Discovering Semantic Sibling Groups from Web Documents with XTREEM-SG

Marko Brunzel and Myra Spiliopoulou

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

Abstract. The acquisition of explicit semantics is still a research challenge. Approaches for the extraction of semantics focus mostly on learning hierarchical hypernym-hyponym relations. The extraction of co-hyponym and co-meronym sibling semantics is performed to a much lesser extent, though they are not less important in ontology engineering.

In this paper we will describe and evaluate the XTREEM-SG (Xhtml TREE Mining - for Sibling Groups) approach on finding sibling semantics from semi-structured Web documents. XTREEM takes advantage of the added value of mark-up, available in web content, for grouping text siblings. We will show that this grouping is semantically meaningful. The XTREEM-SG approach has the advantage that it is domain and language independent; it does not rely on background knowledge, NLP software or training.

In this paper we apply the XTREEM-SG approach and evaluate against the reference semantics from two golden standard ontologies. We investigate how variations on input, parameters and reference influence the obtained results on structuring a closed vocabulary on sibling relations. Earlier methods that evaluate sibling relations against a golden standard report a 14.18% F-measure value. Our method improves this number into 21.47%.

1 Introduction

The discovery of semantic relations among terms is a crucial task in many applications on the understanding of text and of semantics: ontologies, the backbone of the Semantic Web, rely on making semantic relations explicit. There are many methods for the discovery of vertical, hypernym-hyponym relations. There is less work on the discovery of concepts that stand in a horizontal relation to each other and are the children of a common, not a priori known (and possibly not interesting) parent concept. This horizontal relation can be referred to as co-hyponymy and co-meronymy.

In the field of ontology engineering, there are different approaches for the discovery of semantic relations. There are many approaches which use unstructured plain text (also semi-structured content is converted to plain text) as input [FN99, MS00, and BCM05]. On the other hand, there are approaches using existing structures such as dictionaries, glossaries or database schemas as input [K99, SSV02]. But these

S. Staab and V. Svatek (Eds.): EKAW 2006, LNAI 4248, pp. 141 – 157, 2006.

approaches are practically limited to the rare case that such resources are available. Out method uses semi-structured content as input.

In [BS06], we have presented the first version of XTREEM. In this publication we extend the workshop publication with an improved description of the process, including formalization and evaluation. We will show that the XTREEM-SG method helps to discover groups of terms that indeed stand in sibling relation with higher accuracy than earlier methods. The main contribution of XTREEM-SG is the identification of siblings in a data driven way without any a priory restrictions: No linguistic resources are needed, beyond the input vocabulary.

The paper is organized as follows: In the next section, we discuss related work. In section 3, we present XTREEM-SG. Section 4 and 5 are devoted to evaluation using two golden standard ontologies from the domain of tourism.

2 Related Work

The broad domain of research is *ontology learning*: A comprehensive overview on this subject has appeared recently in [BCM05]. Those approaches are focusing on ontology learning from text. There are also approaches performing *Ontology Learning from structure* [K99, SSV02]: However, these methods use existing database schemas or other conceptualizations as input and are therefore limited to cases where such schemas are available, which is usually not the case. Closer related are studies also discovering semantics on the Web.

Hearst patterns [H92] are used to find relations among terms in text collections. Also co-hyponym relations can be found with this approach. But the disadvantage is that such patterns are rare, the coverage is low, even on big document collections. Cimiano et al also discover (co-)hyponymy relations by finding examples of Hearst patterns via the Google API and then analyzing retrieved content [CS04]. In [P05] instances of WordNet concepts are found within big Web document collections with a rule base mechanism ignoring the mark-up. The document structure is also taken into account for the establishment of a knowledge base of extracted entities from the WWW in [E04]. There are also approaches from the field of ontology learning and ontology enhancement using the WWW [FS02, AHM00].

Kruschwitz [K01a, K01b] uses *mark-up* sections of Web documents to learn a *domain model*. Similarly to our approach, Kruschwitz exploits the mark-up for the representation of similar concepts inside Web documents. However, as opposed to our approach, the tree structure of (X)HTML documents is not incorporated. [ST04] uses also different tags of HTML documents for acquiring hyponymy relations. They only use list *itemizations*. There is no mentioning of using the tree structure of (X)HTML documents in general, where contributions also from other tags than item elements can be expected.

The idea of using structural similarities [ZLC03, B04], including path structures, of XHTML/XML documents is used for several goals, such as clustering documents on structural similarities [DCWS04, TG06, and CMK06]. In contrast we use the path information to infer siblings. The constitution of the paths is not used itself; no comparison with paths from other documents is performed with XTREEM-SP.

3 Finding Sibling Groups with XTREEM-SG

We present the XTREEM-SG method for the extraction of semantic relations through the exploitation of Web document Structure (Xhtml TREE Mining - for Sibling Groups). XTREEM-SG is based on mark-up conventions that are present in almost all Web documents in the (X)HTML format. Authors use different nested tags to structure pieces of information in Web documents, as shown in Fig. 1. We find terms that adhere to the same syntactic structure within an XHTML document and apply data mining to reduce the potential large amounts of candidate sets to find semantically related sibling 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 Table 1. Both text-spans {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
… <h2>WordNet</h2> <p>Was developed …</p> <h2>Germanet</h2> <p>Analogous …</p> …	… <p> … there are different important standards for building the <strong>Semantic Web</strong>. … is <strong>RDF</strong>. … <strong>RDFS </strong> adds … whereas <strong>OWL </strong> is … </p> …

The XTREEM-SG procedure, which aims to organize a given vocabulary of terms into co-hyponym groups, entails pre-processing (the innovative core of the XTREEM-SG approach), processing (clustering) and post-processing (cluster labelling), which are shown in the following data–flow diagram and described in the following sections.

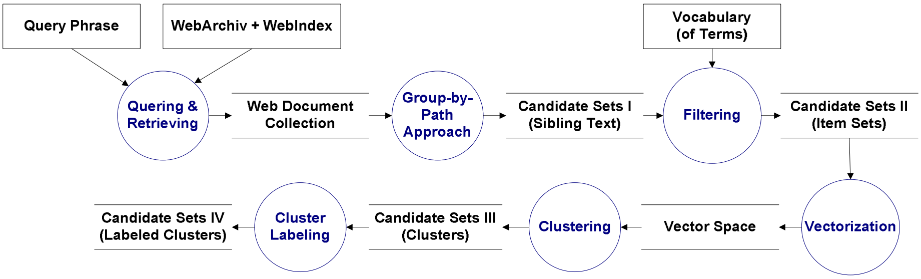

Fig. 1. Data-Flow Diagram of the XTREEM-SG procedure

3.1 The Group-by-Path Operation on Web Documents

First we will describe the operation which represents the core of the overall XTREEM-SG method. We consider Web documents to find sibling relations among terms. Specifically we use the following definitions.

Definition 1: A *Web document* (web page) D is a semi-structured document following the W3C XHTML standard. *D is a tree structure*.

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. Hence, an XHTML document can be seen as a tree, text is represented by leaf nodes and the intermediate nodes are mark-up elements. We use the term text-span to denote the textual contents, the character data sequences of XML elements. The XML elements formed by the tags we will denote as mark-up elements or tags.

Definition 2: Let M be the set of tags supported in the XHTML format and let d be a Web document in XHTML format. A "tag path" p in d is a sequence of tags leading from the root tag element of d to a text-span appearing in d, i.e. p has the form $p=<m_1,m_2,\ldots,m_v>$, where $m_i \in M$ $i=1,\ldots,v$. We use the notation (p,e) to indicate that e is the text-span to which p leads.

By this definition, p is a branch of an XHTML tree; for each m_i, m_{i+1} $(i=1,\ldots,v-1)$ it holds that m_i is the tag surrounding m_{i+1}. A Tag Path is therefore a special kind of Xpath expression. Moreover, a document *D is a collection of pairs* of the form (p,e), where p is a Tag Path and e is the text-span to which p leads.

For example, consider the example document of Fig.2: In line 8, we see the Tag Path "`<html><body><h2>`" leading to the text-span "`Wordnet`".

Let $B=\{e_1,...,e_r\}$ be a set of text-spans. For one document several B can be found by the following Group-By-Path algorithm (Algorithm 1). This is different to traditional "text treatment", where for one text unit (e.g. document, paragraph or

```
<html>
<html><head>
<html><head>…
<html></head>
<html><body>
<html><body><h1>Lexical Resources …</h1>
<html><body><p>…</p>
<html><body><h2>WordNet</h2>
<html><body><p>Was developed …</p>
<html><body><h2>Germanet</h2>
<html><body><p>Analogous to WordNet for the English
…</p>
<html><body>…
<html></body>
</html>
```

Fig. 2. A XHTML Document viewed as a collection of Tag Path - Text-Span Pairs

sentence) a corresponding “Bag of Words” is obtained. Here a b is obtained for each distinct path p of a document d.

Let $A=\{B_1,\ldots,B_t\}$ be the collection which contains the sets of text-spans. The following Algorithm reflects the way A is obtained from D. This grouping operation is the core of the XTREEM-SG procedure.

Algorithm 1: The Group-By-Path algorithm on a XHTML document
Input: Web document D
Output: Collection A of sets of text spans B_i, i=1,...,t

```
extract from D the set $Y=Y(D)$ of (p,e) - pairs, where p is a tag path
according to Def. 2 and e is its target text-span
$A= \emptyset$
let Z be the set of tag paths in Y
for all p in Z
        set $B=\{e \mid (p,e) \text{ in } Y\}$
        insert B to A
end for
return A
```

We group text-spans that have the same tag 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-span sets stemming from one document are possible. Here precision is preferred over recall, since only “valuable” sets of terms will frequently re-occur in a bigger Web document collection. D is now represented as a *collection of text-span sets.*

Summary: The Group-By-Path approach performs a transition of a Web document from a tree, to a collection of tag-path/text-span pairs to a collection of text-span sets.

3.2 The XTREEM-SG Procedure

We now introduce our algorithm XTREEM-SG that takes as input a collection of documents, observing each document as collection of text-span sets. A following clustering is used to perform a “compression”; groups of related terms are the result, such that the terms in each group stand in sibling relationship to each other.

Step 1 – Querying & Retrieving: The XTREEM procedure operates on a *Web Document Collection.* Such a Web document collection is obtained by querying a *Archive+Index Facility* on a *Query Q* with a Web document collection $W=\{D_1,\ldots,D_s\}$ as result, for which Q is satisfied. Q constitutes the domain of interest whereupon semantics should be discovered. It should therefore encircle the documents which are supposed to entail domain relevant content, e.g. “touris*”.

The Web document collection should be big enough to contain manifold occurrences of the desired concepts. The Web document collection is not supposed to be a small manually handcrafted document collection; bigger amounts of web content which have an appropriate coverage of the domain are more desirable. Here, recall is more important than precision. To obtain such a comprehensive Web document collection, alternatively a focused web crawl can be performed; when a vocabulary is given, this vocabulary can also be used to obtain Web document references via the web services of internet search engines.

Step 2 - Group-By-Path: For each $D_i \in W$ with $i=1,\dots,s$ the Group-By-Path algorithm (Algorithm 1, described in section 3.1.) is applied. As result we obtain the collection of text-span sets $H'=\{B_1,\dots,B_u\}$.

Step 3 - Filtering: The aim of the procedure described in this publication is to group a given Vocabulary into semantically motivated sibling groups. Let $V=\{v_1,\dots,v_p\}$ be the vocabulary of terms given as input. For the following steps we only consider all text-spans $e \in B$ which are contained in V. $H''=\{B_1,\dots,B_u\}$ so that for all $e \in B$ it is also true $e \in V$.

In the following we will eliminate all sets b with cardinality of less than two, since only sets containing at least two elements are able to reflect a sibling relation among their elements. $H'''=\{B_1,\dots,B_n\}$, $H'''=\subseteq H''$ where $B_i \in H'''$ if the cardinality of $B_i \in H''>1$ for $i=1,\dots,u$.

Step 4 - Vectorization: Let $F=(f_1,\dots f_p)$ be the Feature Space of Vectorization X. F corresponds to the vocabulary V. X is obtained by creating vectors for each term set $B \in H'''$. TF-IDF [SB88] weighting is applied. X is a 2-dimensinal matrix given by values x_{ij} per term set $1 \leq i \leq n$ and feature $1 \leq j \leq p$. Thus each set of sibling terms is represented by a vector $x_i=(x_{i1},\dots,x_{ip})$ over the feature space.

Step 5 - K-Means Clustering: The vectorization obtained in the prior step has a bias towards sibling related features. Clustering is a method to reduce the potentially large number of instances to a presentable limited number of patterns. 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. A cluster $C \subseteq X$ is a set of vectors. The clustering consist of k clusters $C=\{c_1,\dots,c_k\}$. A cluster can be empty (cardinality=0).

Step 6 - Cluster Labelling: 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.

A cluster label is a set L of frequent features f of a cluster c. A frequent feature is a feature which has an in-cluster-support over a threshold τ.

Definition 3: (in cluster support): Let $C \subseteq X$ be a cluster, where X is the vector space over the instances H''' for the feature space F. Let $f \in F$ denote a feature. The in-cluster-support of a feature f in C is the count of vectors $x \in X$ that contain feature f (i.e. $x_k \neq 0$) divided by the cardinality of C.

Let $L_k=\{f_1,\dots f_v\}$ denote the set of features which have a in-cluster-support $> \tau$ within C_k. According to our hypothesis, the elements of L are siblings to each other.

Let $M=\{l_1,\dots,l_w\}$ be the overall set of generated sibling groups. The cardinality of M may be less than K, since some clusters can be empty and some clusters may not have at least two features with an in-cluster-support $> \tau$.

Summary: XTREEM-SG performs a transition from potentially big numbers of syntactic siblings, obtained from Web documents, to a reduced number of semantic motivated sibling sets, to be presented to an ontology engineer.

4 Evaluation Methodology

There is an ongoing discussion on how evaluation of ontology learning can be performed. Despite that golden standard evaluation can be criticized; we will compare the automatic obtained results against reference semantics. The measured quality is not easily comparable (over different references), but it can help to show tendencies.

Our evaluation objective is: "How good does the method perform on structuring a given vocabulary into co-hyponym groups". We evaluate against golden standards, i.e. ontologies that deliver both the vocabulary (the terms) and the co-hyponymy relations among them. Our goal is to find those relations. The evaluation of sibling relations is performed in [CS05] with the *average sibling overlap* measure. We will compare our results on this measure.

First we investigated how much sibling characteristics are present in the instances obtained by the traditional Bag-of-Words vector space model. Further we used markup of the Web documents, without the path grouping. This approach [K01a] is also based on text-spans created by Tag boundaries.

There are different influences on the results which can be produced with XTREEM-Group-By-Path. First, we vary the documents used as input. We also vary the number of clusters, the cluster labelling thresholds and the required support of the features. The evaluation is performed against two reference ontologies.

It is stressed here that the objective of the evaluation is not the reconstruction of the complete hierarchy, i.e. the naming of the hypernym for each co-hyponymy set. In fact, XTREEM-SG is meant to discover co-hyponym sets, for which the hypernym may or may not be a priori known.

4.1 Description of Experimental Influences

Evaluation Reference: The Evaluation is performed on two golden standard ontologies (GSO), from the tourism domain. The concepts of these ontologies are also terms, thus in the following the expressions "concepts" and "terms" are used interchangeably. The "*Tourism GSO*"[1] contains 293 concepts grouped into 45 sibling sets; the "*Getess annotation GSO*"[2] contains 693 concepts grouped into 90 sibling sets.

There are three Inputs to the XTREEM-SG procedure described in the following:

Input(1): Archive+Index Facility: We have performed a topic focused web crawl on the "tourism" related documents. The overall size of the document collection is about 9.5 million Web documents. The Web documents have been converted to XHTML. With an n-gram based language recognizer non-English documents have been

[1] http://www.aifb.uni-karlsruhe.de/WBS/pci/TourismGoldStandard.isa

[2] http://www.aifb.uni-karlsruhe.de/WBS/pci/getess_tourism_annotation.daml

filtered out. The documents are indexed, so that for a given query a Web document collection can be retrieved.
Input(2): Queries: For our experiments we consider three document collections which result from querying the Archive+Index Facility. The constitution is given by all those documents adhering to Query1 - "touris*", Query2 - "accommodation" and by the whole topic focused Web document collection reflected by Query3 – "*".
Input(3): Vocabulary: The GSO's described before, are lexical ontologies. Each concept is represented by a term. These terms constitute the vocabulary and the feature space.

The overall XTREEM procedure is constituted of pre-processing, processing and post- processing:

Procedure(1): Pre-Processing method: For the evaluation of the Group-By-Path sub procedure we will contrast our Group-By-Path (GBP) method with the traditional Bag-Of-Words (BOW) vector space model and against the solely usage of mark-up (MU) [K01a]. The BOW is the widespread established method on processing of textual data, while MU is a rather new approach which also incorporates the mark-up of Web documents. The variation of these influences is object of our Experiment 1 and Experiment 2.
Procedure(2): Processing – Cluster Number: Each data set (vectorization) is processed by a K-Means clustering with different numbers of clusters to be generated, ranging from rather small to rather big numbers of clusters. For K we used values of 50, 100, 150, 200, 250, 500, 750 and 1000. These numbers encircle the range of numbers of clusters which are appropriate to be shown to a human ontology engineer. This variation is undertaken on all Experiments with exception of Experiment 1.
Procedure(3): Post-Processing – Cluster Labeling Support Threshold: The generated clusters are afterwards post-processed by applying the support threshold cluster labelling strategy. The support threshold is varied from 0.1 to 0.9 in steps of 0.1.

In our experiments we found that some of the terms of the vocabulary are never or very rarely found on rather big Web document collections. E.g. one reference contains the errors "Kindergarden" instead of the correct English "Kindergarten". To eliminate the influence of errors in the reference, we also vary the **Required minimum feature support**. The support is given by the frequency of the features (terms) in the overall text of the Web document collection. We used minimum support thresholds from 0 (all features are used, nothing is pruned) to 100000 (0, 1, 10, 100, 1000, 10000, 100000). When the support is varied, only those features of the vectorization and of the reference fulfilling these criteria are incorporated into the evaluation.

4.2 Evaluation Criteria

Each of the golden standard ontologies delivers a number of reference sets of terms in co-hyponymy relation. Each run of XTREEM-SG delivers a number of term clusters that are suggested as potential co-hyponyms. Intuitively, one would compare each of the suggested clusters against each of the reference sets, select the best match and then count the number of matches; clusters without match and reference sets for which no match was found would be observed as false positives or false negatives. However, the identification of a “best match” is not straightforward, nor is the selection of a “single” best match the most appropriate evaluation strategy.

To highlight this, consider an extreme but not unrealistic example: All towns of the world are co-hyponyms. Within this enormous reference set, there are many subsets of co-hyponyms in different contexts: all towns of the same country, all towns across the same river, all towns close to the same airport, all towns where the same language is spoken etc. Discovering that two towns are co-hyponyms in some of these contexts is more likely than assessing co-hyponymy for 3, 4 ..., n towns. Finding all co-hyponyms of some reference set is more challenging and finding the complete set of co-hyponym towns (all tourist towns of the world) from text analysis is quite improbable. At the same time, finding out that London and Tokyo are co-hyponyms according to several reference sets (capital cities, cities with airport, cities in island countries, very large cities) is information of interest for each of those reference sets. Therefore, we need for our evaluation a measure on the contribution of each cluster of terms to the reference sets of co-hyponyms. To this purpose, we use the “average sibling overlap” measure proposed by Cimiano and Staab in [CS05].

The average sibling overlap SO_{average} is used in [CS05] to compare sets of siblings generated by an automated approach to reference sets of siblings, as delivered by an ontology. This measure is then used to compute F-measure values.

Definition of Average Sibling Overlap (according to [CS05])**:** Let A and B be two sets of co-hyponymy relations where a co-hyponymy relation is a set of sibling terms. Typically one of e.g. A comes from a reference while the other e.g. B is produced by a semi-automated approach. The average Sibling overlap $SO_{\text{average}}(A,B)$ between a set G_A (e.g. co-hyponym groups) of sets H_A (e.g. concepts/terms) from source A to another set G_B of sets H_B from source B, is calculated as follows: For each H_A the relative overlap with each H_B is calculated. This relative overlap is the number of terms present in both sets, divided by the number of unique terms from both sets together. For each H_A, the H_B with the maximal relative overlap value is identified. Over all those maximal values the mean is calculated, representing the average Sibling overlap $SO_{\text{average}}(A,B)$. $SO_{\text{average}}(B,A)$ is calculated accordingly. The F-Measure on average sibling overlap (FMASO) combines both values:

$$\text{FMASO} = \frac{2 \cdot SO_{\text{average}}(A,B) \cdot SO_{\text{average}}(B,A)}{SO_{\text{average}}(A,B) + SO_{\text{average}}(B,A)}$$

5 Experiments

In the following we will show the results obtained from the experiments. Table 2 shows the number of documents which adhere to a certain Query. This corresponds to the size of the Web document collection which is processed by the subsequent following processing steps.

Table 2. Number of Web documents returned by the Web Archiv+Index Facility for the Queries used in the evaluation experiments

Query Name	Query Phrase	Number of Documents
Query1	"touris*"	1,468,279
Query2	"accommodation"	1,612,108
Query3	"*"	9,437,703

5.1 Experiment 1: The Sibling Semantics of the Group-by-Path Method

In our first experiment we want to investigate how much sibling semantics are captured by the Group-By-Path (GBP) method in contrast to the traditional Bag-Of-Words (BOW) vector space model and against the solely usage of mark-up (MU) [K01a]. To do so, the GBP step of the XTEEM-SG procedure was changed to BOW and MU.

We evaluated the collections of sibling sets for the following constellations of Query (Query1, Query2, and Query3); Pre-Processing Method (BOW, GBP and MU) against the reference sibling sets (GSO1 and GSO2) of two golden standard ontologies. Since the two ontologies have different numbers of terms, each constellation of Web document collection results in a different number of vectors after the Vectorization.

Table 6. Results of FMASO for different constellations of queries, pre-processing methods and references; column 2 (cardinality=0) and column 3 (cardinality=1) show the number of candidate sets which are filtered out because they are not true sets

Constellation (Query,Method,Reference)	Number of Sibling Sets (separated according to the cardinality of the set)			FMASO
	Card.=0[3]	Card.=1[4]	Card.>1	
Query1-BOW-GSO1	18,012	29,104	**1,421,163**	**0.206**
Query1-GBP-GSO1	12,589,016	817,289	**222,037**	**0.247**
Query1-MU-GSO1	794,325	343,891	**323,428**	**0.235**
Query2-GBP–GSO1	12,712,295	1,034,741	**293,225**	**0.252**
Query3-GBP-GSO1	63,049,135	3,485,782	**924,045**	**0.256**
Query1-BOW-GSO2	19,399	18,494	**1,430,386**	**0.160**
Query1-GBP-GSO2	12,478,364	831,969	**318,009**	**0.208**
Query1-MU-GSO2	753,657	332,973	**375,014**	**0.199**
Query2-GBP–GSO2	12,677,515	988,944	**373,802**	**0.196**
Query3-GBP-GSO2	62,572,763	3,559,356	**1,326,843**	**0.229**

From these results can be seen that for GSO1 the FMASO is higher for all constellations where the GBP method was involved (0.247, 0.252, and 0.256) compared to the alternative methods (0.235, 0.206). Though it was never claimed that the traditional BOW method is strong on capturing sibling semantics it resulted in the weakest results on capturing sibling semantics. For GSO2 the result of Query1 and MU is slightly better than the result of Query2 and GBP, though for the same Query, GBP performs again best. BOW performs gain worst.

[3] No match with given vocabulary.

[4] Single match with given vocabulary, no true sets, will not be processed.

Conclusion: The candidate sets (text-span siblings) generated by the Group-By-Path pre-processing reveal a stronger sibling characteristic than the traditional BOW vector space model. Though it was never claimed that BOW has significant sibling characteristics, it can be concluded that the GBP method does not capture sibling semantics by chance; the path information of Web document structure can be used to infer semantics.

5.2 Experiment 2: Sibling Semantics Obtained from Labelled Clusters

Additionally to the intermediate sibling sets evaluated in Experiment 1 also a K-Means clustering was performed for Query1 and the pre-processing methods (BOW, MU, and GBP). The cluster labelling threshold was set to τ=0.2.

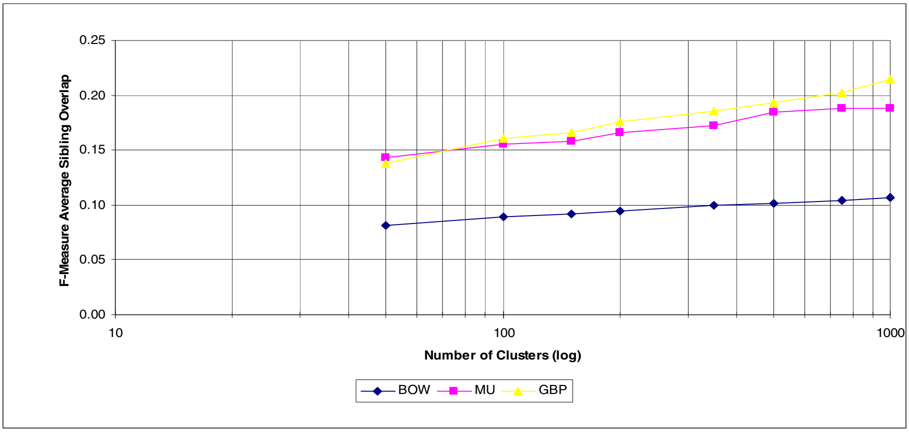

Fig. 3. FMASO on different K for different Pre-Processing methods (Query1, τ=0.2, GSO1)

Fig. 3 shows that the Group-By-Path approach performs also better when the sibling sets are clustered. There is a general trend to better results when higher number of clusters are generated, which is objetive of experiment 4. The analogous diagram for GSO2 (which is not shown) reveals the same finding but with lower values for all three approaches.

The difference between MU and GBP seems to be marginally. A possible explanation for this circumstance is, that when instances are created with MU, those instances have a big overlap with the instances created by GBP since they stem from the same Web mark-up created text-spans, caused by the rather small vocabularies used, which only allow for a fraction of the terms occurring in the Web document collections. Here, also the insensitivity of the FMASO may be responsible for the low measured difference: Whereas siblings not stated by the reference are regarded as false to the same extent as truly not sibling related nominations. This could only be solved by a human expert evaluation. On experiments judged by a human expert one can say that the strong sibling character caused by GBP is recognizable compared to MU.

Conclusion: These results are compatible with the Conclusions of Experiment 1 and verify our hypothesis that GBP performs well on capturing sibling semantics.

5.3 Experiment 3: Varying the Cluster Labelling Threshold

For Query1 in combination with GBP we varied the cluster labeling support threshold from τ=0.1 to τ=0.9 in steps of 0.1 resulting in the following chart of Figure 4. The best results have been obtained on the biggest used number of clusters (K=1000) in combination with a cluster labeling strategy using a support threshold of τ=0.2, resulting in an FMASO of 21.47% (Fig. 6). The results on GSO2 are (again) worse than the results for GSO1. The best FMASO of 15.88% for GSO2 is obtained on K=1000 and τ=0.3. The second reference ontology is more than twice as big as the first one, so structuring the vocabulary into sibling sets may be more difficult. We suspect that this has to do large size of the ontology. There are many terms, but not all sibling relations which can be found in the world, are explicit in the reference. For GSO2 we show only the diagramm of Experiment 5 (Fig. 8), for all other diagrams of GSO2 the charts are compatible to the findings obtained for GSO2.

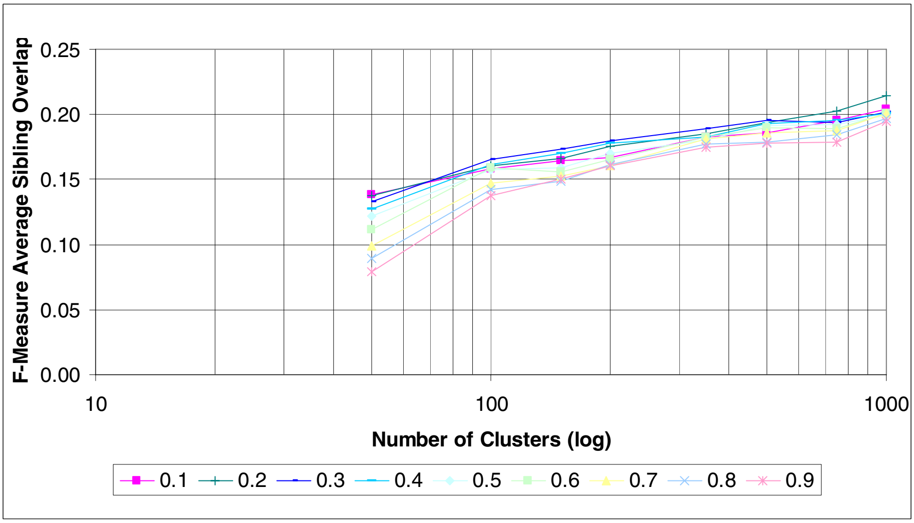

Fig. 4. FMASO on different K for different τ (Query1, GSO1)

5.4 Experiment 4: Observing the Number of Clusters

As already shown in Experiment 2 and Experiment 3, with an increasing number of clusters generated, the F-Measure on average sibling overlap increases too, but with saturation (the number of clusters is logarithmic scaled).

The increasing number of clusters has the drawback that the amount of information, which is compared against the reference, increases too. For automatic evaluation this is not a problem, but if a human would inspect the generated data, this is relevant. We additionally count the number of features appearing in the cluster labels for all clusters of a clustering. This sum of terms/features in the cluster labels over all clusters of a clustering we will refer to as “Sum of Features in Cluster Labels” (SOFICL). The number of distinct features/terms used for cluster labeling we will refer to as “Number of distinct Features in Cluster Labels” (NODFICL).

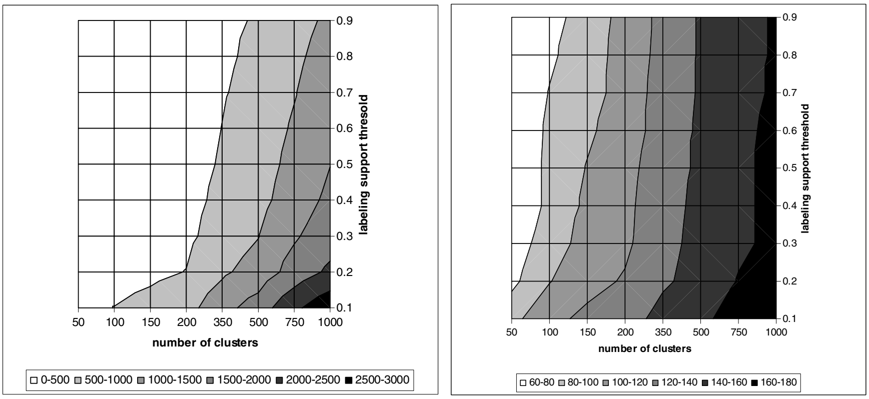

Fig. 5 (SOFICL) and **Fig. 6.** (NODFICL) for different K and τ (Query1, GSO1)

As Fig. 5 and Fig. 6 show, the values of SOFICL and NODFICL are correlated, with an increasing K (and decreasing τ) more terms are used for labeling in the sum (SOFICL) but also more distinct terms (NOFICL) are incorporated into the labeling. An increasing NOFICL means that a bigger share of the vocabulary is indeed incoprorated in the results. The lower right corner of Fig. 6 shows that for high numbers of clusters 160-180 out of 293 of the features are used for cluster labeling. The circumstance that on lower numbers of cluster many terms/features are not used for labeling may be caused by the different support the features have within the vectorization. The low frequent features may have never the chance to be frequent enough for cluster labelling. But this is rather a problem for automatic evaluation. In semi-automatic settings, one can present a ranked list of features for a cluster to the user, who is free to choose also lower frequent features.

5.5 Experiment 5: Variations on the Web Document Collection

Now we will investigate the influence of the processed Web document collection. Since the Web document collection is given by a Query, we will apply the XTREEM-SG procedure for Query1 (“touris*”), Query2 (“accommodation”) and Query3 (“*”; whole topic focused web crawl).

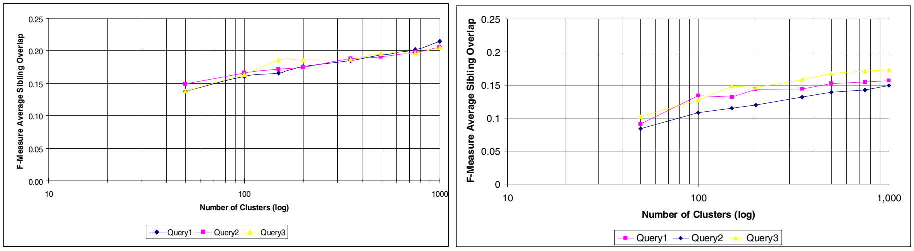

Fig. 7 and **Fig. 8.** FMASO on different K for different Queries (τ=0.2) for GSO1 and GSO2

The results depicted by Fig. 7 show, that there are no big differences on the results measured by the FMASO regarding the choice of a domain constituting query for GSO1. This is in so far a positive finding, that the domain expert should only roughly state which topic he is interested in. While doing so, minor varyations do not lead to significantly worse or better results. The results are quit stable. For GSO2 Query1 ("touris*") and Query3 (big tourism focused web crawl) shield the best results. An explanation for this may be, that Query1 and Query3, which are more broad than Query2 ("accommodation"), encircle more sibling semantics which have also been encoded in the GSO2.

5.6 Experiment 6: Variations on the Required Support

For Query1 we set a threshold on the required support of terms in the Web document collection. This means, terms/features which are rather weakly support are more and more ignored; both for cluster labeling as well as in the reference sets.

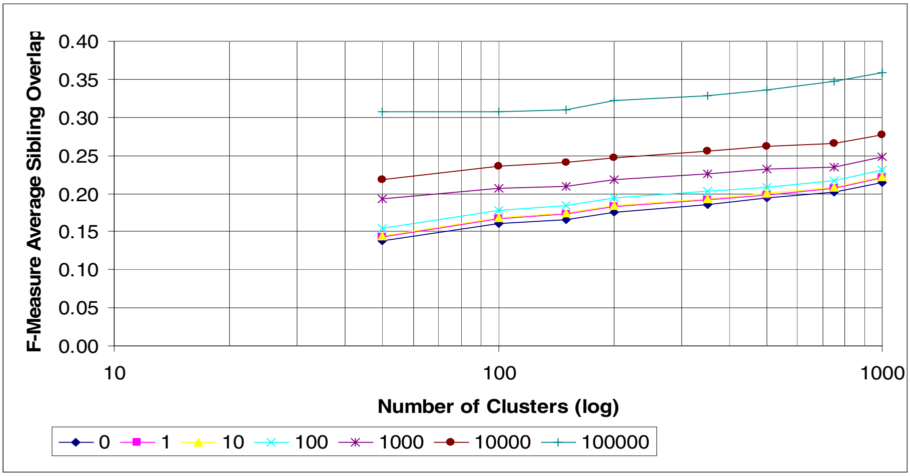

Fig. 9. FMASO for different frequency support levels (Query1, τ=0.2, GSO1)

Fig. 9 shows that while only observing frequent terms, better results on FMASO are shown. With a required minimum support also errors in the reference are smoothed. This is relevant in so far that the relatively low F-Measure values given by our and by other approaches on ontology learning are also caused by "not perfect golden standards", the parts of the reference which are supported by real world data are found reasonably well.

5.7 Conclusion of Evaluation

The application of a Group-By-Path pre-processing with a following K-Means-clustering processing enable to reduce the initial candidate sets significantly by retaining most of the quality measured by the F-Measure on average sibling overlap.

In [CS05] is described, that Cimiano and Staab have obtained average sibling overlap F-Measures from 12.40% to 14.18% on the tourism GSO. With these results, they realized a significant improvement in contrast to Caraballo's method [C99], which gave a sibling overlap F-Measures of 8.96%. We can get good results on this evaluation measure. Our best result gives an F-Measure of 21.47% using a K-Means clustering with 1000 clusters and labeling the clusters by using all features which have a support within the cluster of 20%. This is a significant improvement and confirms that the XTREEM approach delivers good results for mining co-hyponym semantics.

The amount of clusters influences the abstraction forced on the constitution of the resulting sibling groups. For real world settings the expert may decide to handle the tradeoff between reachable quality and the amount of generated information according to his objectives of how detailed the semantics should be. This golden standard evaluation does not capture this aspect, but this can be seen by manually inspecting the results. Cimiano and Staab reported that the results of their approach get better quality valuation by a human expert inspection; the same holds true for the results obtained with XTREEM-SG. The found sibling groups are surprisingly meaningful. On the other side this is not astonishing, the results are based on many thousands, often hand crafted, manifestations on the WWW.

6 Conclusions and Future Work

We have presented XTREEM-SG, a method for the discovery of semantic sibling relations among terms on the basis of structural conventions in Web documents. XTREEM-SG processes Web documents collected from the WWW and thus eliminates the need for a well-prepared document corpus. Furthermore, it does not rely on linguistic pre-processing or NLP resources. So, XTREEM-SG is much less demanding of human resources.

We investigated how variations on input, parameters and reference influence the obtained results on structuring a vocabulary on sibling relations. The reported results from the literature of an F-measure on average sibling overlap regarding a golden standard evaluation of 14.18% are improved by our approach to 21.47%.

Our method is only a first step on the exploitation of the structural conventions in Web documents for the discovery of semantic relations. In our future work we want to investigate the impact of individual mark-up element tags like <p>, <li>, and <dt> on the results. Discovering the corresponding hypernym for the co-hyponyms is a further desireable extension. We are also interested in minimizing the number of clusters which have to be inspected to find co-hyponym relations.

References

[AHM00] E. Agirre, O. Ansa, E. Hovy, and D. Martinez. Enriching very large ontologies using the WWW, *In Proc. of the Workshop on Ontology Construction ECAI-2000*

[B04] D. Buttler. A short survey of document structure similarity algorithms. *In Proc. of the International Conference on Internet Computing*, June 2004.

[BCM05] P. Buitelaar, P. Cimiano, Bernardo Magnini, Ontology Learning from Text: Methods, Evaluation and Applications, Frontiers in Artificial Intelligence and Applications Series Volume 123, IOS Press, Amsterdam, 2005

[BS06] M. Brunzel, M. Spiliopoulou. Discovering Multi Terms and Co-Hyponymy from XHTML Documents with XTREEM. *In Proc. of PAKDD Workshop on Knowledge Discovery from XML Documents (KDXD 2006)*, LNCS 3915, Singapore, April 2006

[C99] S. Caraballo, Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proc. of the 37th Annual Meeting of The Association for Computational Linguistics ACL*

[CMK06] I. Choi, B. Moon, H-J- Kim. A Clustering Method based on Path Similarities of XML Data. *Data & Knowledge Engineering*, Feb. 2006

[CS04] P. Cimiano and S. Staab. Learning by googling. *SIGKDD Explorations,* 6(2):24-34, December 2004.

[CS05] P. Cimiano, S. Staab. Learning concept hierarchies from text with a guided hierarchical clustering algorithm. *Workshop on Learning and Extending Lexical Ontologies at ICML-2005*, Bonn 2005.

[DCWS04] T. Dalamagas, T. Cheng, K. J. Winkel, T. Sellis, Clustering XML documents using structural summaries, *In Proc. of the EDBT Workshop on Clustering Information over the Web (ClustWeb04)*, Heraklion, Greece, 2004

[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. *In Proc 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, LNCS 1621

[FS02] A. Faatz, R. Steinmetz, Ontology Enrichment with Texts from the WWW, *In Proc. of the First International Workshop on Semantic Web Mining, European Conference on Machine Learning 2002*, Helsinki 2002

[H92] M. Hearst, Automatic acquisition of hyponyms from large text corpora. *In Proceedings of the 14th International Conference on Computational Linguistics*, pp. 539–545, (1992).

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

[K01b] U. Kruschwitz. Exploiting Structure for Intelligent Web Search. *In 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. *In 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.

[P05] M. Pasca. Finding Instance Names and Alternative Glosses on the Web: WordNet Reloaded. *In Proc CICLing-2005*, Springer LNCS 3406, 2005.

[SB88] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Information Processing & Management, 24(5):513-523, 1988.

[SSV02] L. Stojanovic, N. Stojanovic, R.Volz. Migrating data-intensive Web Sites into the Semantic Web. *In 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.

[TG06] A. Tagarelli, S. Greco. Toward Semantic XML Clustering. *6th SIAM International Conference on Data Mining (SDM '06)*. Bethesda, Maryland, USA, April 20-22, 2006

[ZLC03] Z. Zhang, R. Li, S. Cao, and Y. Zhu. Similarity metric for XML documents. In Proc. of the Workshop on Knowledge and Experience Management, October 2003.