Structuring Domain-Specific Text Archives by Deriving a Probabilistic XML DTD

Karsten Winkler* and Myra Spiliopoulou

Leipzig Graduate School of Management (HHL), Department of E-Business
Jahnallee 59, D-04109 Leipzig, Germany
{kwinkler,myra}@ebusiness.hhl.de
http://ebusiness.hhl.de

Abstract. Domain-specific documents often share an inherent, though undocumented structure. This structure should be made explicit to facilitate efficient, structure-based search in archives as well as information integration. Inferring a semantically structured XML DTD for an archive and subsequently transforming its texts into XML documents is a promising method to reach these objectives. Based on the KDD-driven DIAsDEM framework, we propose a new method to derive an archive-specific structured XML document type definition (DTD). Our approach utilizes association rule discovery and sequence mining techniques to structure a previously derived flat, i.e. unstructured DTD. We introduce the notion of a probabilistic DTD that is derived by discovering associations among and frequent sequences of XML tags, respectively.

1 Introduction

Up to 80% of a company's information is stored in unstructured textual documents. Hence, *document warehousing* and *text mining* are emerging disciplines for capturing and exploiting the flood of textual information for decision making [1]. However, acquiring interesting and actionable knowledge from textual databases is still a major challenge for the data mining community. Creating semantic markup is one form of providing explicit knowledge about text archives to facilitate search and browsing or to enable information integration with related data sources. Unfortunately, most users are not willing to manually create meta-data due to the efforts and costs involved [2]. Thus, text mining techniques are required to (semi-) automatically create semantic markup.

In this paper, we describe a KDD methodology for establishing a quasi-schema in the form of a probabilistic XML document type definition (DTD). This work is pursued in the research project DIAsDEM that focuses on text archives with domain-specific vocabulary and syntax. The DIAsDEM framework for semantic tagging of domain-specific texts was introduced in [3,4].

* The work of this author is funded by the German Research Society (DFG grant: SP 572/4-1) within the research project DIAsDEM. The German acronym stands for "Data Integration for Legacy Data and Semi-Structured Documents by Means of Data Mining Techniques".

Currently, the Java-based DIAsDEM Workbench derives a preliminary, flat and unstructured XML DTD from an archive of semantically tagged XML documents. However, we ultimately aim at integrating these XML documents with related, structured data sources. In this context, the derived list of XML tags should be transformed into a schema. As a first step, we derive an archive-specific *probabilistic DTD* from these tags, which (i) describes the most likely orderings of elements and (ii) adorns each element with statistical properties. We define a probabilistic DTD as a graph-based data structure describing the structural properties of the corresponding XML archive. Our future work consists of using this probabilistic DTD for the derivation of an archive-specific XML Schema and a relational schema, respectively. We introduce an algorithm for inferring a probabilistic DTD that utilizes association rule discovery algorithms and sequence mining techniques.

The rest of this paper is organized as follows: The next section discusses related work. Section 3 provides a concise presentation of the original DIAsDEM approach to semantic tagging. In section 4, we introduce the notion of probabilistic DTDs for textual archives and develop a method for deriving them. Section 5 summarizes a real-world case study. Finally, we conclude and give directions for future research.

2 Related Work

Concerning related knowledge discovery work [5,6,7], our approach shares with this research thread the objective of extracting semantic concepts from texts. However, concepts to be extracted in DIAsDEM must be appropriate to serve as XML DTD elements. Among other implications, discovering a concept that is peculiar to a single text unit is not sufficient for our purposes, although it may perfectly reflect the corresponding content. To derive a DTD, we need to discover groups of text units that share semantic concepts. Moreover, we concentrate on domain-specific texts, which significantly differ from average texts with respect to word frequency statistics. These archives can hardly be processed using standard text mining software, because the integration of domain knowledge is a prerequisite for successful knowledge discovery.

Currently, there are only a few research activities aiming at the transformation of texts into semantically annotated XML documents: Bruder et al. introduce the search engine GETESS that supports query processing on texts by creating and processing XML text abstracts [8]. These abstracts contain language-independent, content-weighted summaries of domain-specific texts. In DIAsDEM, we do not separate meta-data from original texts but rather provide a semantic annotation, keeping the texts intact for later processing or visualization. Additionally, the GETESS approach requires an a priori given DTD that corresponds to a domain-specific ontology. Erdmann et al. introduce a system that supports the semi-automated and ontology-based semantic annotation of Web pages [2]. The authors associate previously extracted text fragments (mostly named entities) with concepts of an a priori given ontology. In contrast, DIAsDEM aims at deriving an XML DTD from unstructured text documents.

To transform existing contents into XML documents, Sengupta and Purao propose a method that infers DTDs by using already tagged documents as input [9]. In contrast, we propose a method that tags plain text documents and derives a DTD for them. Moore and Berman present a technique to convert textual pathology reports into XML documents [10]. In contrast to our work, the authors neither derive an XML DTD nor apply a knowledge discovery methodology. They rather employ natural language processing techniques and a medical thesaurus to map terms and noun groups onto medical concepts. Thereafter, medical concepts serve as XML tags that semantically annotate the corresponding terms. Closer to our approach is the work of Lumera, who uses keywords and rules to semi-automatically convert legacy data into XML documents [11]. However, his approach relies on establishing a rule base that drives the conversion, while we use a KDD methodology that reduces necessary human intervention.

Semi-structured data is an area of related database research [12]. A lot of effort has recently been put into methods inferring and representing structure in similar semi-structured documents [13,14]. However, these approaches only derive a schema for a given set of semi-structured documents. Given a collection of marked up semi-structured texts, some authors employ grammatical inference techniques to create an XML DTD [15,16]. In contrast to our approach, these authors infer a probabilistic DTD for manually marked up XML or SGML documents. In DIAsDEM, we have to simultaneously solve the problems of both semi-structuring texts by semantic tagging and inferring an appropriately structured XML DTD.

3 The DIAsDEM Framework

Our work on creating an archive-specific and probabilistic XML DTD is based on the DIAsDEM framework for semantic tagging of document archives with domain-specific XML tags [3,4].

In DIAsDEM, the notion of semantic tagging refers to annotating texts with domain-specific XML tags that can have attributes describing named entities (e.g., names of persons). Normally, a document consists of many structural text units such as sentences or paragraphs. Hence, rather than classifying entire documents or tagging single terms, we aim at semantically annotating these structural text units in order to make their semantics explicit. The following excerpt illustrates two tagged sentences contained in a German Commercial Register entry, whereas each sentence corresponds to a text unit:

```
<BusinessPurpose> Der Betrieb von Spielhallen in Teltow und das Aufstellen von
Geldspiel- und Unterhaltungsautomaten. </BusinessPurpose>
<AppointmentManagingDirector Person="Balski; Pawel"> Pawel Balski ist zum
Geschäftsführer bestellt. </AppointmentManagingDirector>
```

Semantic tagging in DIAsDEM is a two-phase process: In its first phase, our proposed knowledge discovery in textual databases (KDT) process discovers clusters of semantically similar text units, tags documents in XML according to the results and derives an XML DTD describing the archive-specific document structure. The KDT process results in a final set of clusters whose labels serve as XML tags and DTD elements. Huge amounts of new documents can be converted into XML documents in the second, batch-oriented and productive phase of the DIAsDEM framework.

Besides the initial text documents to be tagged, the following domain knowledge constitutes input to our KDT process: A thesaurus containing a domain-specific taxonomy of terms and concepts, a preliminary conceptual schema of the domain and descriptions of specific named entities, e.g. persons and companies. The conceptual domain schema reflects the semantics of named entities and the relationships among them, as they are initially conceived by application experts. This schema might serve as a reference for the DTD to be derived from discovered semantic tags, but there is no guarantee that the final DTD will be contained in or will contain this schema.

Similarly to a conventional KDD process, our process starts with a preprocessing phase that includes basic NLP preprocessing tasks such as tokenization, normalization and word stemming as well as named entity extraction. Instead of removing stop words, we establish a drastically reduced feature space by selecting a limited set of terms and concepts (so-called text unit descriptors) from the thesaurus and the conceptual schema. Text unit descriptors are currently chosen by the knowledge engineer because they must reflect important concepts of the application domain. All text units are mapped onto Boolean vectors of this feature space. Thereafter, the Boolean text unit vectors are further processed by applying an information retrieval weighting schema (i.e. TF-IDF).

In the pattern discovery phase, all text unit vectors contained in the initial archive are clustered based on content similarity. The objective is to discover dense and homogeneous text unit clusters. Clustering is performed in multiple iterations. Each iteration outputs a set of clusters, which is partitioned into "acceptable" and "unacceptable" ones according to our quality criteria. A cluster of text unit vectors is qualitatively "acceptable", if and only if (i) its cardinality is large and the corresponding text units are (ii) homogeneous and (iii) can be semantically described by a small number of text unit descriptors. Members of "acceptable" cluster are subsequently removed from the dataset for later labeling, whereas the remaining text unit vectors are input data to the clustering algorithm in the next iteration. In each iteration, the cluster similarity threshold value is stepwisely decreased such that "acceptable" clusters become progressively less specific in content. The KDT process is based on a plug-in concept that allows the execution of different clustering algorithms within the DIAsDEM Workbench.

In the postmining phase, all "acceptable" clusters are semi-automatically assigned a semantic label. The DIAsDEM Workbench performs both a preselection and a ranking of candidate cluster labels for the expert to choose from.

The default cluster labels are derived from prevailing feature space dimensions (i.e. text unit descriptors) in each “acceptable” cluster. Cluster labels actually correspond to XML tags that are subsequently used to annotate cluster members. Thereafter, all original documents are annotated using valid XML tags that have attributes reflecting previously extracted named entities and their values. Finally, an unstructured XML DTD is derived that describes the semantic structure of the XML collection by enumerating existing XML tags. The following DTD excerpt was created in a recent case study [4]:

```
<!ELEMENT CommercialRegisterEntry ( #PCDATA | FoundationPartnership |
ShareCapital | AppointmentManagingDirector | (...) | ConclusionArticles |
BusinessPurpose | Owner )* > (...) <!ELEMENT Owner (#PCDATA)>
```

Obviously, this preliminary DTD has two shortcomings: Its lacks structure and has no indications of mandatory, optional or interdependent elements. We alleviate these shortcomings by deriving a probabilistic DTD from the archive of XML documents described by this enumeration of tags.

4 Establishing a Probabilistic DTD

Our new method of establishing a probabilistic DTD aims at specifying the most appropriate ordering of XML tags, identifying correlated or mutually exclusive XML tags and adorning each XML tag and each correlation among them with statistical properties. These properties form the basis for reliable query processing, because they determine the expected precision and recall of the query results. In the following, we first introduce the statistical properties of DTD elements, we afterwards describe a methodology for computing these statistics for associated XML tags and sequences of tags, Finally, we apply a heuristic pruning algorithm to derive a structured probabilistic DTD of frequent elements.

4.1 Statistical Properties of Semantic DTD Elements

For the derived XML tags, we are interested in whether they should be observed as mandatory inside the DTD, whether they are associated with other tags and whether their most likely relative position inside the DTD can be assessed. In some application domains, a human expert might identify the mandatory parts of the DTD and specify the ordering of XML tags. However, archive documents may still violate specifications of the expert, either because the authors did not respect the specifications, or, as in our case, there has been no a priori DTD for the archive.

Therefore, we define statistic properties of XML tags, associations of tags and sequences of tags. Thereafter, structure and optionality of DTD elements can de determined on the basis of these property values. In particular, let d be an XML document contained in an XML archive $D = \{d_1, \ldots, d_{|D|}\}$. Additionally, let $T = \{t_1, \ldots, t_{|T|}\}$ be the set of XML tags contained in the derived XML

DTD, let $\{x, y_1, \ldots, y_n\} \in T$ or abbreviated $x, y_1, \ldots, y_n \in T$ be a set of $n+1$ XML tags and $< y_1 \cdot \ldots \cdot y_n \cdot x >$ or abbreviated $y_1 \cdot \ldots \cdot y_n \cdot x$ be a sequence of $n+1$ adjacent XML tags. The function $Tags(d)$ returns the set of all XML tags contained in d. The function $Seqs(d)$ returns the set of all adjacent sequences of XML tags contained in d. For example, consider document d that consists of three semantically tagged text units, i.e. $t_1 \cdot t_2 \cdot t_1$. In this case, $Tags(d) = \{t_1, t_2\}$ and $Seqs(d) = \{t_1 \cdot t_2 \cdot t_1, t_1 \cdot t_2, t_2 \cdot t_1\}$.

TagSupport of tag x is defined as the relative frequency of tag x among the documents of the archive. It is an indicator of whether this tag is likely to be mandatory:

$$TagSupport(x) = \frac{|\{d \in D | x \in Tags(d)\}|}{|D|}$$

A tag may be mandatory in the entire archive or in a particular subset of documents that are characterized by other, associated tags. We use association rule discovery to identify DTD elements frequently appearing together and define *AssociationConfidence* of tag x with respect to the set of tags $y_1, \ldots, y_n$ as follows:

$$AssociationConfidence(y_1, \ldots, y_n \to x) = \frac{|\{d \in D | x, y_1, \ldots, y_n \subseteq Tags(d)\}|}{|\{d \in D | y_1, \ldots, y_n \subseteq Tags(d)\}|}$$

Similarly to conventional association rule discovery, we need to exclude spurious correlations caused by a very high support of a tag in the entire population. To alleviate this problem, we use lift or improvement of association rules and define *AssociationLift* of tag x given tags $y_1, \ldots, y_n$ as follows:

$$AssociationLift(y_1, \ldots, y_n \to x) = \frac{AssociationConfidence(y_1, \ldots, y_n \to x)}{TagSupport(x)}$$

To identify potential orderings of tags in a structured DTD, we perform sequence mining among the XML tags of all documents. On the basis of tag sequences frequently appearing in the archive, we define the *SequenceConfidence* of tag x after a sequence of adjacent tags $y_1 \cdot \ldots \cdot y_n$ as follows:

$$SequenceConfidence(y_1 \cdot \ldots \cdot y_n \cdot x) = \frac{|\{d \in D | y_1 \cdot \ldots \cdot y_n \cdot x \in Seqs(d)\}|}{|\{d \in D | y_1 \cdot \ldots \cdot y_n \in Seqs(d)\}|}$$

This definition differs from the conventional statistics known for sequence mining [17], because we are concentrating on adjacent tags, disallowing the occurrence of arbitrary tags in-between. This constraint is necessary for the placement of associated tags in a structured DTD. Conventional sequence miners do not ensure that frequent sequences are comprised of adjacent elements. However, some Web usage miners are capable of distinguishing between adjacent and non-adjacent events [18,19,20].

Analogously to *AssociationLift*, we define the *SequenceLift* of tag x after a sequence of adjacent tags $y_1 \cdot \ldots \cdot y_n$ as follows:

$$SequenceLift(y_1 \cdot \ldots \cdot y_n \cdot x) = \frac{SequenceConfidence(y_1 \cdot \ldots \cdot y_n \cdot x)}{TagSupport(x)}$$

The notion of support holds both for sets of associated tags and for sequences of adjacent tags. Instead of defining two support functions, we introduce the notion of an element "group" g, being either a set of tags $y_1, \ldots, y_n$ or a sequence of adjacent tags $y_1 \cdot \ldots \cdot y_n$. We define *GroupSupport* for groups of tags as follows:

$$GroupSupport(g) = \frac{|\{d \in D | g \in Tags(d) \cup Seqs(d)\}|}{|D|}$$

For any set of at least two tags, this property assumes one value for the set and as many values as are the perturbations of set members. In the following subsection, we show how the statistical information pertinent to individual tags, to groups of tags groups and to relationships among them is modeled in a seamless way.

4.2 A Graph of Associated Groups of XML Tags

Frequent groups of XML tags are discovered by applying association rule discovery and specialized sequence mining algorithms with appropriate support constraints. A formal data structure is required to represent the results. Additionally, an algorithm should be developed to derive a probabilistic DTD from this data structure. We use a directed "graph of associated groups" whose nodes are individual tags, sequences of adjacent tags or sets of co-occurring tags. Each node is adorned with statistical properties pertinent to its tag and its tag group, respectively. An edge represents either a relationship $y_1, \ldots, y_n \rightarrow x$ or $y_1 \cdot \ldots \cdot y_n \cdot x$. The groups of nodes $y_1, \ldots, y_n$ and $y_1 \cdot \ldots \cdot y_n$ are referred to as *source* nodes, whereas x is the *target* node. Similarly to nodes, each edge is adorned with statistics of the order-insensitive or order-sensitive association of tags it represents. Let $V \subseteq T \times (0, 1]$ be the set of graph nodes conforming to the signature:

$$< TagName, TagSupport >$$

Note that XML tag x only appears in the graph if $TagSupport(x) > 0$. For groups of tags, we distinguish between order-sensitive and order-insensitive groups by imposing an arbitrary ordering upon groups of tags, e.g. lexicographical ordering. Thereafter, each tag group is observed as an order-sensitive or an order-insensitive list: An order-sensitive list is a sequence of tags, and an order-insensitive list is a set of tags.

More formally, let $P(V)$ be the set of all lists of elements in V, i.e. (*TagName*, *TagSupport*)-pairs. A tag group $g \in P(V) \times \{0, 1\}$ has the form $(< v_1, \ldots, v_k >, 1)$, where $< v_1, \ldots, v_k >$ is a list of elements from V and the value 1 indicates that this list represents an order-sensitive group. Similarly, $g' = (< v_1, \ldots, v_k >, 0)$ would represent the unique order-insensitive group composed of $v_1, \ldots, v_k \in V$.

For example, let $a, b \in V$ be two tags annotated with their *TagSupport*, whereby a precedes b lexicographically. The groups $(< a, b >, 1)$ and $(< b, a >, 1)$ are two distinct order-sensitive groups. $(< a, b >, 0)$ is the order-insensitive group of the two elements. Finally, the group $(< b, a >, 0)$ is not permitted, because the group is order-insensitive but the list violates the lexicographical ordering of list elements. Using $P(V) \times \{0, 1\}$, we define $V' \subseteq (P(V) \times \{0, 1\}) \times (0, 1]$ with signature:

$$< TagGroup, GroupSupport >$$

V' contains only groups of annotations whose *GroupSupport* value is above a given threshold. Of course, the threshold value affects the size of the graph and the execution time of the algorithm traversing it to build the DTD. The set of nodes constituting our graph is $\mathcal{V} = V \cup V'$, indicating that a node may be a single tag or a group of tags with its/their statistics. An edge emanates from an element of V' and points to an element of V, i.e. from an associated group of tags to a single tag. Formally, we define the set of edges $E \subseteq (V \times V') \times \mathcal{X} \times \mathcal{X} \times \mathcal{X} \times \mathcal{X}$, where $\mathcal{X} := (0, 1] \cup \{NULL\}$, with signature:

$$< Edge, AssociationConfidence, AssociationLift, \\ SequenceConfidence, SequenceLift >$$

In this signature, the statistical properties refer to the edge's target given the group of nodes in the edge's source. If the source is a sequence of adjacent tags, then *SequenceConfidence* and *SequenceLift* are the only valid statistical properties, because *AssociationConfidence* and *AssociationLift* are inapplicable. If the source is a set of tags, then the both *SequenceConfidence* and *SequenceLift* are inapplicable. Inapplicable statistical properties assume the NULL value.

4.3 Deriving DTD Components from the Graph of Associated Groups

The graph of associated groups has one node for each frequent tag and one node for each frequent group of tags. Depending on the support value threshold for discovering association rules and frequent sequences, the graph may contain a very large number of associated groups or rather the most frequent ones. In both cases, we perform further pruning steps to eliminate all associations that are of less importance in the context of a DTD. We consider the following pruning criteria:

- All edges with a lift (association lift or sequence lift) less than 1 are eliminated.
- All edges with a confidence less than a threshold are eliminated.
- All nodes containing tag groups that are not connected to a single tag by any edge, are removed. Such nodes are pruned after pruning all edges pointing to them.

– For each tag having k ingoing edges from tag groups, we retain only groups of maximal size, subject to a confidence threshold. This criterion states that if a tag x appears after a group g with confidence c and a subgroup g' of g with confidence c', the subgroup g' is removed if $c' - c \leq \epsilon$, otherwise the group g is removed.

After this pruning procedure, the graph has been stripped off all groups that (i) reflect spurious associations, (ii) lead to tags with low confidence or (iii) can be replaced by groups that lead to frequent tags with higher confidence. The output of this procedure is a collection of frequent components of the envisaged DTD.

4.4 DTD as a Tree of Alternative XML Tag Sequences

The pruning phase upon the graph of associated groups delivers components of the probabilistic DTD. However, these components do not constitute a well-defined DTD, because there is no knowledge about their relative ordering and placement inside the type definition. Hence, we introduce a complementary DTD establishment algorithm, which derives complete sequences of DTD elements.

We observe an order-preserving DTD as a tree of alternative subsequences of XML tags. Each tag is adorned with its support with respect to the subsequence leading to it inside the tree: This support value denotes the number of documents starting with the same subsequence of tags. Each XML tag may appear in more than one subsequence, because of different predecessors in each one. Observing the DTD as a tree implies a common root. In the general case, each document of the archive may start at a different tag. We thus assume a dummy root whose children are tags appearing first in documents. In general, a tree node refers to a tag x, and its children refer to the tags appearing after x in the context of x's own predecessors. In a sense, the DTD as a tree of alternatives resembles a DataGuide [21], although the latter contains no statistical adornments.

The tree-of-alternatives method is realized by the preprocessor module of the Web usage miner WUM [20]. This module is responsible for coercing sequences of events by common prefix and placing them in a tree structure that is called "aggregated tree". This tree is input to the sequential pattern discovery process performed by WUM. The tag sequences contained in documents can be observed as sequences of events. Hence, the WUM preprocessor can also be used to build a DTD over an archive as a tree of alternative tag sequences.

The tree-of-alternatives can be pruned by the same set of criteria as applied in the graph of associated groups. Since each branch of the tree-of-alternatives corresponds to a sequence of adjacent tags, we only consider the statistical properties $SequenceLift$, *SequenceConfidence* and *GroupSupport* of sequences. It should also be stressed that the constraints placed upon the branches of this tree should be less restrictive than those applied upon *all* sequences of adjacent tags, because the tree branches correspond to complete sequences of tags, from the first tag in a document to the last one.

Finally, the DTD components retained on the graph of associated groups can be exploited to refine the tree-of-alternatives further. In particular, an ordered group of tags (i.e. a sequence of tags) $g = y_1 \cdot \ldots \cdot y_n$ appearing as node in the graph may appear in several branches of the tree-of-alternatives. Then, if the same group appears in multiple branches with different prefixes (i.e. different sequences of tags prior to the group), these prefixes can be considered as alternatives inside the DTD, followed by a mandatory sequence of tags g.

In the current version of the DIAsDEM procedure for DTD establishment, we are still considering the graph of associated groups and the tree-of-alternatives as independent options, and are investigating the impact of heuristics combining them, like the aforementioned one, on the quality of the output DTD.

5 Case Study

To test the applicability of our approach, we have used 1,145 German Commercial Register entries published by the district court of Potsdam in 1999. These foundation entries of new companies have been semantically tagged by applying the original DIAsDEM framework as described in a recent case study [4].

Our current research successfully continued this case study by creating the DTD establishment graph for the previously derived, unstructured XML DTD illustrated in section 3. To this end, the Java-based DIAsDEM Workbench has been extended to analyze the XML document collection in order to compute $TagSupport$ for all XML tags and to employ dedicated algorithms for association rule discovery (i.e. Weka [22]) and mining frequent sequences (i.e. WUM [20]) to compute $GroupSupport$ as well as $Confidence$ and $Lift$ for tag associations and tag sequences. The following expert-given threshold values have been applied by the DIAsDEM Workbench: $TagSupport > 0.5$, $AssociationConfidence > 0.75$, $AssociationLift > 1.2$, $SequenceConfidence > 0.5$ and $SequenceLift > 1.0$.

Figure 1 depicts an excerpt of the DTD establishment graph that contains nodes corresponding to either individual XML tags (e.g., the tag $ShareCapital$

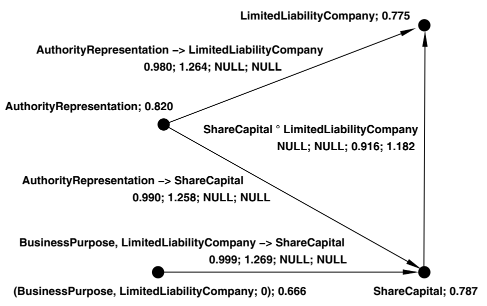
LimitedLiabilityCompany; 0.775
AuthorityRepresentation –> LimitedLiabilityCompany
0.980; 1.264; NULL; NULL
AuthorityRepresentation; 0.820
ShareCapital ° LimitedLiabilityCompany
NULL; NULL; 0.916; 1.182
AuthorityRepresentation –> ShareCapital
0.990; 1.258; NULL; NULL
BusinessPurpose, LimitedLiabilityCompany –> ShareCapital
0.999; 1.269; NULL; NULL
(BusinessPurpose, LimitedLiabilityCompany; 0); 0.666
ShareCapital; 0.787

Fig. 1. Excerpt of the derived DTD establishment graph

with $TagSupport = 0.787$) or a group of XML tags (i.e. the tag set $BusinessPurpose, LimitedLiabilityCompany$ with $GroupSupport = 0.666$). Figure 1 does not depict a node that corresponds to a sequence of tags. The edge connecting $AuthorityRepresentation$ and $ShareCapital$ represents the association rule $AuthorityRepresentation \rightarrow ShareCapital$ with very high $AssociationConfidence = 0.990$ and moderate $AssociationLift = 1.258$. Finally, the edge connecting $ShareCapital$ and $LimitedLiabilityCompany$ represents the frequent tag sequence $ShareCapital \cdot LimitedLiabilityCompany$ with high $SequenceConfidence = 0.916$ and moderate $SequenceLift = 1.182$.

Using the DTD establishment graph, the expert can acquire important insights into the semantic structure of an XML archive. By interactively visualizing this graph, a knowledge engineer can detect subsets of XML tags that are frequently occurring together in semantically annotated documents. In Figure 1, the tag group $BusinessPurpose, LimitedLiabilityCompany$ occurs in 66% of the annotated XML documents. This fact indicates the existence of a semantic subgroup within the text archive focusing on legal matters of limited liability companies. The analysis of frequent sequences of adjacent tags reveals knowledge about the ordering of annotated text units within the XML documents. Referring to Figure 1, the tag $ShareCapital$ is followed by $LimitedLiabilityCompany$ with a probability of 0.916. Additionally, candidates for mandatory tags can be identified such as $AuthorityRepresentation$ which is contained in 82% of the processed XML documents. Hence, the DTD establishment graph serves as an overall description of the semantic structure shared by either the entire domain-specific archive or subsets of semantically related XML documents.

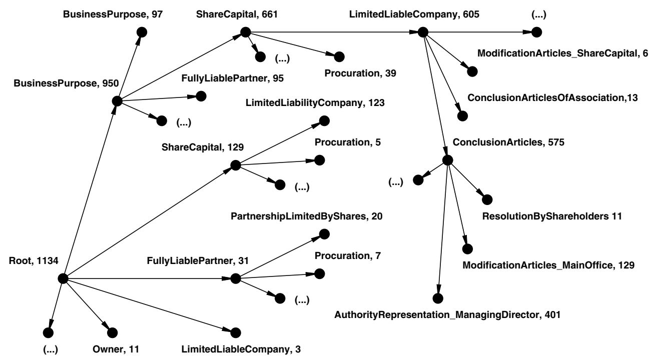
BusinessPurpose, 97
ShareCapital, 661
LimitedLiableCompany, 605
(...)
ModificationArticles_ShareCapital, 6
(...)
Procuration, 39
BusinessPurpose, 950
FullyLiablePartner, 95
ConclusionArticlesOfAssociation,13
(...)
LimitedLiabilityCompany, 123
Procuration, 5
ConclusionArticles, 575
ShareCapital, 129
(...)
(...)
PartnershipLimitedByShares, 20
ResolutionByShareholders 11
Procuration, 7
Root, 1134
FullyLiablePartner, 31
ModificationArticles_MainOffice, 129
(...)
AuthorityRepresentation_ManagingDirector, 401
(...)
Owner, 11
LimitedLiableCompany, 3

Fig. 2. Commercial Register DTD as a tree of alternative tag sequences

Figure 2 depicts the probabilistic XML DTD as a tree of alternative XML tag sequences that has been derived from the Commercial Register archive. Created by the preprocessing module of the Web Utilization Miner WUM [20], this tree explicitly describes sequences of XML tags frequently occurring in the archive.

For example, 950 (out of 1134) XML documents start with a text unit that is annotated with the XML tag *BusinessPurpose*. Following this tag sequence, 661 (out of 950) documents continue with a sentence that is tagged as *Share-Capital*. Using this probabilistic DTD, the expert can acquire knowledge about dominating sequences of XML tags which is essential for imposing an ordering upon the discovered XML tags.

6 Conclusion

Acquiring knowledge encapsulated in documents implies effective querying techniques as well as the combination of information from different texts. This functionality is usually confined to database-like query processors, while text search engines scan individual textual resources and return ranked results.

In this study, we have presented a methodology that structures document archives to enable query processing over them. We have proposed the derivation of an XML DTD over a domain-specific text archive by means of data mining techniques. Our main emphasis has been the combination of XML tags reflecting the semantics of many text units across the archive into a single DTD reflecting the semantics of the entire archive. The statistical properties of tags and their relationships form the basis for combining them into a unifying DTD. We use a graph data structure to depict all statistics that can serve as a basis for this operation, and we have proposed a mechanism that derives a DTD by employing a mining algorithm.

Our future work includes the implementation of further mechanisms to derive probabilistic DTDs and the establishment of a framework for comparing them in terms of expressiveness and accuracy. The data structure *probabilistic DTD* will be utilized to derive an archive-specific XML Schema and a relational schema, respectively. Ultimately, a full-fledged querying mechanism over text archives should be established. To this purpose, we intend to couple DTD derivation methods with a query mechanism for semi-structured data. Each semantic annotation corresponds to a label that semantically describes a discovered text unit cluster. Currently, the underlying clustering algorithm creates non-overlapping clusters. Hence, each text unit belongs to exactly one cluster. Since each text unit can only be annotated with the label of its cluster, the derived XML tags cannot be nested. An extension of the DIAsDEM Workbench to utilize a hierarchical clustering algorithm would allow for the establishment of subclusters and thus for the nesting of (sub)cluster labels. This is planned as future work as well.

References

1. Sullivan, D.: Document Warehousing and Text Mining. John Wiley & Sons, New York, Chichester, Weinheim (2001) 461
2. Erdmann, M., Maedche, A., Schnurr, H. P., Staab, S.: From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In: Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, Luxembourg (2000) 461, 462

3. Graubitz, H., Spiliopoulou, M., Winkler, K.: The DIAsDEM framework for converting domain-specific texts into XML documents with data mining techniques. In: Proceedings of the First IEEE Int. Conference on Data Mining, San Jose, CA, USA (2001) 171–178 461, 463
4. Winkler, K., Spiliopoulou, M.: Semi-automated XML tagging of public text archives: A case study. In: Proceedings of EuroWeb 2001 "The Web in Public Administration", Pisa, Italy (2001) 271–285 461, 463, 465, 470
5. Nahm, U. Y., Mooney, R. J.: Using information extraction to aid the discovery of prediction rules from text. In: Proceedings of the KDD-2000 Workshop on Text Mining, Boston, MA, USA (2000) 51–58 462
6. Feldman, R., Fresko, M., Kinar, Y., Lindell, Y., Liphstat, O., Rajman, M., Schler, Y., Zamir, O.: Text mining at the term level. In: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery, Nantes, France (1998) 65–73 462
7. Loh, S., Wives, L. K., Oliveira, J. P. M. d.: Concept-based knowledge discovery in texts extracted from the Web. ACM SIGKDD Explorations **2** (2000) 29–39 462
8. Bruder, I., Düsterhöft, A., Becker, M., Bedersdorfer, J., Neumann, G.: GETESS: Constructing a linguistic search index for an Internet search engine. In Bouzeghoub, M., Kedad, Z., Metais, E., eds.: Natural Language Processing and Information Systems. Number 1959 in Lecture Notes in Computer Science. Springer-Verlag (2001) 227–238 462
9. Sengupta, A., Purao, S.: Transitioning existing content: Inferring organization-spezific document structures. In Turowski, K., Fellner, K. J., eds.: Tagungsband der 1. Deutschen Tagung XML 2000, XML Meets Business, Heidelberg, Germany (2000) 130–135 463
10. Moore, G. W., Berman, J. J.: Medical data mining and knowledge discovery. In: Anatomic Pathology Data Mining. Volume 60 of Studies in Fuzziness and Soft Computing., Heidelberg, New York, Physica-Verlag (2001) 72–117 463
11. Lumera, J.: Große Mengen an Altdaten stehen XML-Umstieg im Weg. Computerwoche **27** (2000) 52–53 463
12. Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufman Publishers, San Francisco (2000) 463
13. Wang, K., Liu, H.: Discovering structural association of semistructured data. IEEE Transactions on Knowledge and Data Engineering **12** (2000) 353–371 463
14. Laur, P. A., Masseglia, F., Poncelet, P.: Schema mining: Finding regularity among semistructured data. In Zighed, D. A., Komorowski, J., Żytkow, J., eds.: Principles of Data Mining and Knowledge Discovery: 4th European Conference, PKDD 2000. Volume 1910 of Lecture Notes in Artificial Intelligence., Lyon, France, Springer, Berlin, Heidelberg (2000) 498–503 463
15. Carrasco, R. C., Oncina, J.: Learning deterministic regular grammars from stochastic samples in polynomial time. RAIRO (Theoretical Informatics and Applications) **33** (1999) 1–20 463
16. Young-Lai, M., Tompa, F. W.: Stochastic grammatical inference of text database structure. Machine Learning **40** (2000) 111–137 463
17. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of Int. Conf. on Data Engineering, Taipei, Taiwan (1995) 466
18. Baumgarten, M., Büchner, A. G., Anand, S. S., Mulvenna, M. D., Hughes, J. G.: Navigation pattern discovery from internet data. In: [23]. (2000) 70–87 466
19. Gaul, W., Schmidt-Thieme, L.: Mining web navigation path fragments. In: [24]. (2000) 466

20. Spiliopoulou, M.: The laborious way from data mining to web mining. Int. Journal of Comp. Sys., Sci. & Eng., Special Issue on "Semantics of the Web" **14** (1999) 113–126 466, 469, 470, 471
21. Goldman, R., Widom, J.: DataGuides: Enabling query formulation and optimization in semistructured databases. In: VLDB'97, Athens, Greece (1997) 436–445 469
22. Witten, I. H., Frank, E.: Data Mining. Morgan Kaufmann Publishers, San Francisco (2000) 470
23. Masand, B., Spiliopoulou, M., eds.: Advances in Web Usage Mining and User Profiling: Proceedings of the WEBKDD'99 Workshop. LNAI 1836, Springer Verlag (2000) 473
24. Kohavi, R., Spiliopoulou, M., Srivastava, J., eds.: KDD'2000 Workshop WEBKDD'2000 on Web Mining for E-Commerce — Challenges and Opportunities, Boston, MA, ACM (2000) 473