured content within XHTML (an XML dialect which can be obtained from traditional HTML documents) Web Documents.

The XTREEM (Xhtml TREE Mining) method uses structural information, the mark-up in Web content, as indicators of term boundaries and for co-hyponymy relations.

1 Introduction

The realization of the Semantic Web depends on the broad availability of semantic resources, often incorporated in ontologies. Ontology establishment is a process demanding substantial human involvement. To facilitate this demanding process, much research has been devoted to (semi-)automated methods for ontology learning and enhancement. Since semantics are expressed by a lexical layer, such methods must address next to the core task of discovering semantics also the prerequisite task of identifying the terms that represent the concepts [W05]. This terminology issue is still only rarely addressed within ontology learning [BMV01, GTA05].

Many methods tackle this issue by exploiting existing resources such as dictionaries, glossaries or database schemata (e.g. [K99, SSV02]). However, dedicated resources for specific application domains are rare and of low coverage, so that the applicability of such methods is limited. Other methods use plain text as input, converting semi-structured content into plain text [FN99, MS00, BOS05], thereupon eliminating the so-called "syntactic sugar". In this paper, we take the opposite approach: We concentrate on the document structure and use it as guide to the content. Our method XTREEM (XHTML TREE Mining) processes Web sites of XHTML documents and extracts multi-terms and co-hyponyms [COH] by relying solely on page mark-up.

XTREEM has several advantages: It requires minimal human contribution and no linguistic resources. It operates on the syntactic structure, which is independent of

R. Nayak and M.J. Zaki (Eds.): KDXD 2006, LNCS 3915, pp. 22 – 32, 2006.

national languages and application-specific jargons. It is not constrained by textual borders like sentences and paragraphs and is thus able to find terms that stand in a co-hyponymy relation even if they rarely appear in the same document. XTREEM is thus a complementary method to conventional text analysis, exploiting information that is traditionally skipped, while using the whole of the Web as information source.

The rest of the paper is organized as follows: In section 2 we discuss related work. In section 3 we introduce XTREEM and describe how it processes Web pages, derives vectors of terms by building a feature space of mark-up tags, clusters these vectors on semantic similarity and derives conceptual labels of correlated terms for them. Section 4 contains our first experiments. The last section concludes our study.

2 Related Work

A recent overview on Ontology Learning from text has appeared in [BOS05]. Here, we concentrate on methods that consider the Web as information source. Cimiano et al discover hyponymy relations by finding examples of Hearst patterns via the Google API and then analyzing the retrieved documents [CPSS04]. However, they treat documents as plain text, ignoring the semantics implicit in the Web structure.

Web Document structure is used in [E04] to build a knowledge base of extracted entities. Nierman and Jagadish [NJ02] study the structural similarity of XML documents, while Dalamagas et al exploit structural similarities in XML document clustering [DCWS04]. Closer to our work are the studies of Kruschwitz [K01a, K01b], where marked up sections of Web Documents are used to learn a “domain model”, because similar mark-up is often used for the representation of similar concepts in Web Documents. Differently from our approach, only local mark-up is exploited: Tag combinations, as reflected in the tree-like structure of (X)HTML documents are not considered. The same holds for the work of Shinzato and Torizawa, who use different tags of HTML documents to find hyponymy relations [ST04]: They consider items of lists but ignoring the role of tag combinations for the representation of semantics.

3 The XTREEM Method

We present the XTREEM method for the extraction of semantic relations through the exploitation of Web Document structure. XTREEM is based on mark-up conventions that are present in almost all Web Documents in the HTML (respective XHTML which can be obtained by conversion) format. Authors use different nested tags to structure pieces of information in Web Documents. We find terms that adhere to the same syntactic structure within an XHTML document and apply data mining to find semantically related terms. These desired semantically related pieces of text are not necessarily physically "co-located" i.e. appearing in the same narrow context window as can be seen in the headings example of table1. Both text elements {Wordnet, Germanet} share a common syntactic structure, the series of HTML tags they are placed in. We aim to use such syntactic structures to infer semantic relatedness.

Table 1. Semantically related terms, located in different paragraphs or separated by other terms

Headings, located in different paragraphs	Highlighted keywords, separated by normal text
… Wordnet Was developed … Germanet Analogous … …	… … there are different important standards for building the Semantic Web. … is RDF. … RDFS adds … whereas OWL is … …

The tasks of XTREEM are depicted in Fig. 2 and described in Section 3.2. Before doing so, we introduce some basic terminology in Section 3.1.

3.1 Web Documents

Web Document D: A Web Document (Web page) is a semi-structured document following the W3C XHTML standard. XHTML is a XML dialect, wherein the former HTML standard has been adopted to meet the XML requirements. Traditional legacy HTML documents are converted to XHTML documents, as it is performed by all popular Web browsers too. The major constituents of XHTML documents are tags (mark-up elements) which enclose text (text elements) as described in the following. In the XML terminology only the terms "element" and "text" are used, but for audibility we will use "mark-up element" and "text element" in the following.

Text Element T: A "Text Element" within a Web Document is a continuous span of text without tags; tags form its border. It can be either (1) a single token without any white space like "Wordnet" in line 8 of Fig. 1, (2) a multi-token term like "Lexical Resources" in line 6 of Fig. 1 or (3) a long sequence of tokens like the texts surrounded by paragraph tags in the same Figure. For our objectives, we are interested in identifying text elements of the first two types: co-hyponyms can be single or multi-token terms. As we will see in the next subsection, XTREEM skips text elements that occur rarely in the collection, so that texts of the third type are filtered out anyway.

```
1  <html>
2  <html><head>
3  <html><head>…
4  <html></head>
5  <html><body>
6  <html><body><h1>Lexical Resources …</h1>
7  <html><body><p>…</p>
8  <html><body><h2>Wordnet</h2>
9  <html><body><p>Was developed …</p>
10 <html><body><h2>Germanet</h2>
11 <html><body><p>Analogous to Wordnet for the English …</p>
12 <html><body>…
13 <html></body>
14 </html>
```

Fig. 1. Document Paths for Text Elements in a XHTML Tree

Mark-up Element: According to the XHTML standard, a "Mark-up Element" is a fixed set of tags which can be used to structure XHTML documents. These tags are interpreted by Web browsers during document rendering.

Document Path P: For each text element a document path, defined as the sequence of mark-up elements from the document root to the text element within the XHTML tree, can be constructed. For example, the heading "Wordnet" in line 8 of Fig. 2 has the document path `<html><body><h2>`.

Document Path = [Mark-up Element Name]*

3.2 The XTREEM Procedure

The XTREEM discovers multi-terms and co-hyponyms for a domain of discourse by mining Web Documents. The XTREEM process encompasses the tasks depicted in Fig. 2. Those tasks extend the conventional process of text mining by a task that builds the text collection itself from the Web. The core of XTREEM are the parallel tasks for Building the Feature Space and Building the Data Space. Briefly, the feature space consists of text elements, while the data space consists of document paths leading to the text elements, i.e. to the features. The tasks of XTREEM are described below.

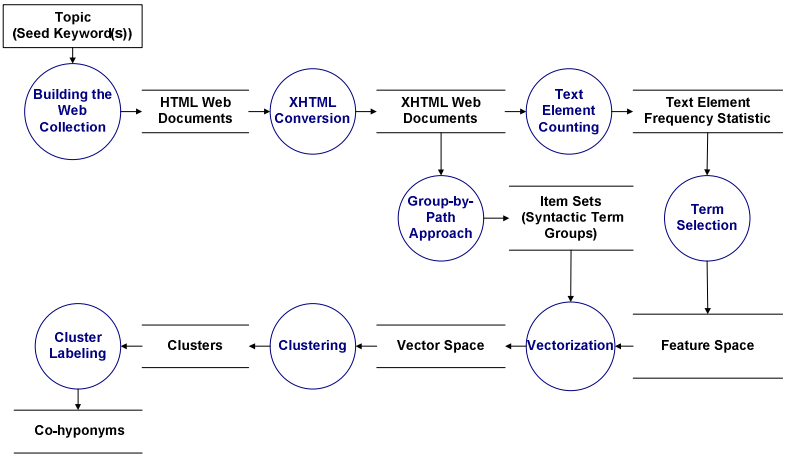

Fig. 2. Data-flow Diagram of the XTREEM procedure

Building the Web Collection: The input to XTREEM is a small set of keywords, the "seed", which characterizes the target domain. Rather than expecting a well-prepared collection of appropriate documents, XTREEM collects documents from the Web by invoking a crawler or by retrieving document references from internet search engine web services.

Hence, the user input to XTREEM is limited to specifying a seed that describes the domain of discourse adequately and guarantees broad coverage. Example seeds may be (1) "Semantic Web" for the Semantic Web, (2) "tourism" for tourism or (3)

"cardiology" for everything associated with heart medicine. More specific terms that characterize the domain, such as "ontology" or "XML" for the Semantic Web or "hotel" for "tourism" are possible but not necessary.

XHTML Conversion: This simple task transforms Web Documents complying to the older HTML standard in XHTML. Moreover, the converter eliminates some existing format errors, thus dealing with malformed Web Documents as well.

Text Element Counting: We create a frequency statistic on all Text Elements. For efficiency, a threshold on the maximum length of text elements can be incorporated to refuse long sequences of text at an early stage. The longer a text element is, the more unlikely that it is a term.

Term Selection: For the feature space construction, the human expert should specify the desired number of features as value of the threshold n. Small values of n are more appropriate if the expert is interested to learn the base terminology for the domain, while large values are more reasonable if the goal is to collect as many terms and multi-terms as possible and acquire co-hyponyms for them.

Due to the low frequency, long text elements (text which is not marked up) have nearly no chance to get into the feature space, while short terms which consist of more than one token and which are used frequently inside the document collection get into the feature space. This has the positive effect, that our approach has an implicit multi term recognition, which otherwise would be a complex Natural Language Processing problem of its own, e.g. the multi token terms "data mining", "Semantic Web" and "Resource Description Format" are recognized by this approach. Web Document specific words such as home, `contact`, `back`, `top`, `site_map` are rejected with help of a domain neutral Web content stopword list.

Group-by-Path Approach: The XHTML tree is traversed by the XTREEM algorithm, for each encountered text element the document path is built. Document path and the text element are stored together for later processing. When the whole document is traversed, we group text elements that have the same document path as its predecessor. E.g. in our example (Fig. 2), Wordnet and Germanet both have `<html><body><h2>` as document path, and, thus become members of the same set of terms `{Wordnet, Germanet}`. Usually, authors use different tags and therefore things separate according to different tags, resulting in different documents paths, therefore several Text Element Sets stemming from one document are possible.

Algorithm 1. The XTREEM Group-By-Path approach on a XHTML document

Input: D
Output: n 'sets of T'

1: for all T in D: create the corresponding P → store P associated with T
2: create the set of n unique P
3: for all n unique P:
    for all T: which T are associated with P → store T
    store set of T
return n 'sets of T'

The resulting sets are filtered: only sets with cardinality greater than min and cardinality smaller than max are further processed. This corresponds to the usage of only those mark-up structures, which are regarded as providing a useful separation. Here precision is preferred over recall.

Next we will contrast how this approach is different to traditional processing of documents.

Traditional processing	XTREEM processing
If a page contains the text elements {Contact, Map, Back, Lexical Resources, Wordnet, Germanet}, one would regard all this terms as a set and model the document as a vector over those terms.	XTREEM processing: According to our approach, which incorporates the structure of the XHTML tree, it is more likely that the text elements form more homogenous term sets, e.g. the 4 term sets {contact, map}, {back}, {lexical structures} and {Wordnet, Germanet}. XTREEM groups text elements with the same document path together, thus resulting in more homogenous instances which facilitate further processing to reveal semantic relations among text elements.

Note that we use element tags only to infer siblingness of elements. We do not consider the meaning of the tags.

The term sets found by this approach can be used for different purposes. In the following we will describe the application of clustering upon these term sets with the goal to eliminate terms which do not belong in such sets, because the semantic relation is of another type than typical inside a set or because there is no semantic relation at all among the set members.

Vectorization: The term sets obtained by the Group-by-Path procedure in step 3 are now vectorized according to the feature space build in step 4. We only process term sets with more than one unique member (for our purpose, finding semantic relatedness, a single term is not useful because for the desired semantic relations at least 2 terms are necessary). Each term set (text element set, transaction) is used to form an instance (vector, record, matrix row). Afterwards, TF-IDF weighting is performed, where IDF refers to the number of vectors, i.e. document paths, rather than to the number of original documents

	Wordnet	Germanet	Euroword net	Semantic Web	…
DocumentA<html><body><h2>	1	1	0	0	…
DocumentB<html><body><table><h1>	1	0	1	0	…
DocumentC<html><body><p>…	0	0	0	1	…
…	…	…	…	…	…

Fig. 3. Exemplary fragment of a Vectorization

Clustering: The objective of the clustering task is the discovery of correlated features, more precisely of co-hyponyms. The Vectorization obtained in the prior step has the tendency to reveal semantic related terms. One way to get these related terms is the application of a clustering algorithm. Association Rules Mining would be an alternative method. For clustering a K-Means algorithm with cosine distance function was applied.

The amount of clusters to be generated can be set on the algorithm. The clustering algorithm creates clusters of instances, which are not useful on our objectives themselves. The desired result (related terms) has to be obtained by the following post processing step.

Cluster Labelling: As we are not directly interested in whether documents paths (with their associated terms) fall into a cluster, we want to see semantic relatedness, expressed through the characteristics of clusters. A "label" is a subset of the features supported by the cluster members, such as the m most frequent features or the features with higher support than a threshold. According to our objectives, these features are semantically correlated, since they appear together in many instances.

4 Experiments

We present here our first preliminary experiments on the discovery of multi-terms and co-hyponyms with XTREEM. The evaluation of an agnostic method like XTREEM is intriguing for the following reasons: First, the establishment of the Web Document collection for a given seed of keywords is part of the XTREEM procedure; hence, we cannot compare with a method that is applied on a well-prepared corpus. Second, only a human expert can decide whether a multi-token object is indeed a multi-term and whether two features are in co-hyponymy relation within an arbitrary domain of discourse. In future work, we intend to test XTREEM against the multi-terms and co-hyponyms of a given ontology, using it as gold standard for a given domain of discourse. In this study, we concentrate on showing the potential of XTREEM in proposing multi-terms and co-hyponymy candidates for the exemplary domain of discourse "Semantic Web, Ontology". For comparison purposes, we have devised a simple agnostic method that discovers correlated features by analyzing the plain text.

4.1 The Web Document Collection

The establishment of the document collection is the first task of the XTREEM procedure. The seed consisted of the keywords "Semantic Web" and "Ontology". We used Google API for retrieving. Under standard settings, Google returns a maximum of 1000 documents per query. To increase the coverage, we have issued for each keyword K in the seed several queries containing the seed and one additional constraint, namely asking for (1) htm documents, (2) html documents, (3) excluding ps and pdf documents and (4) excluding all of the above, so that e.g. php documents could be retrieved. We have thus acquired 4 sets of Web Documents for each keyword. We merged those sets for all keywords, eliminating duplicate documents. The result was a set of 4209 distinct URLs, from which we retrieved 4015 Web

Documents from 2112 domains. From these, we have removed approximatly 10 percent documents that were recognized as non-English language documents.

4.2 Experiment 1 - XTREEM

According to the preprocessing tasks of XTREEM, the Web Documents have been converted to XHTML and the frequencies of text elements over the whole document collections have been counted. We have chosen the 1000 most frequent text elements as features. The Group-by-Path algorithm has processed 49365 document paths, using the threshold values min=1 and max=+infinity. The threshold m on the number of non-zero values per vector was set to 2, so that 6109 vectors were retained.

The vectors have been weighted using TF-IDF and the K-Means clustering algorithm has been applied, setting K=100. We refer to these results as "document path clusters" or "path clusters" for short. Then, each cluster was labeled by its k=10 most frequent features. In Table 2 we show the features in the labels of a selection of three clusters. These clusters were selected because the correlated features in their labels were the easy to interpret. However, many further clusters contained no less informative labels. As can be seen from the table, the cluster labels are quite intuitive. The rightmost one contains 9 publishers where books, journals or articles on the domain of discourse have appeared. The middle cluster contains names of researchers; the two forms of the forename of the last person are remarkable here. The left cluster contains 9 key terms associated with the Semantic Web and with ontologies. Next to the fact that all those terms are related to the domain of discourse, the clear thematic separation of the clusters must be stressed.

Table 2. Clusters of Document Paths (characterized by 10 most frequent features)

ontology	tim_berners_lee	springer
taxonomy	deborah_l_mcguinness	wiley
thesaurus	eric_miller	acm
source	ora_lassila	elsevier
controlled_vocabulary	stefan_decker	ieee
metadata	brian_mcbride	march_april
topic_maps	dan_brickley	mit_press
concept	j_r_me_euzenat	springer_verlag
faceted_classification	jim_hendler	computing
is_a	james_hendler	ieee_computer_society

4.3 Experiment 2 - Application of Conventional Procedure

For comparison purposes, we have designed a conventional text analysis method that has prepared, vectorized and clustered the Web Documents as plain texts. We have used similar constraints: For the feature space, we have selected the 1000 most frequent features. Vectors with less than m=2 non-zero values were removed, resulting in 3089 out of 3829 vectors as input to the clustering algorithm. Again, the K-means with cosine similarity was used, setting K=100. Each of the 100 clusters, hereafter denoted as "document clusters" was labeled with the k=10 most frequent

features in it. The labels of four clusters are shown in Table 3; again, these are the clusters whose labels can be most easily interpreted.

As can be seen, those labels are much more diffuse: The same feature appears in many labels, terms characteristic for the domain are mixed with generic words (e.g. entity and introduction in the second cluster to the right), while the few recognized names of researchers appear together with names of institutions and with some generic names (department of computer science, chair).

We have experimented with this method for larger values of K as well. If K is between 300 and 500, then some homogeneous clusters of similar label quality to those of XTREEM can be found. However, this implies that the human expert must study a much larger number of less interesting clusters to identify reasonable good labels.

Table 3. Clusters of Documents (characterized by 10 most frequent features)

ontoedit	department_of_computer_science	ontology
rdf	university_of_maryland	relation
oil	agents_and_the_semantic_web	abstract
semantic_web	james_hendler	attribute
daml	darpa	conclusion
ontolingua	chair	entity
project	hendler_cs_umd_edu	introduction
semtalk	ian_horrocks	knowledge_base
protege	nature	semantic_web
tool	semantic_web_services	description_logic

4.4 Comparison of the Findings

The differences between the document clusters of the conventional method and the path clusters of XTREEM can be summarized as follows:

- Document clusters are more diffuse, containing features related by arbitrary kinds of semantic relationships.
- The semantic relationships among the features in each path cluster are easily recognizable. This is indicated by the fact that a summarizing concept can be assigned to each of these clusters, serving as parent concept. Hence, the semantic relationship is a sibling-relationship – co-hyponymy: For example, the clusters in Table 2 refer to (1) instruments for the representation of meta-data types, (2) to persons and (3) to publishers.

A posteriori, the supremacy of XTREEM towards simple text analysis is not astonishing: When authors group texts at the same level into itemlists, headlines etc, they are usually motivated by the intention to present sibling concepts in an intuitive way.

For the path-clusters, a human expert can often easily name the implicit but unnamed parent concept and filter out the erroneous terms of the cluster. This requires much less effort than the manual identification of co-hyponyms from groups of loosely correlated features.

The terms in the traditional document-clusters are not semantically unrelated, but the relations are manifold and can bot be eaysls named.

5 Conclusions and Future Work

We have presented XTREEM, an agnostic method for the discovery of semantic relations among terms on the basis of structural conventions in Web Documents. We exploit the interplay of structure and content in Web Documents to find groups of terms which have a certain syntactic structure within a Web Document in common.

Our first results indicate that rerms appearing in the same cluster, i.e. co-occuring in different documents with the same mark-up grouping are good co-hyponymy candidates.

Our method is only a first step on the exploitation of the structural conventions in the Web for the discovery of semantic relations. We will next perform an evaluation of of the extracted terms and co-hyponomy relations. Discovering the coresponding hypernym for the co-hyponyms is a further desireable extension. In our future work we also want to investigate the impact individual mark-up element tags.

References

[BMV01] R. Basili, M. Missikoff, and P. Velardi, Identification of relevant terms to support the construction of Domain Ontologies, ACL-0 1 workshop on Human language Technologies, Toulouse, France, July 2001

[BOS05] P. Buitelaar, D. Olejnik, M. Sintek , Ontology Learning from Text: Methods, Evaluation and Applications, Frontiers in Artificial Intelligence and Applications Series Volume 123, IOS Press, Amsterdam, 2005

[COH] http://www.websters-online-dictionary.org/definition/english/co/co-hyponyms.html

[DCWS04] Dalamagas, T. & Cheng, T. & Winkel, K.-J. & Sellis, T. (2004). A Methodology for Clustering XML Documents by Structure. Information Systems. In press.

[E04] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, A. Yates. Web-Scale Information Extraction in KnowItAll. Proceedings of the 13th International WWW Conference, New York, 2004.

[FN99] D. Faure, C. Nedellec. Knowledge acquisition of predicate argument structures from technical texts using machine learning: the system ASIUM. EKAW '99, volume 1621 of LNCS, pp 329-334.

[GTA05] L. Gillam and M. Tariq and K. Ahmad, Terminology and the Construction of Ontology. Terminology 11 2005 , pp55-81. John Benjamins Publishing Company.

[K01a] Kruschwitz, U. "A Rapidly Acquired Domain Model Derived from Mark-up Structure". In Proceedings of the ESSLLI'01 Workshop on Semantic Knowledge Acquisition and Categorization, Helsinki, 2001.

[K01b] U. Kruschwitz. Exploiting Structure for Intelligent Web Search. Proc. of the 34th Hawaii International Conference on System Sciences (HICSS), Maui Hawaii 2001, IEEE

[K99] V. Kashyap. Design and creation of ontologies for environmental information retrieval. Proc. of the 12th Workshop on Knowledge Acquisition, Modeling and Management. Alberta, Canada. 1999.

[MS00] A. Maedche and S. Staab. Discovering conceptual relations from text. In Proc. of ECAI-2000, pp. 321-325.

[NJ02] Nierman, A. & Jagadish, H.V. (2002). Evaluating Structural Similarity in XML Documents. In Proc. of International Workshop on the Web and Databases, 61-66.

[SSV02] L. Stojanovic, N. Stojanovic, R.Volz. Migrating data-intensive Web Sites into the Semantic Web. Proc. of the 17th ACM symposium on applied computing. ACM press, 2002. 1100-1107.

[ST04] K. Shinzato and K. Torisawa. Acquiring hyponymy relations from web documents. In Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04), pages 73--80, Boston, Massachusetts, 2004.

[W05] H.F. Witschel. Terminology extraction and automatic indexing - comparison and qualitative evaluation of methods. In Proc. of Terminology and Knowledge Engineering (TKE), 2005.