The DIAsDEM Framework for Converting Domain-Specific Texts into XML Documents with Data Mining Techniques

Henner Graubitz,* Myra Spiliopoulou and Karsten Winkler*
Leipzig Graduate School of Management (HHL)
Department of E-Business
Jahnallee 59, D-04109 Leipzig, Germany
{graubitz,myra,kwinkler}@ebusiness.hhl.de

Abstract

Modern organizations are accumulating huge volumes of textual documents. To turn archives into valuable knowledge sources, textual content must become explicit and queryable. Semantic tagging with markup languages such as XML satisfies both requirements. We thus introduce the DIAsDEM framework for extracting semantics from structural text units (e.g., sentences), assigning XML tags to them and deriving a flat XML DTD for the archive. DIAsDEM focuses on archives characterized by a peculiar terminology and by an implicit structure such as court filings and company reports. In the knowledge discovery phase, text units are iteratively clustered by similarity of their content. Each iteration outputs clusters satisfying a set of quality criteria. Text units contained in these clusters are tagged with semi-automatically determined cluster labels and XML tags respectively. Additionally, extracted named entities (e.g., persons) serve as attributes of XML tags. We apply the framework in a case study on the German Commercial Register.*

1. Introduction

Tan points out that up to 80% of a company's information is stored in unstructured textual documents [18]. Undoubtedly, they are a major source of organizational knowledge. Effective knowledge management thus requires techniques to extract actionable knowledge from text archives. Feldman and Dagan coined the phrase "knowledge discovery in textual databases" (KDT) that refers to the extraction of useful knowledge from unstructured text documents [6]. In this paper, we introduce the KDT approach pursued in the research project DIAsDEM for knowledge management over application-specific documents. Our goal is the semantic tagging of textual content with meta-data to facilitate searching, querying and integration with associated texts and relational data. Hence, we aim at deriving an XML DTD that serves as a quasi-schema for the document collection and enables database-like queries on textual data.

DIAsDEM focuses on texts with domain-specific vocabulary and syntax. These application-specific collections contain rather homogeneous texts (e.g., police reports) with several particularities: Firstly, important and discriminating information is contained in fine-grained structural text components, although all texts deal with a limited set of subjects, e.g. the phases of crime investigation. Secondly, all texts adhere to a particular vocabulary as well as to a peculiar syntax and to linguistic conventions that may be far away from everyday language rules. Thirdly, these texts frequently share an inherent, though undocumented structure.

Our approach of converting texts into XML documents is based on clustering their structural components by semantics and making these semantics explicit as cluster labels. To this end, we propose an iterative clustering process: We progressively group text units (e.g., sentences) by similarity and identify concepts that describe the members of each group. For each semantic group, a cluster label is derived and subsequently used as an XML tag constituting meta-data for the corresponding text units. In addition, we identify named entities referenced in text units, e.g. names of persons and companies. Extracted named entities subsequently serve as attribute values of XML tags.

The rest of this paper is organized as follows: The next section discusses related work. Section 3 gives an overview of the proposed framework for semantic tagging, whereas section 4 describes its iterative KDT process in detail. Section 5 concisely describes the process of tagging text documents. Section 6 presents a case study that illustrates the application of the proposed framework. We conclude and present directions for future research in section 7.

*The research project DIAsDEM is funded by the German Research Society, DFG grant no. SP 572/4-1.

2. Related work

The related research can be categorized into knowledge discovery in textual databases, research on semi-structured data and projects pursuing similar objectives.

Tan briefly summarizes the current state of text mining and its future challenges [18]. He introduces a two-phase framework for text mining: In the text refining phase, unstructured texts are first transformed into an intermediary form that is later used to deduce knowledge in the knowledge distillation phase. This general approach is adopted in our proposed DIASDEM framework as well. We perform fine-grained semantic analysis and integrate domain knowledge that are open research problems according to Tan.

Nahm and Mooney propose the combination of methods from KDD and information extraction to perform text mining tasks [13]. 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 [7]. The authors represent each document by a set of terms and construct a taxonomy of terms. The resulting dataset is input to KDD algorithms such as association rule discovery. Our framework adopts the idea of representing texts by terms and concepts. However, we aim at the semantic tagging of text units within the document according to a global DTD and not at the characterization of the entire document's content. Loh et al. suggest to extract concepts rather than individual words for subsequent use for KDD at the document level [9]. Similarly to our framework, the authors suggest to exploit existing vocabularies such as thesauri for concept extraction.

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 DTD. Among other implications, discovering a concept that is only peculiar to a single text unit is not sufficient for our purposes, although it may perfectly reflect its content. In order to derive a DTD, we need to discover groups of text units that share semantic concepts. Moreover, we concentrate on domain-specific texts, which significantly differ from average texts with respect to word frequency statistics. These collections can hardly be processed using standard text mining software because the integration of relevant domain knowledge is a prerequisite for successful knowledge discovery.

Semi-structured data is another topic of intensive research within the database community [3, 1]. A lot of effort has recently been put into methods inferring and representing structure in similar semi-structured documents [14, 19]. 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 [17]. 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 [10]. However, his approach relies on establishing a rule base that drives the conversion, while we employ a KDD methodology to reduce necessary human effort.

Bruder et al. introduce the search engine GETESS that supports query processing on texts by deriving and processing XML text abstracts [2]. These abstracts contain language-independent, content-weighted summaries of domain-specific texts. Instead of creating abstracts, we aim at tagging complete text documents. Decker et al. extract meta-data from Web documents using the ontology-based system ONTOBROKER [4]. Maedche and Staab introduce an architecture for semi-automatically learning ontologies from Web documents [11]. Embley et al. also apply ontologies to extract and to structure information contained in data-rich unstructured documents [5]. 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.

3. The DIAsDEM framework

Our framework pursues two objectives for an archive of text documents: All documents should be semantically tagged and an appropriate XML DTD should be derived for the archive. Rather than classifying entire documents or tagging single terms, the framework aims at annotating structural components of text documents that are referred to as text units. Table 1 illustrates this notion of semantic tagging: The semantics of text units (i.e. sentences) are made explicit by semantic XML tags containing further meta-data as (attribute, value)-pairs. Thus, the input to the DIAsDEM mining phase is the set of all text units and neither the set of documents nor the text units of a single document.

Text units are clustered by similarity of their content. The objectives of DIAsDEM are particularly challenging

(...) `<crime type="burglary" company="Miller's Jewelers Inc.">` A platinum diamond ring was stolen from Miller's Jewelers Inc. on Saturday in one of several thefts reported to police. `</crime>` `<arrest person="Bryan Ray Owens">` The suspect Bryan Ray Owens was immediately arrested. `</arrest>` `<value amountOfMoney="3300 USD">` The ring was valued at $3,300 `</object>` (...)

Table 1. Semantically tagged police report

for the clustering methodology, because only semantically homogeneous clusters can be assigned a reasonable semantic tag. Additionally, a cluster should not be too specific, because a semantic tag comprised of many concepts such as `<immediateArrestOfSuspect>` can hardly be memorized and exploited during query formulation. Moreover, the cluster cardinality should not be too low, since lots of small clusters also imply many highly specialized tags.

Semantic tagging in DIAsDEM is a two-phase process. We have designed a KDT process that constitutes the first phase in order to build clusters according to the aforementioned requirements, to tag documents in XML according to the results and to derive an XML DTD describing the archive. 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. Rather, in each iteration of the KDT process, we re-adjust cluster similarity parameters. This process is also "interactive", because a knowledge engineer is consulted during cluster selection that is performed at the end of each iteration. Phase 1 of the DIAsDEM framework 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. In this phase, 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.

4. The iterative KDT process

In this paper, we focus on the first phase of our framework whose iterative and interactive KDT process is depicted in Figure 1. Besides the text documents to be tagged, the following domain knowledge constitutes input to this knowledge discovery 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 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 preliminary schema.

Similarly to a conventional KDD process, our process starts with a preprocessing phase, in which a reduced feature space is established. All text units are mapped into vectors of this space. Additionally, named entities of interest are extracted from text units by a separate module. For instance, the surname and the forename of a named entity "Person" is recognized in the text. Discovered named

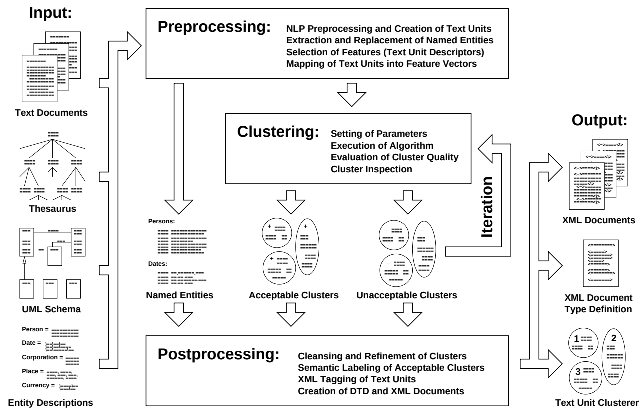

Figure 1. Iterative KDT process

entities and their values subsequently serve as attributes of XML tags. In the pattern discovery phase, text unit vectors 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 are partitioned into "acceptable" and "unacceptable" ones according to our quality criteria described in section 4.2. The postmining phase consists of a labeling step, in which "acceptable" clusters are semi-automatically assigned a label. Cluster labels are derived from feature space dimensions prevailing in the corresponding clusters. 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. This phase is described in section 5.

4.1. Establishing the feature space

In the preprocessing phase, we first determine the text units to be clustered. Since the objective of DIAsDEM is the annotation of documents with semantic tags, a document is not a single entity but a collection of structural components whose semantics should be identified and mapped into tags. Currently, the level of granularity in our analysis is a sentence. A more elaborate approach using noun groups or sliding n-grams as text units is planned as future work.

After performing word stemming, we establish the feature space. In conventional text mining, the feature space is determined by an application-specific controlled vocabulary, including specific terminology and excluding stopwords. In DIAsDEM, the feature space is much smaller, not only to reduce the number of dimensions, but mainly to reach the ultimate goal of DIAsDEM, namely the derivation of a DTD that describes a document collection. Accordingly, the feature space is comprised of terms and concepts that (i) are not rare and (ii) belong to the specific terminology used in the collection of text documents to be tagged.

Requirement (i) excludes rare terms. Among them are named entities (e.g., persons) which have high selective power and are very important in other text mining applications. Nonetheless, named entity identification is important for the applications addressed in DIAsDEM, so that identifiers are detected in the preprocessing phase and incorporated into tags as attributes during the tagging phase.

Requirement (ii) excludes all general purpose terms. This reflects the purpose of the DTD to be derived: It should describe the documents in much the same way a database schema describes records. For example, in a database of police records, a burglary would be rather the value of an attribute "crime type" than an attribute itself. Some terms that are excluded due to this requirement are still taken into account during the tagging phase as attribute values. Currently, this is restricted to a predetermined set of named entities. Ultimately, requirement (ii) can only be satisfied by feature selection performed by a domain expert. In the DIAsDEM framework, we propose that the application domain is conceptually modeled. Terms appearing in this schema as names of entities, associations, attributes or high-level methods form the basis of the restricted feature space.

All terms and concepts remaining in the feature space after applying requirements (i) and (ii) on the vocabulary are frequent or very frequent. In conventional text mining applications, very frequent terms are also excluded because of their low descriptive power. In DIAsDEM, we only exclude words at the rightmost part of the word frequency curve but still retain many words of high frequency. The reason is that some frequent terms appear in combinations that characterize text units. For example, "crime" may be a very frequent term in police records, but it is necessary to characterize something as an "element of crime". Similarly to IR conventions, we only keep a very limited number of word combinations in the feature space [15]. Interesting combinations of terms are discovered during the clustering phase instead.

The feature space is established by the end of this phase. The number of its dimensions is much lower than implied by the cardinality of the controlled vocabulary. Our notion of a "text unit descriptor" or simply a "descriptor" might refer to a single term, a broader term that stands for other narrower terms or a concept reflected by various different terms. In our framework, each text unit descriptor is a dimension of the feature space. Each text unit is then mapped into a boolean text unit vector over this feature space. In particular, we assign an order upon the feature space D, so that each descriptor $d \in D$ obtains an ordinal number $i(d) \in \{1, \ldots, |D|\}$. The set of text units in all documents $\mathcal{T}$ is thereafter mapped into the boolean vector space $[0,1]^{|D|}$ by a function m, so that for each $t \in \mathcal{T}$, $m(t)$ is a boolean vector v with $v[i(d)] = 1$ iff d appears in t and zero otherwise.

4.2. Iterative clustering of text unit vectors

Clustering of text unit vectors into groups of very similar content is the core of our proposed KDT process. This content is reflected by the set of text unit descriptors (i.e. feature space dimensions) that characterize each cluster. These prevailing descriptors are used in the next phase to derive cluster labels. Finally, these cluster labels are utilized to annotate the members of each cluster with XML tags.

The KDT process is based on a plug-in concept that allows the execution of different clustering algorithms within the DIAsDEM workbench. To group text unit vectors by similarity, we currently employ the so-called demographic clustering algorithm available in the IBM DB2 Intelligent Miner for Data [8] that maximizes the Condorcet criterion [12]. This criterion can be perceived as the difference between intra-cluster similarity and inter-cluster similarity. In particular, its value is the difference between the sum of all pair-similarities within the same cluster and the sum of all pair-similarities between vectors in and outside a cluster. This algorithm obtains as input an upper limit of clusters to be built and a similarity threshold value for assigning two vectors to the same cluster. We refer to the latter as the "intra-cluster similarity threshold".

We invoke the clustering algorithm iteratively. By the end of each iteration, a set of clusters is returned. These resulting clusters are evaluated against a set of DIAsDEM-specific cluster-quality criteria described below. If a cluster is found to be acceptable with respect to these criteria, a label is derived for it as described in section 5. Members of acceptable clusters are removed from the dataset, while the remaining text unit vectors are input to the clustering algorithm again. In each iteration, the intra-cluster similarity threshold value is stepwise decreased, so that acceptable clusters become progressively less specific in content. The iterative clustering approach reflects the objectives of XML tagging in DIAsDEM: It is desirable to derive a semantic tag reflecting the content of several text units as precisely as possible. If no precise content description can be found for a group of text units, a coarser one should be considered as well. If the number of text units sharing the same content description is too low, the intra-cluster similarity threshold should also be decreased. This again leads to coarser content descriptions.

In DIAsDEM, the quality of a cluster is high, i.e. the cluster is acceptable, if and only if it is (i) *homogeneous*, (ii) *large* and (iii) has its content described by a *small* number of text unit descriptors. The criterion of homogeneity is dealt with by the similarity-based clustering algorithm. As already noted, the homogeneity is progressively relaxed to allow for the maximization of the other two criteria.

The second criterion states that the cardinality of a cluster should be larger than a threshold lim_{size} provided by the knowledge engineer. The threshold lim_{size} should reflect the number of text units considered adequate to assign a tag to them, bearing in mind that this tag will be an element of the DTD to be derived.

The third criterion concerns the *cluster description*. A cluster c is described by its feature space dimensions, i.e. the text unit descriptors which appear among most of its members. Let $d \in D$ be a descriptor in the feature space D, let $i(d) \in \{1, \ldots, |D|\}$ be its ordinal number and let M_c be the collection of text units whose vectors are assigned to cluster c. Then, the normalized frequency of descriptor d in cluster c is the ratio of vectors containing d to all vectors in the cluster:

$$freq(d, c) = \frac{|\{\{t \in M_c | m(t)[i(d)] = 1\}\}|}{|M_c|}$$

where $\{\{\cdot\}\}$ denotes a multiset instead of a set, taking the fact into account that the collection of text units may contain duplicates. We denote the subcollection of M_c in the numerator as $M_c(d)$. Using this notion of frequency, the third criterion is decomposed into two constraints as follows:

- The ratio of the number of distinct descriptors in a cluster c to the total number of dimensions in the feature space D should be less than $lim_{dimensions}$.

$$\frac{|\{d \in D | M_c(d) \neq \emptyset\}|}{|D|} \leq lim_{dimensions}$$

- The frequencies of the descriptors within a cluster c are grouped into the intervals $HIGH$ $(0.8, 1.0]$, $MEDIUM$ $(0.6, 0.8]$ and LOW $(0, 0.6]$. The ratio of the number of distinct descriptors in the interval $HIGH$ to the total number of distinct descriptors in the cluster should be close to 1.

$$\frac{|\{d \in D | freq(d, c) \in HIGH\}|}{|\{d \in D | M_c(d) \neq \emptyset\}|} \geq 1 - \varepsilon$$

The first constraint excludes clusters characterized by a large number of descriptors, because such clusters are difficult to label in a precise way. The second constraint excludes clusters characterized by descriptors of modest frequency, because the homogeneity of these clusters is rather low. Hence, the third criterion of our quality scheme endeavors to find clusters, in which only a few descriptors appear in most members.

The reader might object that our quality criteria consider only descriptors within one cluster, without comparing them to the descriptors of other clusters. Indeed, a frequent descriptor in cluster c_1 might also be frequent in cluster c_2. This is partially alleviated by the clustering algorithm which maximizes cluster homogeneity and minimizes the similarity among clusters. We should not omit a descriptor that is frequent in more than one cluster altogether, since it may be part of *different* cluster labels. However, the feature space contains only a small number of term combinations, allowing for the free combinations of descriptors to formulate appropriate cluster descriptions and cluster labels.

5. Tagging of documents

The iterative KDT process outputs a set of clusters that satisfy our quality criteria. Each cluster is annotated with statistics calculated by the mining software and with a cluster description comprised of descriptors reflecting the cluster's content. Cluster descriptions and names of named entities are used to create tags for the semantic annotation of text units. XML tags are ultimately determined by the knowledge engineer. Nevertheless, DIAsDEM performs both a pre-selection and a ranking of candidate cluster labels for the expert to choose from. A cluster description consists of the feature space dimensions prevalent in the cluster, accompanied by their statistics. In order to label a cluster c, only descriptors $d \in D$ such that $freq(d, c) \in HIGH$ need to be considered. We distinguish between:

- Group-I descriptors that are considered frequent by the DIAsDEM workbench
- Group-II descriptors that are all other descriptors appearing in text units of a cluster and considered significant by the mining software

The descriptors in both groups determine the content of the corresponding cluster. With respect to the frequency intervals $HIGH$, $MEDIUM$ and LOW, a Group-II descriptor needs not belong to the $HIGH$ interval. For cluster labeling, the knowledge engineer is called to choose upon the Group-I descriptors, ordered by decreasing frequency. Group-II descriptors are also presented, because the expert may decide to concatenate the selected Group-I descriptor(s) with a member of Group-II. The visualization module of DIAsDEM aids in this procedure by presenting the text units assigned to the cluster. Thus, the knowledge engineer may cross-check the cluster's content against the descriptors and select a combination that is consistent with the linguistic style in the application domain.

For example, a cluster of text units from police reports might have the descriptor "location" categorized as a very frequent Group-I descriptor, while the descriptor "crime" is a member of Group-II. The knowledge engineer may check the cluster content and decide that "locationOfCrime" can be an appropriate label for this cluster.

During the preprocessing phase, named entities of interest are extracted by a special DIAsDEM workbench module. In the XML tagging phase, cluster labels are combined with (name, value)-pairs of named entities appearing in the text units to construct attributes of XML tags. In particular, all documents in the collection are tagged as follows:

1. Each document is decomposed into its text units.
2. Named entities appearing in each text unit are extracted and each named entity value is associated with its named entity name.
3. All text units are mapped into the feature space of text unit descriptors to create text unit vectors.
4. Iterative assignment of text unit vectors to clusters: In the i^{th} iteration:
   (a) Consider only clusters built in the i^{th} iteration of the original clustering process
   (b) For each text unit: If the corresponding vector can be assigned to one of the clusters under consideration, then tag the text unit with its label.
   (c) Remove all tagged text units from the dataset.
   (d) The remaining dataset is input to the next iteration.
5. The (name, value)-pairs of extracted named entities appearing in a tagged text unit are incorporated into the tag surrounding it.

DIAsDEM generates a set of tags for an archive that constitute a flat, unstructured XML DTD. This DTD reflects the content of documents and can thus serve as a preliminary database-like schema. If this quasi-schema is implemented in an DBMS, cluster labels correspond to table names, while names of named entities are attributes belonging to the tables. Text units and values of named entities constitute an instance of this schema. Discovering ordered or creating nested XML tags is part of our future work.

6. Case study

DIAsDEM is a general purpose framework. Its workbench can be coupled with application-specific thesauri, specific rule templates for named entity extraction and various clustering algorithms. In our case study, we applied the framework to a collection of German Commercial Register text documents. In Germany, each district court maintains a Commercial Register that contains important information about companies in the court's district. According to German law, company activities like the establishment of branch offices, changes in share capital, mergers and acquisitions must be reported. Knowledge of Commercial Register entries is indispensable for business transactions.

The availability of Commercial Register entries on the Web has a large potential for focused information acquisition. Indeed, due to the intense business demand for this commercial information, there are several information brokers offering both online and offline services to retrieve relevant knowledge from Commercial Registers. However, current services only encompass SQL queries to access relational data and full-text queries to search unstructured texts that contain most of the information.

HRB 12576 06.05.1999	Daniel Spiel-Center GmbH (Potsdamer Straße 94, 14513 Teltow).	publiziert am 19.05.1999
Der Betrieb von Spielhallen in Teltow und das Aufstellen von Geldspiel- und Unterhaltungsautomaten. Stammkapital: 25.000 EUR. Gesellschaft mit beschränkter Haftung. Der Gesellschaftsvertrag ist am 12. November 1998 abgeschlossen und am 19. April 1999 abgeändert. Ist nur ein Geschäftsführer bestellt, so vertritt er die Gesellschaft einzeln. Sind mehrere Geschäftsführer bestellt, so wird die Gesellschaft durch zwei Geschäftsführer oder durch einen Geschäftsführer in Gemeinschaft mit einem Prokuristen vertreten. Einzelvertretungsbefugnis kann erteilt werden. Pawel Balski, 14.04.1965, Berlin, ist zum Geschäftsführer bestellt. Er vertritt die Gesellschaft stets einzeln und ist befugt, Rechtsgeschäfte mit sich selbst oder mit sich als Vertreter Dritter abzuschließen. Nicht eingetragen: Die Bekanntmachungen der Gesellschaft erfolgen im Bundesanzeiger.		

Table 2. German Commercial Register entry

Table 2 contains an exemplary German Commercial Register entry. Each entry consists of a structured part and an unstructured text. The former contains the company's registered name, its record number as an identifier, the business address and relevant dates of registration and publication. This information can easily be extracted using wrapper technologies. The unstructured section of each entry contains the registered text as recorded by the court's clerks. In this case study, we have used 1,145 documents published by the district court of Potsdam. These documents are foundation entries of new companies in 1999.

We have established a preliminary conceptual model that partly reflects the application domain. Its UML class diagrams serve as a reference, against which the derived DTD can be matched. This conceptual model also formed the basis for specifying a controlled vocabulary of the domain. We have used word frequency statistics and the DIAsDEM thesaurus editor to build a hierarchy of 70 descriptors and 109 non-descriptors pointing to valid descriptors. The final feature space consists of 85 descriptors, after adding some terms known to be of importance in this domain.

We have partitioned the documents into text units, whereby the level of granularity was set to a sentence. Afterwards, the multilingual part–of–speech tagger TreeTagger was applied to determine lemma forms of all words [16]. The number of unique word forms was reduced from 10,613 to approx. 5,400. Our Java-based named entity extractor was employed to identify instances of named entities such as "Person", "Company", "Date" and "AmountOfMoney".

In Table 3, we summarize the size of the dataset and parameter settings in each of three clustering iterations performed. The KDT process was stopped by the knowledge engineer after three iterations. Altogether, 73 acceptable clusters were identified. They represent approx. 85% of all text units in the collection. This high proportion of tagged sentences can be explained with the fact that Commercial Register entries are composed of rather regular German lan-

Clustering iteration of KDT process	1	2	3
Number of input text units	10,785	1,818	1,648
Intra-cluster similarity threshold	0.95	0.90	0.80
Maximum number of clusters	200	200	200
Visualization threshold (cluster size)	10	5	3
Number of output clusters	122	121	67
Global Condorcet value	0.8090	0.9147	0.8176
Number of acceptable clusters	42	12	19
Text units in acceptable clusters	8,969	168	74

Table 3. Summary of iterative clustering

guage. In the future, the DIAsDEM framework will be evaluated against other archives such as company profiles and ad hoc news of publicly quoted companies.

In the last phase, labels of acceptable clusters were used to annotate sentences. In Table 4, the document of Table 2 is partly depicted after semantic tagging. The first sentence of this XML document is tagged as one referring to the business purpose of the new company. The second sentence refers to its share capital and contains a named entity, i.e. the amount of money invested in the company. Accordingly, the tag is extended to accommodate the named entity name “AmountOfMoney” and its value. The 5th tag refers to the manager appointed for the company: The named entity “Person” thus annotates this tag. Its value reflects the way persons are identified in many entries, i.e. by specifying surname, forename, current domicile and date of birth.

```
<?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
NE="AmoutOfMoney=[25000 EUR]"> Stammkapital: 25.000 EUR.
</ShareCapital> <LimitedLiabilityCompany> Gesellschaft
mit beschränkter Haftung. </LimitedLiabilityCompany>
<ConclusionArticles NE="Date=[12.11.1998], Date=
[19.04.1999]"> Der Gesellschaftsvertrag ist am 12. November 1998
abgeschlossen und am 19. April 1999 abgeändert. </Conclusion
Articles> (...) Einzelvertretungsbefugnis kann erteilt werden.
<AppointmentManagingDirector NE="Person=[Balski;
Pawel; Berlin; 14.04.1965]"> Pawel Balski, 14.04.1965, Berlin,
ist zum Geschäftsführer bestellt. </AppointmentManaging
Director> (...) <PublicationMedia> Nicht eingetragen: Die
Bekanntmachungen der Gesellschaft erfolgen im Bundesanzeiger.
</PublicationMedia> </CommercialRegisterEntry>
```

Table 4. Semantically tagged XML document

Table 5 contains an excerpt of the flat, unstructured XML DTD that was automatically derived from all discovered XML tags. It coarsely describes the semantic structure of the resulting XML collection. Currently, named entities are not fully evaluated. Named entities are denoted by the attribute “NE” in the DTD, without taking the exact named entity name into account. As part of our future work, attributes will be assigned a semantic name as well.

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

<!ELEMENT CommercialRegisterEntry (#PCDATA |
BusinessPurpose | ShareCapital | FullyLiablePartner
| AppointmentManagingDirector | GeneralPartnership |
InitialShareholders | (...) | FoundationPartnership)* >

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

<!ATTLIST BusinessPurpose NE CDATA #IMPLIED> (...)
<!ATTLIST FoundationPartnership NE CDATA #IMPLIED>
```

Table 5. Excerpt of the derived flat XML DTD

In contrast to text classification, there are no preclassified documents in our application domain, upon which the effectiveness of the DIAsDEM workbench could be measured. Instead, we have drawn a random sample containing approx. 5% of the 1,145 text units and asked a domain expert to detect tagging errors. We distinguish between two types of tagging errors, namely false positives and false negatives. A false positive occurs if the tag associated with a text unit does not entirely reflect its content. A false negative occurs when an un-tagged text unit conforms to a semantic concept that is part of the derived DTD.

Within the 5% sample, the false positive (false negative) error rate is 0.375% (3.565%).The percentage of false positives is very low. If a text unit is tagged, the tag is most likely to be correct. The percentage of false negatives is higher, indicating that some text units were not placed in the cluster they semantically belonged to. Our preliminary explanation for the comparatively high rate of false negatives is that these text units were characterized by terms that were not included in the feature space. The reader may recall that there was no thesaurus available for this case study, so that one had to be built from word statistics. A thesaurus contains several concepts, each of them expressed by many alternative terms. If some of these alternatives are less frequent than others, they may be ignored when building the thesaurus and deriving a feature space from it. Text units containing these infrequent words are thus mapped into vectors of poor quality.

The overall error rate in the 5% sample is 3.940%. With 0.95 confidence, the error rate in the entire dataset is in the interval [2.591%, 5.948%] which is a very promising result.

7. Conclusion

Collections of unstructured text documents contain information of great potential value. However, they can only be retrieved with full-text search in most cases. In this paper, we have presented a framework for semantic tagging and derivation of XML DTDs from domain-specific text archives. Our Java-based DIAsDEM workbench operates in two phases. An interactive and iterative KDT process groups all text units of all documents in the archive into a set of large, homogeneous clusters by their semantics, semi-automatically derives cluster labels that serve as XML tags and finally annotates text units with these tags, extended with information about named entities referenced in them.

We have tested our framework on a document collection from the German Commercial Register and shown that our approach is very successful, showing a very low error rate. This application area is particularly important for e-business, because Commercial Register entries contain indispensable information for business interactions among companies. While existing information brokers process these documents with conventional information retrieval techniques, DIAsDEM enables a more focused search through appropriate XML query languages that exploit XML tags and their associated attribute values.

Our future work includes the derivation of structured XML DTDs in contrast to the currently derived, rather unstructured and preliminary ones. We also intend to combine natural language processing techniques and n-gram clustering instead of sentence clustering. Additionally, further clustering algorithms and similarity metrics should be evaluated with respect to the objectives of our framework. Finally, we intend to reduce the human effort by (i) exploiting association rules during thesaurus construction and (ii) by extending the ranking mechanism that proposes cluster labels to the expert into a recommendation system that takes the preliminary schema into account.

8. 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] I. Bruder, A. Düsterhöft, M. Becker, J. Bedersdorfer, and G. Neumann. GETESS: Constructing a linguistic search index for an Internet search engine. In *Proc. of the 5th Int'l Conf. on Applications of Natural Language to Information Systems*, pages 227–238, Versailles, France, June 2000.

[3] 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.

[4] S. Decker, M. Erdmann, D. Fensel, and R. Studer. ONTOBROKER: Ontology based access to distributed and semi-structured information. In R. Meersman, editor, *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher, Boston, 1999.

[5] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Proc. of the 1998 ACM 7th Int'l Conf. on Information and Knowledge Management*, pages 52–59, Bethesda, MD, USA, November 1998.

[6] R. Feldman and I. Dagan. Knowledge discovery in textual databases (KDT). In *Proc. of the First Int'l Conf. on Knowledge Discovery and Data Mining*, pages 112–117, Montreal, Canada, August 1995.

[7] R. Feldman, M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir. Text mining at the term level. In *Proc. of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 65–73, Nantes, France, September 1998.

[8] IBM DB2 Intelligent Miner for Data. http://www.ibm.com/software/data/iminer.

[9] 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.

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

[11] A. Maedche and S. Staab. Learning ontologies for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.

[12] P. Michaud. Clustering techniques. *Future Generation Computer Systems*, 13(2–3):135–147, November 1997.

[13] U. Y. Nahm and R. J. Mooney. Using information extraction to aid the discovery of prediction rules from text. In *Proc. of the KDD-2000 Workshop on Text Mining*, pages 51–58, Boston, MA, USA, August 2000.

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

[15] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[16] H. Schmid. Probabilistic part–of–speech tagging using decision trees. In *Proc. of Int'l Conf. on New Methods in Language Processing*, pages 44–49, Manchester, UK, September 1994.

[17] A. Sengupta and S. Purao. Transitioning existing content: Inferring organization-spezific document structures. In *Tagungsband der 1. Deutschen Tagung XML 2000*, pages 130–135, Heidelberg, Germany, May 2000.

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

[19] 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.