Extraction of Semantic XML DTDs from Texts Using Data Mining Techniques

Karsten Winkler*[*] and Myra Spiliopoulou
Leipzig Graduate School of Management
Department of E-Business
Jahnallee 59, D-04109 Leipzig, Germany
{kwinkler,myra}@ebusiness.hhl.de

Abstract
Although composed of unstructured texts, documents contained in textual archives such as public announcements, patient records and annual reports to shareholders often share an inherent though undocumented structure. In order to facilitate efficient, structure-based search in archives and to enable information integration of text collections with related data sources, this inherent structure should be made explicit as detailed as possible. Inferring a semantic and structured XML document type definition (DTD) for an archive and subsequently transforming the corresponding texts into XML documents is a successful method to achieve this objective. The main contribution of this paper is a new method to derive structured XML DTDs in order to extend previously derived flat DTDs. We use the DIAsDEM framework to derive a preliminary, unstructured XML DTD whose components are supported by a large number of documents. However, all XML tags contained in this preliminary DTD cannot a priori be assumed to be mandatory. Additionally, there is no fixed order of XML tags and automatically tagging an archive using a derived DTD always implicates tagging errors. Hence, we introduce the notion of probabilistic XML DTDs whose components are assigned probabilities of being semantically and structurally correct. Our method for establishing a probabilistic XML DTD is based on discovering associations between, resp. frequent sequences of XML tags.

Keywords
semantic annotation, XML, DTD derivation, knowledge discovery, data mining, clustering

INTRODUCTION

Most organizations are not only "drowning" in data, they are also "struggling" to cope with huge amounts of text documents. Tan points out that up to 80% of a company's information is stored in unstructured textual documents [26]. Hence, capturing interesting and actionable knowledge from textual databases is a major challenge for the data mining community. Creating semantic markup is one form of providing explicit knowledge about text archives to facilitate searching and browsing or to enable information integration with related data sources. Unfortunately, most users are not willing to manually create metadata due to the efforts and costs involved [7]. Thus, text mining techniques are required that (semi-) automatically create semantic markup and tag documents accordingly.

In this paper, we present the KDD approach pursued in the research project DIAsDEM whose German acronym stands for "Data Integration for Legacy Systems and Semi-Structured Documents by Means of Data Mining Techniques". Our goal is semantic tagging of textual content with meta-data to facilitate searching, querying, identification of and integration with associated texts and relational data. Hence, we aim at deriving a structured XML DTD that serves as a quasi-schema for the document collection and enables the provision of database-like querying services on textual data. DIAsDEM focuses on text collections with domain-specific vocabulary and syntax that frequently share an inherent, but undocumented structure.

The DIAsDEM framework for semantic tagging of domain-specific texts was introduced in [12, 11]. However, applying the Java-based *DIAsDEM Workbench* to a text archive currently results in a collection of semantically tagged XML documents that are described by the extracted flat, unstructured XML DTD. However, we ultimately aim at integrating the resulting XML documents with other related data sources. In this context, the derived unstructured, rather preliminary DTD should be transformed into more structured DTD that reflects both ordering and optionality of tags. Given that all XML tags are derived by data mining techniques (i.e. iterative clustering as explained in section 3), they are not crisp due to tagging errors. Taking this critical fact into account, we introduce the notion of a *probabilistic DTD* that describes the most likely orderings of XML tags and that contains statistical properties for each tag. The structured DTD will be the basis for future information integration efforts that involve XML archives generated by the *DIAsDEM Workbench*. We introduce two algorithms for inferring a probabilistic DTD that utilize association rule discovery algorithms and sequence mining techniques.

The rest of this paper is organized as follows: The next

[*]The work of this author is funded by the German Research Society (DFG grant no. SP 572/4-1).

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE CommercialRegisterEntry SYSTEM 'CommercialRegisterEntry.dtd'>

<CommercialRegisterEntry> <BusinessPurpose> Der Betrieb von Spielhallen in Teltow und das
Aufstellen von Geldspiel- und Unterhaltungsautomaten. </BusinessPurpose> <ShareCapital
AmoutOfMoney="25000 EUR"> Stammkapital: 25.000 EUR. </ShareCapital>
<LimitedLiabilityCompany> Gesellschaft mit beschränkter Haftung. </LimitedLiabilityCompany>
<ConclusionArticles Date="12.11.1998; 19.04.1999"> Der Gesellschaftsvertrag ist am 12.11.1998
abgeschlossen und am 19.04.1999 abgeändert. </ConclusionArticles> (...) Einzelvertretungsbefugnis kann erteilt
werden. <AppointmentManagingDirector Person="Balski; Pawel; Berlin; 14.04.1965">
Pawel Balski, 14.04.1965, Berlin, ist zum Geschäftsführer bestellt. </AppointmentManagingDirector> (...)
<PublicationMedia> Nicht eingetragen: Die Bekanntmachungen der Gesellschaft erfolgen im Bundesanzeiger.
</PublicationMedia> </CommercialRegisterEntry>
```

Table 1: XML document containing an annotated Commercial Register entry

section briefly discusses related work. Section 3 gives an overview of our framework for semantic tagging of domain-specific text collections. Section 4 introduces the notion of probabilistic DTDs for textual archives and develops two methods for deriving them. Finally, we conclude and give directions for future research in section 5.

RELATED WORK

Nahn and Mooney propose the combination of methods from KDD and information extraction to perform text mining tasks [19]. They apply standard KDD techniques to a collection of structured records that contain previously extracted, application-specific features from texts. Feldman et al. propose text mining at the term level instead of focusing on linguistically tagged words [8]. The authors represent each document by a set of terms and additionally construct a taxonomy of terms. The resulting dataset is input to KDD algorithms such as association rule discovery. Our DIAsDEM framework adopts the idea of representing texts by terms and concepts. However, our goal is the semantic tagging of structural text units (e.g., sentences or paragraphs) within the document according to a global DTD and not the characterization of the entire document's content. Loh et al. suggest to extract concepts rather than individual words for subsequent use in KDD efforts at the document level. [15]. Similarly to our framework, the authors suggest to exploit existing vocabularies such as thesauri for concept extraction. Mikheev and Finch describe a workbench to acquire domain knowledge from texts [18]. Similar to the *DIAsDEM Workbench*, their approach combines methods from different fields of research in a unifying framework.

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 elements of the XML DTD. 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. In order to derive a DTD, we need to discover groups of text units that share some semantic concepts. Moreover, we concentrate on domain-specific texts, which significantly differ from average texts with respect to word frequency statistics. These collections can hardly be processed using standard text mining software because the integration of relevant domain knowledge is a prerequisite for successful knowledge discovery.

There are only a few research activities aiming at the transformation of texts into semantically annotated XML documents: Becker et al. introduce the search engine GETESS that supports query processing on texts by deriving and processing XML text abstracts [4]. 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. Given the aforementioned linguistic particularities of the application domains we investigate, a DTD characterizing the content of the documents is more appropriate than inferences on their content. In order to transform existing content into XML documents, Sengupta and Purao propose a method that infers DTDs by using already tagged documents as input [23]. In contrast, we propose a method that tags plain text documents and derives a DTD for them. Closer to our approach is the work of Lumera, who uses keywords and rules to semi-automatically convert legacy data into XML documents [16]. However, his approach relies on establishing a rule base that drives the conversion, while we use a KDD methodology that reduces human effort.

Semi-structured data is another topic of related research within the database community [6, 1]. A lot of effort has recently been put into methods inferring and representing structure in similar semi-structured documents [21, 27, 14]. However, these approaches only derive a schema for a given set of semi-structured documents. In DIAsDEM, we have to simultaneously solve the problems of both semi-structuring text documents by semantic tagging and inferring an appropriately structured XML DTD that describes the related archive. We are not aware of any scientific or commercial approaches employing probabilistic document type definitions as introduced in this paper for describing text archives or integrating texts with related data sources.

THE DIAsDEM FRAMEWORK

In this paper, the notion of semantic tagging refers to the activity of annotating texts with domain-specific XML tags that might contain additional attributes. Rather than classifying entire documents or tagging single terms, we aim at semantically tagging text units such as sentences or paragraphs. Table 1 illustrates this concept of semantic tagging, whereas each sentence of this German Commercial Register entry is a text unit. In this example, the semantics of most sentences are made explicit by XML tags that partly contain additional attributes describing extracted named entities (e.g., names of persons and amounts of money). The XML document depicted in Table 1 was created by applying the DIAsDEM framework to a collection of 1,145 textual Commercial Register entries containing 10,785 text units. This collection includes all entries related to foundations of companies in the district of the German city Potsdam in 1999. In Germany, companies are obliged by law to submit various information about business affairs to local Commercial Registers. Although Commercial Registers are an important source of information in daily business transactions, their textual content can only be searched using full-text queries at the moment. Hence, semantically semi-structuring these textual archives provides the basis for information integration and creation of value-adding services related to information brokerage. XML query languages could be employed to submit both both content- and structure-based queries against semantically tagged XML archives.

Our framework pursues two objectives for a given archive of text documents: All text documents should be semantically tagged and an appropriate, preliminary flat XML DTD should be derived for the archive. Semantic tagging in DIAsDEM is a two-phase process. We have designed a knowledge discovery in textual databases (KDT) process that constitutes the first phase in order to build clusters of semantically similar text units, to tag documents in XML according to the results and to derive an XML DTD describing the archive. The KDT process that was introduced in [12, 11] 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. All text units contained in new documents are clustered by the previously built text unit clusterer and are subsequently tagged with the corresponding cluster labels.

In DIAsDEM we concentrate on the semantic tagging of similar text documents originating from a common domain. Nevertheless, the DIAsDEM approach is appropriate for semantically tagging various kinds of archives such as public announcements of courts and administrative authorities, quarterly and annual reports to shareholders, textual patient records in health care applications as well as product and service descriptions published on electronic marketplaces.

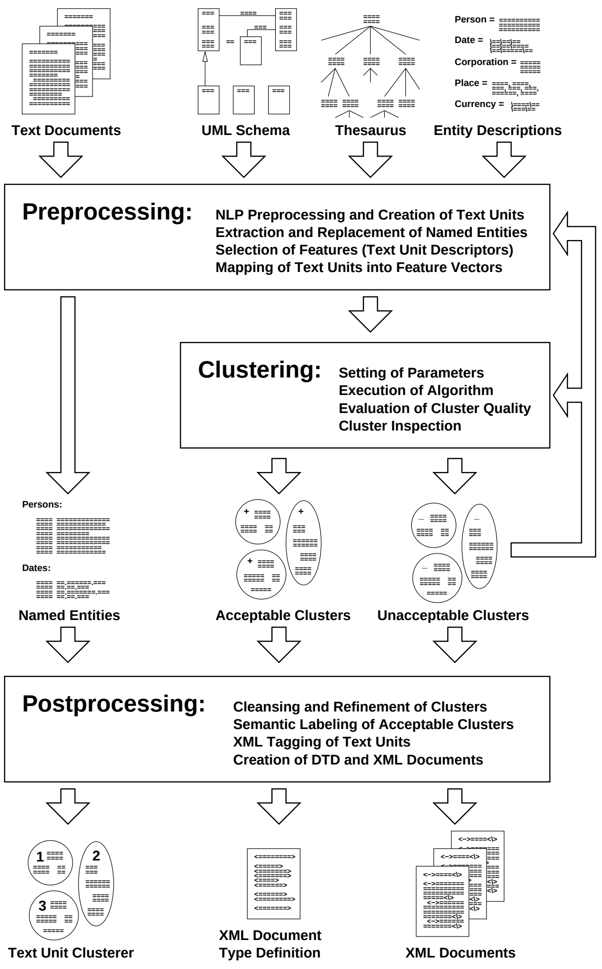
Text Documents
UML Schema
Thesaurus
Person =
Date =
Corporation =
Place =
Currency =
Entity Descriptions
Preprocessing:
NLP Preprocessing and Creation of Text Units
Extraction and Replacement of Named Entities
Selection of Features (Text Unit Descriptors)
Mapping of Text Units into Feature Vectors
Clustering:
Setting of Parameters
Execution of Algorithm
Evaluation of Cluster Quality
Cluster Inspection
Persons:
Dates:
Named Entities
Acceptable Clusters
Unacceptable Clusters
Postprocessing:
Cleansing and Refinement of Clusters
Semantic Labeling of Acceptable Clusters
XML Tagging of Text Units
Creation of DTD and XML Documents
1
2
3
Text Unit Clusterer
XML Document Type Definition
XML Documents

Figure 1: Iterative and interactive KDT process

In the remainder of this section, we briefly introduce the first phase of the DIAsDEM framework whose iterative and interactive KDT process is depicted in Figure 1. This process is termed “iterative” because the clustering algorithm is invoked repeatedly. Our notion of iterative clustering should not be confused with the fact that most clustering algorithms perform multiple passes over the data before converging. This process is also “interactive”, because a knowledge engineer is consulted for cluster evaluation and final cluster naming decisions at the end of each iteration.

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 UML schema of the domain and descriptions of specific named entities of importance, e.g. persons and companies. The UML schema reflects the semantics of named entities and the relationships among them, as they are initially conceived by application experts. This schema serves as a reference for the DTD to be derived from discovered semantic tags, but there is no guarantee that the

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT CommercialRegisterEntry ( #PCDATA | BusinessPurpose | ShareCapital |
ModificationMainOffice | FullyLiablePartner | AppointmentManagingDirector |
GeneralPartnership | InitialShareholders | NonCashCapitalContribution |
LimitedLiabilityCompany | ConclusionArticles | ModificationRegisteredName |
SupervisoryBoard | (...) | Owner | FoundationPartnership )* >

<!ELEMENT BusinessPurpose (#PCDATA)>
<!ELEMENT ShareCapital (#PCDATA)> (...)
<!ELEMENT FoundationPartnership (#PCDATA)>
```

Table 2: Preliminary flat, unstructured XML DTD of Commercial Register entries

final DTD will be contained in or will contain this schema.

Similarly to a conventional KDD process, our process starts with a preprocessing phase: After setting the level of granularity by determining the size of text units to be tagged, the Java- and Perl-based *DIAsDEM Workbench* performs basic NLP preprocessing such as tokenization, normalization and word stemming using TreeTagger [22]. 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 UML 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 into Boolean vectors of this feature space. Additionally, named entities of interest are extracted from text units by a separate module of the *DIAsDEM Workbench*. In our case study, we created a small thesaurus and selected 70 relevant descriptors and 109 non-descriptors pointing to descriptors.

In the pattern discovery phase, all text unit vectors contained in the initial archive are clustered based on similarity of their content. 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 the *DIAsDEM Workbench* partitions into ”acceptable” and ”unacceptable” ones according to our quality criteria. A cluster of text unit vectors is ”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 stepwise decreased such that “acceptable” clusters become progressively less specific in content. The KTD process is based on a plug-in concept that allows the execution of different clustering algorithms within the *DIAsDEM Workbench*. In the case study, we employed the demographic clustering function included in the IBM Intelligent Miner for Data that maximizes the value of Condorcet’s criterion. After three iterations, the *DIAsDEM Workbench* discovered altogether 73 “acceptable” clusters containing approx. 85% of text units.

The postmining phase consists of a labeling step, in which “acceptable” clusters are semi-automatically assigned a label. Ultimately, cluster labels are determined by the knowledge engineer. However, the *DIAsDEM Workbench* performs both a pre-selection and a ranking of candidate cluster labels for the expert to choose from. All default cluster labels are derived from feature space dimensions (i.e. from text unit descriptors) that are prevailing in each “acceptable” cluster. Cluster labels actually correspond to XML tags that are subsequently used to annotate cluster members. Finally, all original documents are tagged using valid XML tags. Additionally, XML tags are enhanced by attributes reflecting previously extracted named entities and their values. Table 2 contains an excerpt of the flat, unstructured XML DTD that was automatically derived from XML tags in the case study. It coarsely describes the semantic structure of the resulting XML collection. Currently, named entities that serve as additional attributes of XML tags are not fully evaluated by the *DIAsDEM Workbench*.

ESTABLISHING A PROBABILISTIC DTD

The output of the *DIAsDEM Workbench* is a set of semantic XML tags which should be used as XML tags to describe the content of the archive documents. To reflect the content of the archive at an abstract level, it is essential to compose the tags into a DTD. Since the semantic annotations are derived with data mining techniques, they are not crisp. Thus, it is essential that the validity of each tag is expressed in quantitative terms and is estimated properly. Furthermore, an ordering should be imposed upon the tags. Hence, after deriving semantic XML tags, we combine them into a *probabilistic DTD* by (i) deriving the most likely ordering of the tags and (ii) computing the statistical properties of each tag inside the document type definition.

The reader may recall that a semantic annotation is actually the label of a cluster discovered by the *DIAsDEM Workbench*. The underlying clustering mechanism produces non-overlapping clusters. This implies that a text unit belongs to exactly one cluster, to the effect that it can be annotated with the label of this cluster only. Hence, the tags/labels derived the *DIAsDEM Workbench* cannot be nested. An extension of the *DIAsDEM Workbench* by a hierarchical clustering algorithm would allow for the establishment of subclusters and thus for the nesting of (sub)cluster labels. However, this is planned as future work.

The objectives of the DTD establishment method are the specification of the most appropriate ordering of tags, the identification of correlated or mutually exclusive tags and the adornment of each tag and each correlation among tags 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 we consider for the DTD tags and their associations and describe the methodology for computing these statistics. To model the complete statistical information pertinent in these tags and their relationships, we use a hypergraph structure. We then introduce a mechanism that derives a probabilistic DTD from this graph.

Statistical Properties of Semantic XML Tags

The statistical properties of DTD tags are depicted in Table 3 and described in the following paragraphs. The first column contains the names of the properties. The second column reflects whether the property is peculiar to the whole set of tags as cluster labels (i.e. the whole "model"), to each tag or to a group of associated tags. The last column names the mechanism to be applied to derive the value of each property for each tag.

Accuracy The *DIAsDEM Workbench* derives semantic XML tags as labels of clusters. These clusters constitute a *model* over the data, in the conventional statistical sense. In terms of data classification, such models are subject to *misclassification errors*. We identify two types of misclassification:

- *Error type I:* A text unit is assigned to the wrong cluster, i.e. the cluster label does not reflect the content of the text unit.
- *Error type II:* A text unit is not assigned to any cluster, although there is a cluster with a label reflecting the content of the text unit.

For the envisaged DTD, only the error type I is relevant. We use the term *accuracy* of the model as the probability that cluster labels reflect the content of cluster members. The accuracy value affects the DTD as a whole, it is not peculiar to individual tags. Therefore, we do not incorporate this value in the statistical adornment of the individual tags.

In order to evaluate the quality of out approach in absence of pre-tagged documents, we drew a random sample containing 5% out of 10,785 text units and asked a domain specialist to verify the annotations of these text units with respect to both error types. Within the sample, error type I (error type II) occured in 0.4% (3.6%) of text units. Hence, tagged text units are most likely to be correctly processed. The percentage of error type II text units is higher, indicating that some text units were not placed in the cluster they semantically belong to. With 0.95 confidence, the overall error rate in the entire dataset is in the interval [2.6%, 5.9%] which is a promising result.

TagSupport The tags of the DTD are cluster labels derived by a statistical approach. Thus, in terms of XML, they are observed as optional per se. In many application areas, a domain expert can provide suggestions as to which tags should be observed as mandatory. Despite this, there is no guarantee that the expert's suggestions hold true in the archive: The text unit containing this information may have been misclassified by the *DIAsDEM Workbench*, or the information may be simply absent from the document. For example, although one would expect that each movie has a regisseur, there are movies whose regisseur is unknown or inapplicable, due to the nature of the movie. The property *TagSupport* offers an indicator of whether a tag may be considered as potentially mandatory. We define it as the ratio of documents where this tag appears to the total number of documents in the archive.

AssociationConfidence In association rules' discovery, the miner identifies items occuring together. Equivalently, we are interested in tags that affect the appearance of other tags. We use the term *AssociationConfidence* for a tag x given the tags $y_1, \ldots, y_n$ in much the same way as confidence is defined for association rules [5]: It is the ratio of documents, where the tags $y_1, \ldots, y_n$ and x appear to the documents containing $y_1, \ldots, y_n$.

AssociationLift Similarly to the association rules' paradigm, the correlation among a tag x and a set of tags $y_1, \ldots, y_n$ can be spurious, caused by a very high support of x in the whole population. The statistic called lift or improvement is defined to alleviate this problem: it is the ratio of the *AssociationConfidence* of x given $y_1, \ldots, y_n$ to the *TagSupport* of x in the whole population [5]. In our case, this would be the ratio $\frac{AssociationConfidence(x, y_1, \ldots, y_n)}{TagSupport(x)}$.

LocationConfidence The aforementioned statistical properties on associated tags do not take the ordering of tags into account. In a DTD, the ordering of tags is essential. We use the term *LocationConfidence* of a tag x given the sequence of adjacent tags $y_1 \cdot y_2 \cdot \ldots \cdot y_n$ as the number of documents containing the sequence $y_1 \cdot y_2 \cdot \ldots \cdot y_n \cdot x$ to the number of documents containing $y_1 \cdot y_2 \cdot \ldots \cdot y_n$. This definition differs from the conventional statistics known for sequence mining [2], because we are concentrating on adjacent tags, disallowing the occurrence of arbitrary tags in-between. Conventional sequence mining do not satisfy this requirement. However, some Web usage miners have been designed to distinguish between adjacent and non-adjacent events [3, 9, 24, 20].

GroupSupport In most of the above statistics, we juxtapose the frequence of appearance of a tag with the frequency of a group of tags, be it a set or a sequence. We use the term

Property	*Radius*	*Computation method*
Accuracy	model	*DIAsDEM Workbench*
TagSupport	tag	simple statistics
AssociationConfidence	set of tags	association rule discovery
AssociationLift	set of tags	association rule discovery (ARD)
LocationConfidence	sequence of tags	sequence mining (SeqM)
GroupSupport	set or sequence of tags	ARD/SeqM

Table 3: Statistics for derived XML tags

GroupSupport as the ratio of the number of documents containing a group of tags to the total number of documents. In fact, 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 we model the statistical information pertinent to individual tags, to tag groups (i.e. sets or sequences) and to relationships among them in a seamless way.

Modeling Statistics of Associated XML Tags

Some of the values of the statistical properties depicted in Table 3 are already made available as part of the *DIAsDEM Workbench* output, while the remaining ones can be computed by data mining algorithms. To exploit these values for the establishment of a probabilistic DTD, we need a representation model and an algorithm that builds the DTD when processing this model. We introduce here a generic graph structure, in which all statistical information on tags, groups of tags and tag relationships is depicted. This structure is appropriate for the establishment of a DTD or an XMLschema with rich statistical adornments. In the next subsection, we discuss two algorithms for DTD establishment.

We represent the tags and their associations in a directed graph. Its nodes are individual tags, sequences of adjacent tags or sets of co-occuring tags. Each node is adorned with the statistical properties pertinent to a tag, resp. tag group. An edge represents a relationship of the form $y_1 \dots y_n \rightarrow x$; however, we use the convention that x is the *source* node and the group of nodes in the rule's LHS is the *target*. Similarly to nodes, an edge is adorned with the statistics of the order-insensitive or order-sensitive association it represents.

Semantic Tags as Graph Nodes Let A be the set of semantic XML tags derived by the *DIAsDEM Workbench*, and let $V \subseteq A \times (0, 1]$ be the set of graph nodes conforming to the signature:

$$< TagName, TagSupport >$$

By this definition, a tag can only appear in the graph if its *TagSupport* is more than zero. This is consistent with the fact that XML tags are derived with a KDD method.

For the groups of tags, we must distinguish among order-sensitive and order-insensitive groups. To do so, we perform three steps. First, we model tag groups as ordered lists. Second, we annotate each list with a flag that indicates whether the group depicted by the list is order-sensitive or order-insensitive; in the latter case, the ordering of the list is irrelevant but must be unique. Third, we guarantee uniqueness, i.e. that all permutations of the same group of tags are mapped into the same order-insensitive list, by requiring that order-insensitive list are lexicographically ordered.

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

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 of the two elements. The group $(< a, b >, 0)$ 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 (default) 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:

$$< GroupOfTags, GroupSupport >$$

where V' contains only those groups of annotations, for which the *GroupSupport* value is above a given threshold. This threshold can be specified as input to the mining software, as is usual in KDD applications, or may be set as low as 0. Of course, the threshold value affects the size of the graph and the execution time of the algorithm that traverses it to build the DTD.

Tag Relationships as Graph Edges The set of nodes constituting our graph is $\mathcal{V} = V \cup V'$, indicating that a node may be a singleton 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 a tag to an associated group of tags. Formally, the set of edges E is a subset of $(V \times V') \times \mathcal{X} \times \mathcal{X} \times \mathcal{X}$, where $\mathcal{X} := (0,1] \cup \{NULL\}$, with signature:

$$< Edge, AssociationConfidence, \\ AssociationLift, LocationConfidence >$$

In this signature, the statistical properties refer to the edge's source given the group of nodes in the edge's target. If the target is a sequence of adjacent tags, then the location confidence is the only valid statistical property, while the association confidence and lift are inapplicable. If the target is a set of tags, then the location confidence is inapplicable. When a statistical property is inapplicable, it assumes the NULL value.

Graph Properties The components of our "DTD-establishment graph" are tags, groups of tags and relationships among them, all adorned with statistical values. All tags discovered by the *DIAsDEM Workbench* are present in this graph. Which groups of tags are present depends on the threshold value for the group support. Conceivable are both a minimalistic approach with a high threshold, by which only very frequent groups are present, and a maximalistic approach with a zero-value threshold, by which all tag combinations occuring in the documents are present.

If we opt for the minimalistic approach, the graph will not be connected in the general case. It will contain only the groups of tags being more frequent than a threshold, and the frequent relationships among them. Certain tags may be isolated, because they only rarely appear in combination with other tags. Contrary to it, the maximalistic approach ensures that all combinations of tags appearing together in documents are depicted in the graph, and that the graph is connected, except of the unlikely case that some documents contain a single tag not occuring in any other documents.

The size of the graph depends on the threshold value for group support and for the confidence and lift values. The maximalistic approach delivers an upper limit. To compute it, let m be the number of tags/cluster labels output by the *DIAsDEM Workbench* and let $n \leq m$ be the largest number of distinct tags appearing in any document. For each tag, there are $\zeta_1 := \sum_{i=1}^{n-1} \binom{n}{i}$ perturbations to be considered, resulting in an equal number of order-sensitive groups and in $\zeta_2 := \frac{n(n+1)}{2}$ order-insensitive ones. There is one edge per (*Tag*,*Group*)-pair. Moreover, a tag participates in a maximum of $\zeta_1 + \zeta_2$ groups, thus resulting in $m + m \times (\zeta_1 + \zeta_2)$ graph nodes and $m \times (\zeta_1 + \zeta_2)$ edges.

The upper limit to the graph size indicates that threshold values for the statistical properties are essential for obtaining a manageable graph. On the other hand, each cutoff value implies an information loss. Therefore, we observe the DTD-establishment graph under the maximalistic approach as a *reference structure* and introduce two algorithms that derive a probabilistic DTD by constructing only a part of this graph.

DTD Derivation

The DTD-establishment graph in its maximalistic version captures all relationships among the semantic tags found by the *DIAsDEM Workbench*. Similarly to the process of schema establishment for a conventional database application, the designer must decide which relationships among the real-world entities are worth capturing and which are not. In our context, "worth capturing" refers to statistical values, presuming that a DTD should reflect the relationships usually present in the documents rather than the rare ones. However, the DTD-establishment graph contains relationships among sets and among sequences of tags, each one adorned with different (and only partially comparable) statistics.

In the following, we present two algorithms that derive a DTD by constructing part of the DTD-establishment graph. Each tag of this DTD is adorned by only two (derived) probabilistic values, one referring to the tag itself and one to its location inside the DTD. The algorithms are using different heuristics to derive this DTD: the first one concentrates on the pairs of tags appearing most frequently together, while the second one gives preference to maximal sequences of tags. The reader may recall that the computation of the statistics for the relationships among the tags require the activation of data mining software. Hence, each of the algorithms is backed by a miner that returns the desired statistics.

Backward Construction of DTD Sequences

This algorithm observes a DTD as a set of alternative sequences and builds each sequence backwards, starting at each last tag and proceeding until the first one. Concretely, the algorithm builds "maximal" sequences, where maximality means that the first tag of the sequence is the first tag in most of the documents supporting the sequence.

Backward expansion of tag-subsequences. The algorithm starts with an arbitrary tag $\tau \in A$ and then identifies the tag most likely to appear before τ:

- If no such tag exists, then the sequence cannot be expanded anymore. It is marked as *done* and the algorithm shifts to the next sequence that is not done yet, or to the next arbitrary tag from A, until all tags are processed.
- If there is a most likely predecessor of τ, say τ', it is prepended to the sequence. The next iteration starts, in which the most likely predecessor of $\tau' \cdot \tau$ (in general: of the subsequence built thus far) must be found.
- If there are k predecessor tags, none of which is more likely than the others, k alternative incomplete sequences are produced by duplicating the sequence built thus far. The algorithm processes them iteratively.

The predecessors of a tag τ can be found by invoking a sequence miner that returns all frequent sequences of adjacent tags. For the first iteration, the algorithm uses the frequent pairs $x_1 \cdot \tau, \ldots, x_u \cdot \tau$, each one leading to τ with a (location) confidence $c_1, \ldots, c_u$ respectively. Since these tags are the immediate predecessors of τ it holds that $\sum_{i=1}^{u} c_i = 1 - c_0$, where c_0 is the ratio of documents where τ has no predecessor divided by the total number of documents.

The maximum among $c_0, \ldots, c_u$ determines the rest of the procedure: If c_0 is maximum, τ is the first element of the sequence. In this case, the sequence is marked as "done" *and* as "maximal" according to the maximality criterion already mentioned.

If there is a c_i larger than the other elements, then x_i is the predecessor of τ in the sequence. However, it can be the case that the maximum is only marginally larger than the other values. In other words, there are k tags with $k \leq u$, such that (i) one of them has shows the maximum location confidence but (ii) the difference of this value from the location confidences of the other $k-1$ tags is less than some small ε. Then, all k tags are acceptable alternatives, resulting to k alternative subsequences.

Identifying maximal tag-sequences. In each iteration, the algorithm considers longer frequent sequences returned by the sequence miner, namely those ending with each subsequence already built. If no frequent sequence is found for a subsequence $s = \tau_1 \ldots \tau_k$, then s is "done" but it must also be checked whether it is maximal. This implies computing the ratio of documents starting with τ_1 over the whole number of documents and comparing this value x to the tag support of τ_1, say c. The comparison is performed across the same guidelines as for alternative tag predecessors: if $|x - c| \leq \varepsilon$, then most documents containing s start with τ_1 and thus s is maximal. Otherwise, documents starting with τ_1 mostly adhere to a different maximal sequence.

Statistics of maximal tag-sequences. At a final step, the algorithm filters out all sequences that are done but are not maximal. It then assigns probability values to each tag τ inside each maximal sequence s containing it:

- The *TagConfidence* is the tag's TagSupport multiplied by the accuracy of the model output by the *DIAsDEM Workbench*.
- The *TagPositionConfidence* is the location confidence of this tag with respect to the subsequence of s leading to it.

Backward versus forward sequence construction. The backward-sequence-construction method generates alternative sequences of DTD tags by pruning the frequent sequences of adjacent tags produced by a sequence miner. An equivalent method can be devised by forward-sequence construction. This would have the advantage of being appropriate for incorporation to a sequence miner's core as well, since most miners of this category perform forward sequence construction: In that case, the mining kernel would be modified to expand a sequence by the most likely successor tag only.

A DTD as a Tree of Alternatives

This algorithm observes a DTD as a tree of alternative subsequences and adorns each tag with its support with respect to the subsequence leading to it inside the tree: this is the number of documents starting with this subsequence of tags. Similarly to the sequence-construction algorithm described above, a tag may appear in more than one subsequences, having 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 assume a dummy root, the children of which are those tags that appear first in documents. In general, a tree node refers to a tag τ, and its children refer to the tags appearing after τ in the context of τ's own predecessors. In a sense, the DTD as a tree of alternatives resembles a DataGuide as proposed in [10], although the latter contains no statistical adornments.

The tree-of-alternatives differs from the sequence-construction algorithm in two ways: Firstly, it considers all sequences of tags that appear in documents instead of frequent ones only. Secondly, it only observes complete sequences, while a sequence miner returns arbitrary subsequences of tags.

The tree-of-alternatives method is realized by the preprocessor module of the Web usage miner WUM [25, 24]. This module is responsible for coercing sequences of events by common prefix and placing them in a tree structure, called "aggregated tree". This tree is input to the navigation pattern discovery process performed by the WUM core. The sequences of tags in documents can be observed as sequences of events, to the effect that the WUM preprocessor can also be used to build a DTD over an archive as a tree of alternative tag sequences. Figure 2 depicts an example of such a tree that related to our case study. Note that the XML document depicted in Table 1 is partly described by this DTD excerpt.

CONCLUSION

Most of the knowledge hidden in electronic media of an organization is encapsulated in documents. Acquiring this knowledge implies effective querying of the documents as well as the combination of information pieces from different textual assets. This functionality is usually confined to database-like query processors, while text search engines scan individual assets and return ranked results. In this study, we have presented a methodology that enables query processing and joining of text sources by structuring them. We propose the derivation of an XML DTD over a domain-specific text archive by means of data mining techniques.

The semantic characterization of text units is the core of our approach as well as the derivation of XML tags from these characterisations. This is undertaken by the *DIAsDEM Workbench* which is concisely described in the first part of this study. Our main emphasis is on combining these tags that reflect the semantics of many text units across the archive into a single DTD that reflects the semantics of the archive as a whole. We have shown that this DTD is a probabilistic ap-

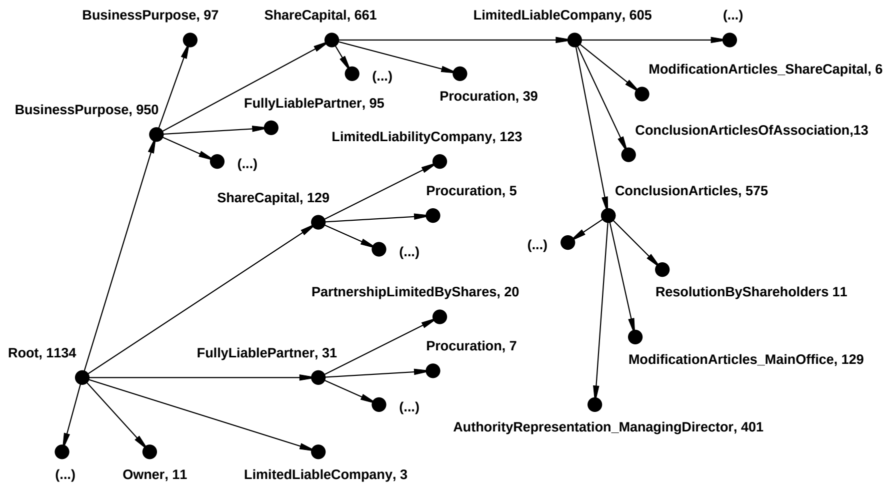
BusinessPurpose, 97
ShareCapital, 661
LimitedLiableCompany, 605
(...)
(...)
ModificationArticles_ShareCapital, 6
Procuration, 39
BusinessPurpose, 950
FullyLiablePartner, 95
ConclusionArticlesOfAssociation,13
(...)
LimitedLiabilityCompany, 123
ShareCapital, 129
Procuration, 5
ConclusionArticles, 575
(...)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

Figure 2: A DTD as a tree of alternative tag sequences

proximation of the archive content and have derived a set of statistical properties that reflect the quality of this approximation, for the whole DTD, for tags inside the DTD and for relationships among these tags.

The statistical properties of tags and of their relationships form the basis for combining them into a complete DTD in the XML sense or even into an XMLschema. We use a graph structure to depict all statistics that can serve as a basis for this operation and propose two mechanisms that derive DTDs by employing a mining algorithm and a set of heuristic rules. We have tested our methodology on an archive of documents from a regional Commercial Register in Germany: We have derived a set of tags with the *DIAsDEM Workbench* and then implemented one of the proposed mechanisms to derive a DTD for it.

Our future work includes the implementation of the second mechanism for DTD derivement and the establishment of a framework for the comparison of derived DTDs in terms of expressiveness and accuracy. Of course, the ultimate goal of our work is the establishment of a full-fledged querying mechanism over the text archives. To this purpose, we intend to couple our DTD derivation methods with a query mechanism for semi-structured data. Since the DTDs we derive are of probabilistic nature, this implies also the design of a model that evaluates the quality of the query results.

ACKNOWLEDGMENTS

We thank the German Research Society for funding the project DIAsDEM, the Bundesanzeiger Verlagsgesellschaft mbH for providing data and our project collaborators Evguenia Altareva and Stefan Conrad for helpful discussions. The IBM Intelligent Miner for Data is kindly provided by IBM in terms of the IBM DB2 Scholars Program.

REFERENCES

1. S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufman Publishers, San Francisco, 2000.

2. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of Int. Conf. on Data Engineering*, Taipei, Taiwan, Mar. 1995.

3. M. Baumgarten, A. G. Büchner, S. S. Anand, M. D. Mulvenna, and J. G. Hughes. Navigation pattern discovery from internet data. In *[17]*, pages 70–87. 2000.

4. M. Becker, J. Bedersdorfer, I. Bruder, A. Düsterhöft, and G. Neumann. GETESS: Constructing a linguistic search index for an Internet search engine. In *Proceedings of the 5th International Conference on Applications of Natural Language to Information Systems*, Versailles, France, June 2000.

5. M. J. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., 1997.

6. P. Buneman. Semistructured data. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 117–121, Tucson, AZ, USA, May 1997.

7. M. Erdmann, A. Maedche, H.-P. Schnurr, and S. Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. *ETAI Journal - Section on Semantic Web*, 6, 2001. To appear.

8. R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir. Text mining at the term level. In *Proceedings of the Second European Symposium on Principles of Data Mining and*

Knowledge Discovery, pages 65–73, Nantes, France, September 1998.

9. W. Gaul and L. Schmidt-Thieme. Mining web navigation path fragments. In *[13]*, 2000.

10. R. Goldman and J. Widom. DataGuides: Enabling query formulation and optimization in semistructured databases. In *VLDB'97*, pages 436–445, Athens, Greece, Aug. 1997.

11. H. Graubitz, M. Spiliopoulou, and K. Winkler. The DIAsDEM framework for converting domain-specific texts into XML documents with data mining techniques. In *Proceedings of the First IEEE International Conference on Data Mining*, San Jose, CA, USA, November/December 2001. To appear.

12. H. Graubitz, K. Winkler, and M. Spiliopoulou. Semantic tagging of domain-specific text documents with DIAsDEM. In *Proceeding of the 1st International Workshop on Databases, Documents, and Information Fusion (DBFusion 2001)*, pages 61–72, Magdeburg, Germany, May 2001.

13. R. Kohavi, M. Spiliopoulou, and J. Srivastava, editors. *KDD'2000 Workshop WEBKDD'2000 on Web Mining for E-Commerce — Challenges and Opportunities*, Boston, MA, Aug. 2000. ACM.

14. P. A. Laur, F. Masseglia, and P. Poncelet. Schema mining: Finding regularity among semistructured data. In D. A. Zighed, J. Komorowski, and J. Żytkow, editors, *Principles of Data Mining and Knowledge Discovery: 4th European Conference, PKDD 2000*, volume 1910 of *Lecture Notes in Artificial Intelligence*, pages 498–503, Lyon, France, September 2000. Springer, Berlin, Heidelberg.

15. S. Loh, L. K. Wives, and J. P. M. d. Oliveira. Concept-based knowledge discovery in texts extracted from the Web. *ACM SIGKDD Explorations*, 2(1):29–39, 2000.

16. J. Lumera. Große Mengen an Altdaten stehen XML-Umstieg im Weg. *Computerwoche*, 27(16):52–53, 2000.

17. B. Masand and M. Spiliopoulou, editors. *Advances in Web Usage Mining and User Profiling: Proceedings of the WEBKDD'99 Workshop*, LNAI 1836. Springer Verlag, July 2000.

18. A. Mikheev and S. Finch. A workbench for acquisition of ontological knowledge from natural language. In *Proceedings of the Seventh conference of the European Chapter for Computational Linguistics*, pages 194–201, Dublin, Ireland, March 1995.

19. U. Y. Nahm and R. J. Mooney. Using information extraction to aid the discovery of prediction rules from text. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, pages 51–58, Boston, MA, USA, August 2000.

20. A. Nanopoulos, D. Katsaros, and Y. Manolopoulos. Effective prediction of web-user accesses: A data mining approach. In *Proceeding of the Workshop WEBKDD 2001: Mining Log Data Across All Customer Touch-Points*, San Francisco, CA, USA, August 2001. To appear.

21. S. Nestrov, S. Abiteboul, and R. Motwani. Inferring structure in semi-structured data. *SIGMOD Record*, 26(4):39–43, 1997.

22. H. Schmid. Probabilistic part–of–speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, September 1994.

23. A. Sengupta and S. Purao. Transitioning existing content: Inferring organization-spezific document structures. In K. Turowski and K. J. Fellner, editors, *Tagungsband der 1. Deutschen Tagung XML 2000, XML Meets Business*, pages 130–135, Heidelberg, Germany, May 2000.

24. M. Spiliopoulou. The laborious way from data mining to web mining. *Int. Journal of Comp. Sys., Sci. & Eng., Special Issue on "Semantics of the Web"*, 14:113–126, Mar. 1999.

25. M. Spiliopoulou and L. C. Faulstich. WUM: A Tool for Web Utilization Analysis. In *extended version of Proc. EDBT Workshop WebDB'98*, LNCS 1590, pages 184–203. Springer Verlag, 1999.

26. A.-H. Tan. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, pages 65–70, Beijing, China, April 1999.

27. K. Wang and H. Liu. Discovering structural association of semistructured data. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):353–371, May/June 2000.