Integrating Data and Probabilistically Structured Text Documents

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

Abstract

Commercial, non-profit and public organizations are accumulating huge amounts of electronically available text documents. Although composed of unstructured texts, documents contained in archives such as annual reports to shareholders, medical patient records and public announcements often share an inherent, though undocumented structure. In order to enable information integration of text collections with related structured data sources, this inherent structure should be made explicit as detailed as possible. The goal of this study is the establishment of a methodology for the integration of text documents with structured records into a hyper-archive of application-specific entities. The text documents are of implicit structure which has been explicated by data mining techniques as proposed in the DIAsDEM framework for semantic tagging of domain-specific text documents. The result is a probabilistic DTD that serves as a basis for the matching of schemata and for the matching of data instances.

1 Introduction

Organizations are seeking for ways to explicate, store and disseminate their corporate knowledge. This formidable task involves many "soft" issues like motivating people to contribute information into a common pool and to exploit the pool in their everyday activities, as well as technical issues. One of the latter is the efficient acquisition of information. Despite the many achievements in search engine functionality, two major problems are not alleviated yet: Firstly, people think in terms of application objects and not of keywords in text corpora. Secondly, information about an application object is rarely located in a single data source. Regularly, assets from multiple sources must be combined to form the application object desired by the user.

In this study, we consider the second issue. The combination of assets to form an aggregate object is a data integration task involving multiple sources. As an example, consider a company employee as a composite application entity: The personnel database surely contains all data on employees as required for payroll and tax reports. The qualifications of each employee may (or may not) be stored in HR databases, in internal yellow-pages catalogues (as used e.g. in Microsoft) or in directories of Web pages for the organization's intranet. The projects in which this employee has been involved contain further information

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

about her. This information is mostly contained in a projects' database that is often maintained by the employee's department. The project reports she has authored or co-authored also bear information about her background, activities and expertise.

We propose a preliminary methodology for the integration of documents from a text collection with records from a structured database into composite application objects, which we call "application entities" or simply "entities". Firstly, we derive a probabilistic DTD for the text archive and discuss how this DTD can be used for schema integration. Secondly, we address the issue of finding the best n document matches for a database record by exploiting the derived semantic annotations that describe them.

In the next section we discuss advances on the derivation of schema-like text descriptions for document archives and on the integration of schemata over federated sources. In section 3, we provide a concise description of the DIAsDEM Workbench, which we use for DTD derivation from texts. In section 4, we define the notion of a hyper-archive over text and data sources and discuss first the issue of integrating database schemata and DTDs and then the issue of linking a database record to related documents. The last section concludes the study.

2 Related Work

Nahm and Mooney propose the combination of methods from KDD and information extraction to perform text mining tasks [18]. 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 [13]. 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 [16]. As 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 an XML document type definition. 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: Bruder et al. introduce the search engine GETESS that supports query processing on texts by deriving and processing XML text abstracts. These abstracts contain language-independent, content-weighted summaries of domain-specific texts [3]. In DIAsDEM, we do not separate metadata 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 documents is more appropriate than inferences on their content. In order to transform existing contents into XML documents, Sengupta and Purao propose a method that infers DTDs by using already tagged documents as input [21]. 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 [14]. 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 [4, 1]. A lot of effort has recently been put into methods inferring and representing structure in similar semi-structured documents [19, 24, 12]. 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.

The research project DIAsDEM aims at the integration of textual archives with both textual and structured data sources. Doan et al. present the LSD system that employs machine learning techniques to semi-automatically discover semantic mappings between source schemata and an existing mediated schema [7]. The authors assume data sources to be XML documents that are described by DTDs. In this context, the schema matching problem aims at finding correspondences between elements of the mediated schema and the source DTDs. However, the authors restrict themselves to discover one-to-one mappings between tags of the source and the mediated schema. Miller et al. introduce an interactive schema mapping creation paradigm to transform source data into a given target schema [17]. It is based on the notion of value correspondences that show how the value of a target attribute can be derived from a set of source attributes. Clio is a research prototype that implement this approach by incrementally creating SQL queries realizing the mappings implied by value correspondences [11]. Madhavan et al. review the wide range of current trends in schema matching and propose a taxonomy of schema matching approaches [15]. The authors present the generic schema matching approach Cupid that discovers mappings between schema elements (e.g., in XML DTDs) based on element names and data types as well as constraints and schema structure. The Cupid approach combines both linguistic and structure similarity into a weighted similarity value for each pair of schema elements that is subsequently used to determine a matching between the most similar elements in both schemata.

In contrast to the previously discussed schema matching approaches, DIAsDEM derives a probabilistic XML document type definition for a text archive due to the employed KDD process for DTD derivation. Altareva and Conrad introduce the problem of uncertainty during the integration of structured schemata [2]. The authors identify three types of uncertainties caused by application of data mining techniques to aquire metadata about legacy data sources. In contrast to integration of structured schemata, we consider data integration based on probabilistic DTDs of text archives and a related relational schema.

3 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

```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

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 contents 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.

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 document type definition (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 [10, 9] 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.

3.1 Derivation of unstructured DTDs

In 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

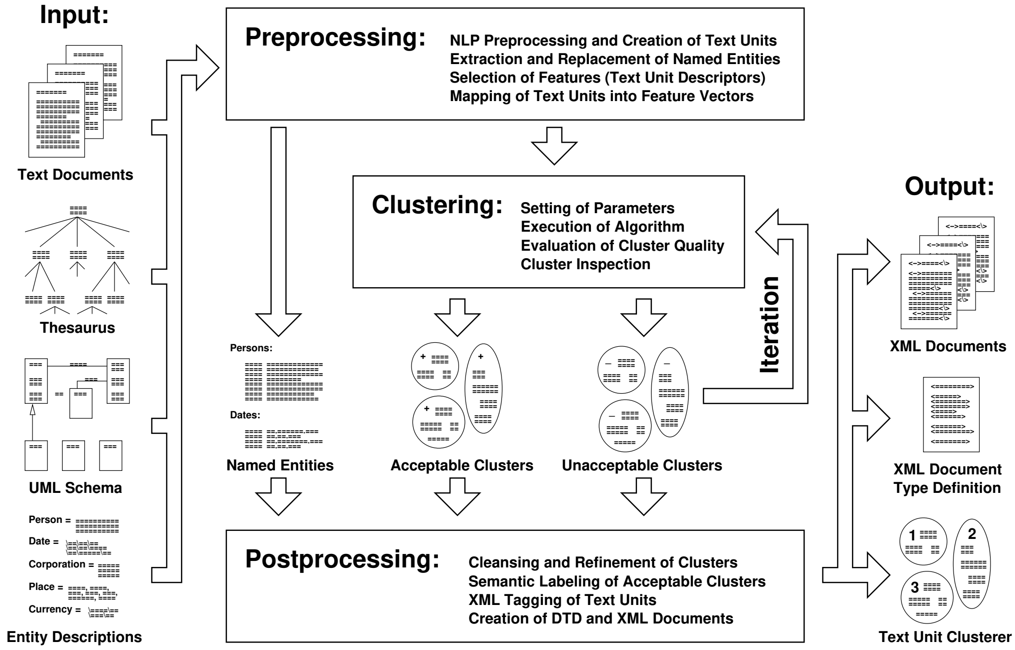

Figure 1: Iterative and interactive KDT process of the DIAsDEM framework

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 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 [20]. 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 contents. 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 KDT 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 and thus preliminary 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.

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 the following error types:

- *Error type I:* A text unit is annotated with a wrong XML tag, i.e. the tag does not properly reflect the contents of the text unit.
- *Error type II:* A text unit is not annotated at all, although there exists an XML tag in the derived DTD reflecting the contents of the text unit.

Within the sample, error type I (error type II) occurred 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.

```
<?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

3.2 Derivation of structured and probabilistic DTDs

As summarized in the previous section, the DIAsDEM Workbench currently derives a flat, unstructured DTD that semantically describes an archive of XML documents. However, this rather preliminary DTD is not sufficient for subsequent information integration. All tags contained in this flat DTD were derived using data mining techniques. Thus, they are not crisp and need to be annotated with quantitative measures of validity. Additionally, an ordering should be imposed upon the tags to structure the DTD. Hence, we introduced the notion of a structured, *probabilistic DTD* that includes (i) the most likely ordering of tags and (ii) certain relevant statistical properties of each tag inside the DTD [25]. Although the resulting probabilistic DTD is structured, nested XML tags cannot be discovered at the moment due to the nature of the iterative clustering phase.

The objectives of establishing a probabilistic DTD 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 query results. In particular, we defined the following statistical properties of DTD tags that are relevant in terms of the solution introduced for data integration in this study:

- The *Accuracy* is the probability that an XML tag correctly reflects the content of its text unit. This notion refers refers to error type I as defined in section 3.1. It is determined by domain experts who are supported by the DIAsDEM Workbench. The *Accuracy* value affects the DTD as a whole instead of being peculiar to individual tags.
- The *TagSupport* of XML tag x is defined by the ratio of XML documents that contain x to the total number of documents in the archive. This statistical property can be computed by simple frequency counts and it is peculiar to tag x. *TagSupport* is an indicator of whether an XML tag might be considered as mandatory in the DTD.
- The *AssociationConfidence* of XML tag x given the set of tags $y_1, \ldots, y_n$ is defined by the ratio of XML documents that contain the tags $y_1, \ldots, y_n$ and x to the documents containing $y_1, \ldots, y_n$. This statistical property can be computed by association rule discovery. *AssociationConfidence* is used to identify correlated tags within the archive.
- The *LocationConfidence* of tag x given the sequence of adjacent tags $y_1 \cdot y_2 \cdot \ldots \cdot y_n$ is defined by the ratio of XML documents that contain the sequence $y_1 \cdot y_2 \cdot \ldots \cdot y_n \cdot x$ to the documents containing $y_1 \cdot y_2 \cdot \ldots \cdot y_n$. This statistical property takes ordering of tags into account and can thus be discovered by sequence mining.

We proposed to employ a directed graph to represent a probabilistic DTD in [25]. Its nodes are individual XML tags, sequences of adjacent XML tags or sets of co-occurring XML tags. Each node is adorned with statistical properties pertinent to a tag, a set or a sequence of tags. An edge represents a relationship of the form $y_1 \ldots y_n \rightarrow x$. Similarly to nodes, an edge is adorned with the statistics of the order-insensitive or order-sensitive association it represents. This probabilistic DTD reflects the relationships usually present in documents rather than rare ones. We introduced two algorithms that derive a probabilistic DTD by constructing a part of the directed DTD-establishment graph. The algorithms utilize different heuristics for the DTD derivation: the first one (i.e. *Backward Construction of DTD Sequences*) concentrates on pairs of tags appearing most frequently together and the second one (i.e. *DTD as a Tree of Alternatives*) gives preference to maximal sequences of tags.

In this paper, the second algorithm will be employed for data integration. It is assumed that the DIAsDEM Workbench was previously used to derive the preliminary unstructured

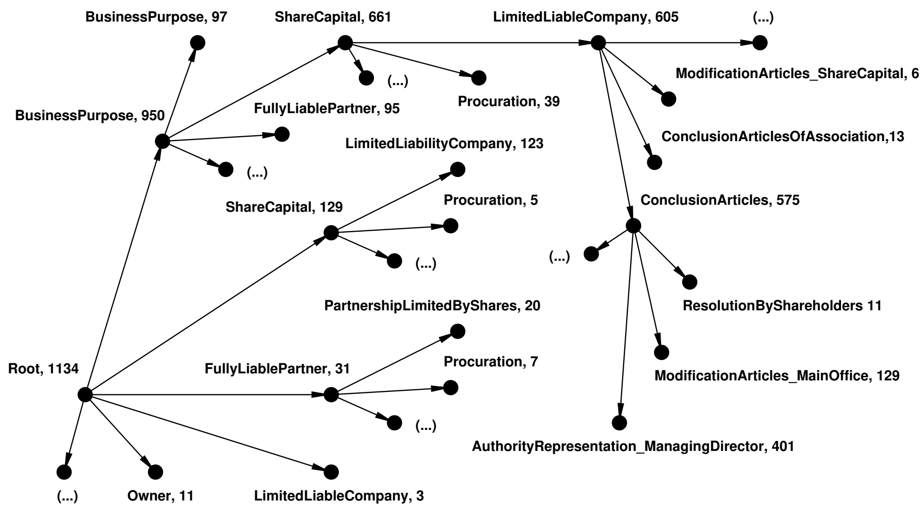

Figure 2: A DTD as a tree of alternative tag sequences

DTD for the corresponding archive and to semantically tag all documents. 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. 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 with a different tag. We assume a dummy root, whose children are all first tags 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. This method is realized by the preprocessing module of the Web usage miner WUM [23, 22] which is responsible for coercing sequences of events by a common prefix and placing them in a tree structure (i.e. the "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 excerpt of such a tree that is related to our case study. Note that the XML document depicted in Table 1 is partly described by this DTD excerpt.

4 A Scheme for Data and Text Integration

We study the issue of information integration in the context of an information system comprised of unstructured text collections and structured databases. We assume that the goal of integration is the specification of application objects whose contents are obtained by *combining* information assets from multiple underlying sources.

4.1 A Hyper-Archive of Application Entities

We use the term "hyper-archive" to denote a special type of federation over such sources: A hyper-archive models application objects which are comprised of information assets in the underlying sources. Thus, the hyper-archive plays the role of a view over the sources, ensuring that the individual assets are combined properly to constitute application objects.

Hyper-archive entities have structured, unstructured and semi-structured components.

Thus, the hyper-archive schema should be conceived as a combination of a rigidly structured database schema and a previously derived, non-crisp DTD. Object-oriented data models and possibly XML Schema can be used to describe the hyper-archive schema. A terminological alignment, at least at the schema level is necessary before aggregating the individual assets.

4.2 Alignment of Databases and Archives

In the general case, schema integration involves the identification of schema elements referring to the same real-world entity, the detection and treatment of structural and semantic heterogeneities. If one of the sources is a text archive, no structural heterogeneities occur, but the semantic discrepancies must be resolved between schema and text thesaurus.

The DIAsDEM Workbench approach reduces this problem into the alignment of semantic tags and database attributes: The text units of each document are assigned to clusters, the cluster labels become the XML tags that annotate the documents. Although homonyms and synonyms may still occur at the data/text level and need to be dealt with when combining instances, a first set of heterogeneity-resolving rules can be established by comparing database schemata and the semantic tags output by the DIAsDEM Workbench.

The alignment process involves (i) the schema attributes, including class names, for the crisp part of the hyper-archive schema and (ii) the semantic annotations over the text archive. The latter are individual tags rather than tag combinations, because the current version of the DIAsDEM Workbench does not support the nesting of annotations. For each pair (*SetOfAttributes*, *SetOfTags*) that describe the same aspect of the hyper-archive entity, a set of mapping rules should be formulated, which effectively "translate" the attributes to the tags and vice versa.

The alignment of (*SetOfAttributes*, *SetOfTags*)-pairs to resolve heterogeneities of linguistic nature is only one type of alignment necessary in schema integration. In the general case, all pairs of schema elements describing the same (component of a) real-world entity must be detected and mapped to each other. The literature on schema integration suggests different approaches to this purpose. They can be applied both for the necessary linguistic alignment and for complete schema matching. In any case, the schema/tags alignment mechanism must take the probabilistic nature of the derived tags into account.

We stress explicitly the constraint-based approach of [5, 6] in which rules for heterogeneity resolution and mapping take the form of constraints over the federated information system and the semi-automated schema matching approach of Cupid [15] which includes a module for linguistic matching of terms and deals both with schemata and DTDs.

Mappings rules on probabilistic DTDs. A mapping between a schema element and a DTD must be assigned a validity value. If the mapping is expressed as a rule, this is the confidence of the rule. The confidence of a rule on a set of tags $\tau_1, \ldots, \tau_n$ is computed as the product of:

- the ratio of documents where all these tags appear to all documents in the archive, i.e. the *confidence* of the group of tags inside the archive
- the accuracy of the model output by the DIAsDEM Workbench

The first factor of this product considers the joint appearance of tags in the documents, irrespectively of their order in the document body. If the ordering is essential for the relationship among these tags, the above confidence for the set of tags should be replaced by the confidence for the ordered list of tags. In [25], we give a detailed description on how the confidence of order-sensitive and order-insensitive relationships among tags can be computed.

Matching schema and DTD subcomponents. Cupid considers linguistic and structural similarities between two schema elements and computes a weighted similarity among them. The weighted similarity of complex schema elements is computed from the similarities of their components in a recursive way [15]. In our context, matching between schema elements and DTD subcomponents (sets or sequences of tags) requires the weighting of each structural similarity value with the confidence of the tag-group it refers to.

The alignment of schemata and DTDs by means of mapping rules and/or schema matching ensures that schema similarities can be exploited during data integration. Data integration takes the form of matching structured records with text entries and expanding the best matches into hyper-archive entities.

4.3 Matching Records with Text Entries

For the aggregation of data and text entries to form the individual entities, linguistic alignment is necessary, so that schema attributes and XML tags are at least comparable. A thorough schema integration however does not remove the necessity for performing data integration at the instance level: Firstly, the derived DTD is only partially supported by individual documents. Secondly, some of the information needed for integration is located in the content rather than in the tags of documents.

Scenario and assumptions. We investigate one particular case of data integration, in which a hyper-archive entity is comprised of *one* database record and a set of documents that enrich the record with additional information. We make the assumptions that (a) the database has a crisp schema and (b) the record is built from the attributes of a well-defined view over multiple tables or classes and has a well-defined primary key. Hence, the database records identify the application entities in a unique way, and the goal of the text archive integration is the enrichment of the database records with additional information "hidden" in text entries. This is a quite realistic scenario, especially for intra-organizational applications requiring a complete description of a well-defined entity, like employee, customer or project.

It should be stressed that the two assumptions determine whether a hyper-archive entity can be described by a single record. They imply that discrepancies in the business logic should be resolved before attempting the integration with the documents. Such discrepancies may occur, if e.g. the marketing and the accounting departments of a company have different definitions of what a customer is, resulting in database records that cannot be integrated.

Record schema and DTD components. Let $\mathcal{D}$ be the database of records to be enhanced with textual entries. $\mathcal{D}$ adheres to a crisp signature of typed attributes $< a_1, \ldots, a_n >$. We assume that $K := \{a_1, \ldots, a_m\}$, with $m \leq n$, is the set of prim attributes. Then, we denote by K' the set $\{a_{m+1}, \ldots, a_n\}$. For a record $r \in \mathcal{D}$, we denote by $r.a_i$ the value of r for attribute a_i. If $X \subseteq K \cup K'$, we denote by $r(X)$ the projection of r over the attributes in X.

Let $\mathcal{T}$ be a text archive. We assume that the linguistic alignment between the schema of $\mathcal{D}$ and the probabilistic DTD over $\mathcal{T}$ has been performed. Thus, we can observe an attribute and a tag that refer to the same concept as being lexicographically identical.

Let $d \in \mathcal{T}$ be a document described by the sequence of tags $T_d = \tau_1 \cdot \ldots \cdot \tau_k$, some of which may incorporate (attribute,value)-pairs derived by the NEEX module of the DIAsDEM Workbench. We denote by $d.\tau_i$ the text unit of d marked by τ_i, and by $d.\tau_i : a$ the value of d for the attribute a inside the tag τ_i. The confidence of each tag τ_i inside d is the number of documents described by the tag-sequence $\tau_1 \cdot \ldots \tau_i \cdot y$ with y an arbitrary subsequence of tags, divided by the number of documents in $\mathcal{T}$. This confidence can be acquired by traversing the root children of the tree DTD until the branch $\tau_1 \cdot \ldots \cdot \tau_i$ is found.

Matching tags, attributes and contents. The procedure of expanding *one* record $r \in \mathcal{D}$ with textual information involves testing several matching candidates. Since this procedure has to be performed for each record in $\mathcal{D}$, it is essential that the number of candidates per record be kept low and that the testing phase be kept short.

Firstly, we observe that the only appropriate candidate documents are those referring to the same entity as r. This fact cannot be assessed with certainty. However, we can exploit the semantic annotations found by the DIAsDEM Workbench. Especially, let K_d be the set of tags in T_d that either correspond to the prim attributes of K in terms of the linguistic alignment, or contain attributes that correspond to those prim attributes. In a similar way, we define K'_d for the non-prim attributes in $\mathcal{D}$. Either of K_d, K'_d may be empty. Using these sets, we consider the following documents as candidates:

I: each document d, such that there is $\tau \in K_d$ with $d.\tau$ containing the same values as r for the prim attributes to which τ corresponds, either in the enclosed text unit or in the attributes adorning τ itself

II: each document d, such that there is $\tau \in K'_d$ with $d.\tau$ containing the same values as r for the non-prim attributes to which τ corresponds, either in the enclosed text unit or in the attributes adorning τ itself

III: each document d containing some of the values in $r.K$ in its body

IV: each document d containing some of the values in $r.K'$ in its body

Obviously, the four groups return candidates of different quality. For example, a document containing the value of $r.a_i$ with $a_i \in K$ inside a tag corresponding to a_i is a more liable candidate than a document containing a word that happens to be equal to a prim attribute value for r.

In a second step, we identify candidates that have no other tags than those corresponding to attributes of $\mathcal{D}$. The information contained in these documents cannot be exploited well in the hyper-archive, since it lacks structure and its quality cannot be assessed. These candidates form Group B (with subgroups B.I-B.IV). Candidates with further tags form Group A (with subgroups A.I-A.IV).

In a third step, we quantify the similarity of documents in each group to the record r. For the quantification, we use a base function f, which compares the value of r for attribute $a_i \in K \cup K'$ with some string s. According to the four types of candidates above, this string can be a tag's content, the content of a tag's attribute or an arbitrary document string. Then, the value of f is:

$$f(r.a_i, s) = \begin{cases} 1 & r.a_i = s \\ -1 & r.a_i \neq s \end{cases}$$

This function assigns to all matches a positive score, while mismatches get a negative score.

Subsequently, the matches inside the groups A.I, A.II and B.I, B.II should be weighted with the confidences of the tags involved in the matching. For a document d, a tag $\tau \in T_d$ may contribute to several matches of database attributes, by means of the content it encloses and of the attributes it is adorned with. Let S_τ be the set of strings contributed by τ and let $c(d, \tau)$ be the confidence of τ in its location inside document d. Then, the contribution of τ is computed as:

$$contrib(r, d, \tau) = c(d, \tau) \times \sum_{s \in S_\tau} \left(\sum_{a_i \in K \cup K'} f(r.a_i, s) \right)$$

In this formula, we compute the contribution of a tag to the values of all attributes in the schema of $\mathcal{D}$. The formula is subject to the assumption that the ordering of the tags in the document should be taken into account. If this is not the case (and in many applications it should not be), then the location confidence of τ inside d should be replaced by the support of τ in the archive, i.e. by the ratio of documents containing τ to the total number of documents in $\mathcal{T}$.

Finally, the similarity of each document d to the record r is computed as the sum:

$$sim(r,d) = \sum_{\tau \in T(d)} contrib(r,d,\tau)$$

The complete procedure of document selection and similarity computation produces four ordered groups of matching documents. If the goal of the similarity matching procedure is to select the best N matches rather than finding all matches and ranking them, then the similarity values can be computed sequentially from group A.I to group B.II, so that computations stop as soon as the N best matches are found.

5 Conclusion

In this study, we have summarized the DIAsDEM framework for semantic tagging of domain-specific texts. We have employed the Java- and Perl-based DIAsDEM Workbench to derive semantic tags that describe the text units in an archive of German Commercial Register entries. These tags have been combined into a probabilistic DTD over the archive by the sequence miner WUM. However, Commercial Register entries are composed of rather regular and antiquated German language which might contribute to the low overall error rates in the case study. Therefore, we are currently working on a different application domain characterized by a greater linguistic diversity: Ad hoc news are issued by publicly quoted companies and contain information about current developments that potentially influence share prices. Both stakeholders and public authorities pursuing investor protection, market transparency and market integrity have a particular interest in this public source of information. This latest case study is not finished yet. However, the preliminary results of evaluating the tagging quality are once again very promising.

As the main contribution of this paper, we have presented a preliminary methodology for the integration of data with semantically annotated text documents. Our approach is based on the DIAsDEM framework. We have considered the goal of establishing a hyper-archive of application entities and have shown how this probabilistic DTD can be used to combine documents with database records into the target hyper-archive entities. Our mechanism performs similarity matching between the textual content enclosed by derived tags and the values of database attributes for a given target record, whereby the non-crisp nature of the tags is quantified and taken into account. Our mechanism builds eight groups containing candidate documents of decreasing matching quality. Inside each group, it computes the similarity of each document to the target and ranks the documents.

Our approach of integrating structured data and probabilistically structured text documents is preliminary in nature. Thus, the first priority of our future work is its implementation within the DIAsDEM Workbench, experimental validation and comparison with other methods from the area of record linkage. Moreover, we anticipate that the ranking scheme lacks a baseline, since the range of the ranking values is unbounded and the notion of distance between two consecutive rank values is not defined. Therefore, we intend to consider alternative schemes that do not have this shortcoming and thus are more appropriate for comparative experiments.

6 Acknowledgments

We thank the German Research Society for funding the project DIAsDEM and the Bundesanzeiger Verlagsgesellschaft mbH for providing data. 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] E. Altareva and S. Conrad. The problem of uncertainty and database integration. In *Proceedings of the Workshop on Engineering Federated Information Systems (EFIS 2001)*, Berlin, Germany, October 2001. To appear.

[3] I. Bruder, A. Düsterhöft, M. Becker, J. Bedersdorfer, and G. Neumann. GETESS: Constructing a linguistic search index for an Internet search engine. In *Proceedings of the Fifth International Conference on Applications of Natural Language to Information Systems*, pages 227–238, Versailles, France, June 2000.

[4] 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.

[5] S. Conrad, I. Schmitt, and C. Türker. Behandlung von Integritätsproblemen bei Schemastrukturierung und Schemaintegration. In K. R. Dittrich and A. Geppert, editors, *Datenbanksysteme in Büro, Technik und Wissenschaft*, Informatik aktuell, pages 352–369. Springer-Verlag, Berlin, 1997.

[6] S. Conrad, I. Schmitt, and C. Türker. Considering integrity constraints during federated database design. In S. M. Embury, N. J. Fiddian, A. W. Gray, and A. C. Jones, editors, *Advances in Databases, 16th British National Conference on Databases, BNCOD 16, Cardiff, Wales, July 1998*, volume 1405, pages 119–133, Berlin, 1998. Springer-Verlag.

[7] A. Doan, P. Domingos, and A. Y. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data on Management of Data*, pages 509–520, Santa Barbara, CA, USA, May 2001.

[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] 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.

[10] 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.

[11] M. A. Hernández, R. J. Miller, and L. M. Haas. Clio: A semi-automatic tool for schema mapping. In *Proceedings of the 2001 ACM SIGMOD International Confer-*

ence on Management of Data on Management of Data, page 607, Santa Barbara, CA, USA, May 2001.

[12] 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.

[13] 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.

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

[15] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proceedings of 27th International Conference on Very Large Data Bases*, Roma, Italy, September 2001. To appear.

[16] 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.

[17] R. J. Miller, L. M. Haas, and M. A. Hernández. Schema mapping as query discovery. In *Proceedings of 26th International Conference on Very Large Data Bases*, pages 77–88, Cairo, Egypt, September 2000.

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] K. Winkler and M. Spiliopoulou. Extraction of semantic XML DTDs from texts using data mining techniques. Submitted for publication, July 2001.