BRIDGING THE GAP BETWEEN HETEROGENEOUS AND SEMANTICALLY DIVERSE CONTENT OF DIFFERENT DISCIPLINES

Siarhei Bykau, Nadzeya Kiyavitskaya, Chrisa Tsinaraki, Yannis Velegrakis

University of Trento
{bykau,nadzeya,chrisa,velgias}@disi.unitn.eu

ABSTRACT

The Web has been flooded with highly heterogeneous data sources that freely offer their data to the public. Careful design and compliance to standards is a way to cope with the heterogeneity. However, any agreement and compliance is practically hard to achieve across different communities. In this work we describe a framework that enables the exploitation of content across different scientific disciplines. Our approach combines several novel techniques at the syntactic, structural and semantic level. In particular, we advocate that integration should take place at the much higher level, factoring out any syntactic discrepancies, and facilitating the exchange of information. We show how a novel technique for data annotation using intentional attributes can cope with data associations in high data volumes, we present a way to overcome the multilingualism barrier, and describe a new kind of database that considers data evolution as first class citizen with the additional ability to annotate free text.

1. INTRODUCTION

Recent advances in information and telecommunication technologies have led a large majority of data owners to make their data available online. To fully exploit the potential of this information, modern information systems and individuals alike need to be able to successfully locate, access, and consume the information related to a task at hand from many different sources. To achieve this goal, interoperation is necessary. Unfortunately, the majority of these sources have been independently developed and for different goals, thus interoperability is a challenging task. A long line of research has already studied the different aspects of the problem for more than two decades [1]. The developed solutions are either compliance to standards or some form of tight integration.

In a globalized world true progress can be achieved only through successful knowledge dissemination and cross discipline fertilization. Unfortunately, both compliance to standards and tight integration across different disciplines are unrealistic. Traditional data management techniques are becoming day after day limited to cope with the problem of cross discipline information exchange. This is mainly due to the fact that people in different disciplines see the world from different perspectives, which results to different ways of modeling reality. They speak different languages, use different terminologies, consider different relationships among the data, and many others. Thus, the need for novel techniques for discovering and integrating information from cross-discipline sources is becoming apparent.

One of the first topics that need to be revisited are the principles on which our data models and query languages are based. Of the most prevalent types of heterogeneity is structural heterogeneity, i.e., the use of different structures to represent the same characteristics of a real world entity. A typical way to handle this heterogeneity is through mappings. The mappings are queries, or transformation scripts that translate data from one format to another. The format is described by the schema of the sources or the applications that need to exchange data. The mapping specification is typically performed by expert users that have a good knowledge of the structure and the semantics of the schemas of the different sources. This is a manual, time consuming and error prone process. To assist the data architect in specifying this kind of mappings, a number of tools, referred to as *schema mapping tools* [2], have been developed. Unfortunately, these tools suffer from two main limitations. First, they assume the existence of a schema, an assumption that may make sense in a large number of application scenarios but is not realistic in many others that involve highly heterogeneous content, where the schema can simply not be described. The second limitation, is that the transformation languages are designed for managing data structures, i.e., tables, tuples, attributes. This is ok for technical people. The advent of Web 2.0 with the social media and new technologies like mashups and Yahoo pipes, have brought the data integration task to the regular Internet users, that are thinking not in terms of formal data models, but in terms of real world entities. This means that models and languages need to be adjusted and become not data structure transformation tools but real world entity transformation tools.

A second issue that needs to be revisited is the data linking mechanisms. To effectively communicate the data semantics, data curators are typically annotating the data with meta-information. Existing annotation creation and management techniques are implemented on top of the standard attribute or reference mechanisms offered by the various data models. A

limitation of the attribute modeling as currently implemented in ontologies or other data modeling formalisms is its static nature. More specifically, the existence of an attribute between two concepts or individuals depends solely on whether it has been explicitly defined or not. This prevents the implementation of batch assignment of attributes to groups of concepts/individuals that are currently present in the knowledge base or that may appear in the future. For instance, in many practical scenarios, attributes may need to be assigned to individuals based on some common characteristics. Currently, this task requires first to find the individuals that have these characteristics, iterate over them, and explicitly assign to each one the attribute of interest. Furthermore, if one or more individuals satisfying these characteristics are introduced at some future point in time, they will not be automatically assigned the attribute, unless a special ad-hoc mechanism has been put in place, or the ontology administrator manually assigns it to each such individual.

Another main obstacle of information dissemination and system interoperability has been the language barrier. By language we do not mean only the use of different official languages but the general practice of using the same words to represent different concepts in different contexts, or the use of different words to represent the same concept under different contexts. The context may include a whole range of parameters such as, the actual language, the location, the time, etc. In the core of the majority of the existing data querying techniques is the string comparison operator. This means that if for the same concept different words have been used in the data without the respective translation, all these techniques will fail. Unfortunately, text and data, translation has been proved to be a complicated task [3]. Translation of queries and data values expressed in one language/context into others in an efficient way has yet to be seen.

One more issue that has not received considerable attention is that of semantic evolution. The fact that data is evolving continuously has been known and studied for quite some time now [4] but this evolution concerns the structural evolution, i.e., the evolution of the values. However, as time passes and real world entities are evolving, i.e., aging, they can either continue to be represented in the sources by the same data structures, or new structures are introduced to model the new evolved real world entities. Scaling this to the size of the pluralism in the modern web results into a situation in which sources developed at different points in time naturally contain terminology and modeling structures that differ, even if they model the same real world event, entity, or concept. This semantic and conceptual evolution has not been taken into consideration by the modern information systems, resulting into the loss of valuable information during query answering.

Traditionally, highly structured repositories have been used as the main means of information storage. This was mainly because of the nature of the business data. The modern web transformed the regular Internet user from a passive data consumer of the web information into an active data producer and provider. Blog posts, social networking sites, twitter messages, and many others are the new web applications that day by day load the web with additional data. To fully exploit this treasure of information that is daily becoming available, we need to understand the semantics of the produced text. Unfortunately, text likes a predefined schema or a well defined format, and as such any semantic meaning has to come from a careful analysis of the textual context.

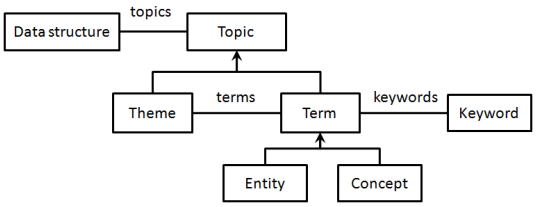

Fig. 1. Data source modeling

In this work we describe a number of solutions we have materialized into the TRENDS system in order to cope with the above challenges posed by the interdisciplinary search requirement. In particular, we describe a new high level and generic modeling technique (Section 2) that allows to factor out any structural discrepancies, a novel data linkage mechanism (Section 3), a method for dealing with semantic evolution (Section 5), a framework for passing the language barrier (Section 4), and a technique to annotate text and provide its semantics (Section 6).

2. MODELING DIFFERENT WORLDS

To allow for a user-friendly modeling of the domain knowledge, we need a mechanism enabling the representation of generic conceptual facts about the data that needs to be queried, factoring out any structural or syntactic discrepancies that may exist in the data. To this end, we advocate a domain-independent schema illustrated in Figure 1.

According to this schema, an information source, called *data structure*, can be assigned a set of *topics*. A topic can be a general classification label, called *theme*, as for instance, "Social anthropology" or "Demography", or a *term* that represents a physical object or an abstract notion described by the data structure. Among terms we may distinguish *entities*, such as prominent people, organizations or relevant locations, e.g., "Adam Smith", "Catholic Church", "People's Republic of China", or other terms called *concepts*, e.g., "social phenomena", "globalization", "ethical concern".

The advantage of our modeling is that it can be applied to any target domain by instantiating it with domain-specific knowledge. This instantiation can be supported at many levels by several automated methods. Themes, being general topics that describe data structures, normally have to be defined manually depending on a task at hand. For instance, a school administrator may need to classify a collection of educational materials by the subject matters they cover: eco-

nomics, ecology, religion, and others; whereas a local social studies department can be interested in classifying these materials according to their ideological viewpoints: capitalistic, environmentalistic, humanistic, and similar. Those subject matters compose domain themes that describe the data structures, i.e., educational materials, at the coarse level of detail.

However, themes may not always suffice to allow establishing useful mappings between different domains. Therefore, in each theme we may need to identify finer-grained topics, i.e., terms. For populating the suggested schema with entities, we utilize the Stanford Named Entity Recognizer [5]. Alternatively, one may adopt an existing gazetteer of named entities. For instance, the GeoNames database provides a list of location names[1]. Finally, for identifying concepts from a collection of textual documents in the target domain, one can take advantage of the machine learning methods for concept mining, or the knowledge base manually built by a user community, such as Wikipedia, WordNet or any other available term glossary. In our applications, we have been using the method that combines a linguistic processor with the Wikipedia database [6]. Once the domain schemas are instantiated, the mappings between the different domains can be defined, either by an automated tool or manually.

To deal with the high heterogeneity, we have designed a novel flexible query language. The language assumes a data model that is based on entities. The previously defined semantic modeling is actually implemented on top of this low-level entity based model that is close to the one used in every dataspace application [7]. The syntax of the language is exactly the same as the one of datalog. However, the semantics are different. In particular, while in datalog a term means an iteration over a relation set, in our case the terms mean template matching. For instance, the expression P(name:John, age:22) in datalog means the discovery of those P tuples that have attribute name and age with values John and 22, respectively. In our language, it would mean to search within the whole database to find the entities that have these attributes and values, and have also an identified P. The fundamental difference here is that it is possible that these entities have additional attributes, something that was not possible in datalog, and also that no schema information is required.

We have used the same language to express the mappings. Recall from the mapping literature, that a mapping is a pair of queries of the form $Q1 \rightarrow Q_2$ that specify how data from one world are expressed in terms of another world. In contrast to the mappings we have used in some previous work of ours [2], these mappings offer additional flexibility and transformations that could not have been expressed with these languages [8].

3. INTENSIONAL ATTRIBUTES

To address the problems of automatic data annotation, batch attribute assignments for the current and future data we introduced the notion of *intensional attributes*, i.e., attributes whose domain and range have been intensionally defined instead of explicitly stated. We have successfully used the idea before in the relational [9] and the RDF [10] world. In this work we have applied it for models based on the Dataspace notion, which was much more challenging due to its heterogeneity.

Individuals are assigned to the intensional attributes' domain and ranges in a similar fashion to which they are assigned to the extensions of defined concepts in Description Logics (DL) TBoxes (as opposed to the explicit way individuals are assigned to the primitive concepts). We employ queries in our extensive query language that we mentioned in the previous section in order to specify the domain and range of the intensional attributes. In particular, the intensional attributes have the following form: $\langle Q_d, name, Q_r \rangle$, where Q_d and Q_r are queries that specify, respectively, the domain and range of the intensional attribute $name$. Although we proposed to use SPARQL as a query language, it can be easily replaced with any other available one. Intensional attribute interpretation can be realized through the materialization of domain and range queries; We create normal attributes having the same name with the corresponding intensional attribute by the inter-connection of all the pairs of instances obtained after the query execution.

We have showed that our queries are excellent tools to implement intensional attributes since they provide the ideal means to refer to sets of data declaratively.

As an example of the applicability of the intensional attributes, lets assume that a user would like to add some superimposed information on the countries, indicating that every country with a population less than 20 millions will have to be financially audited. To add this kind of information on the countries, the user will have to explicitly add a special attribute with the appropriate text to each such country. Allowing the user to add attributes of this kind may not always be feasible or desirable. It may not be feasible if, for instance, the user does not have permission to edit the database. Even if this is not the case, it may not be desirable since adding attributes to the database concepts and individuals without some control mechanism may alter their semantics. On the contrary, using an intensional attribute between a string with the aforementioned statement and the query that returns all the countries with population less than 20 millions, the desired result can be achieved even without having permissions to modify the database values.

4. COPING WITH POLYSEMY

We have developed a context-based framework in order to support the interpretation of polysemous query terms. The idea is that the query terms should not be interpreted in isolation, but only relative to the context of the query they appear in. Every query term is a string value that is associated to some high level concept which is generic and independent of any language or any other factor. The term, however, is as-

[1]GeoNames project: http://www.geonames.org/

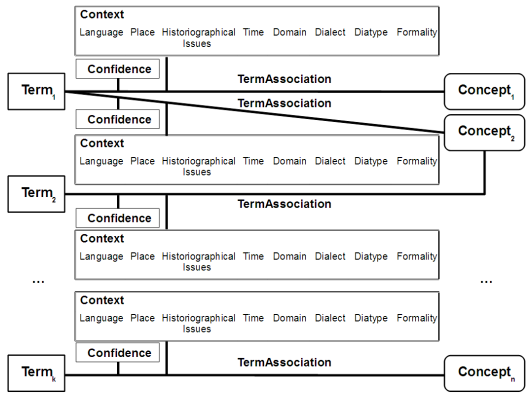

Fig. 2. Context-based association of Concepts and Keywords

sociated to the concept only under certain conditions that are determined through a set of parameters representing a context. This idea is graphically depicted in Fig. 2. One of the advantages of such a modeling is that it facilitates the representation of term evolution throughout the time, even if these terms are expressed in different languages.

Let t be a term, s be a concept and c a context, such that s is associated with t under the context c through an association $a(s, t, c, w)$, where w is a numeric value in the range $[0, 1]$ and describes the confidence of the association.

A context c, is a vector $c\langle d_1 : v_1, \ldots, d_k : v_k\rangle$, with each d_i,v_i pair being a *context dimension*.

For the interpretation of polysemous query terms, a context c comprises of the following $k = 8$ dimensions: **(i)** $d_1 = l$, which represents the language of c; **(ii)** $d_2 = p$, which represents the place of c; **(iii)** $d_3 = t$, which represents the time period(s) covered by c; **(iv)** $d_4 = d$, which represents the application domain of c; **(v)** $d_5 = h$, which represents the historiographical issues (i.e. social conditions, economical issues etc.) that should hold for c to be valid; **(vi)** $d_6 = dl$, which represents the dialect of c; **(vii)** $d_7 = dt$, which represents the diatype of c (i.e. a language variation, determined by its social purpose [11] like, for example, the specialized language of an academic journal); and **(viii)** $d_8 = f$, which represents the formality of c and may take the values "Very formal", "Formal", "Neutral", "Informal", "Very informal").

The users may specify (explicitly or implicitly) in their queries some context c and receive results related to the concepts associated with the query terms under c.

As an example, consider the term 'lorry', which describes, in the UK English, a specific type of vehicle. This vehicle type is described by the term 'truck' in US English; the same term, though, is used in the UK English in order to describe a part of a train, also described by the term 'wagon' in both the US and the UK English. Our technique allows to associate the term 'truck' with the concept 'wagon' through an association $a_1(`wagon', `truck', c_1(l : `English', dl : `US - English'), w_1)$, while associating the term 'truck' with the concept 'lorry' through an association $a_2(`lorry', `truck', c_2(l : `English', dl : `US - English'), w_2)$. Thus, if a user specifies the 'truck' term in a query posed under a context $c(l : `English', dl : `US - English')$, he will receive documents referring to the vehicle type, while he will receive documents referring to wagons if he specifies the 'truck' term in a query posed under a context $c'(l : `English', dl : `UK - English')$.

5. MANAGING EVOLUTION

To support semantic evolution we employ five special attributes that are used to associate different artifacts in the data repository. These associations specify some form of evolution relationship among these artifacts. They are: *split, merge, detach, evolve* and *join* [12]. More specifically, *split* models the fact that some entity appears at the time of *split* and inherits some parts of its ancestor. On the contrary, *merge* defines the fact that some entity becomes a part of another. The same works for *detach* and *join* with the difference that entities exchange only their parts without changing their life spans.

To support the aforementioned semantics we define two primitive attributes, *becomes* and *part-of*. The first one defines the causality between entities, and the second – the mereological relations between them. Using different combinations of the primitive attributes we formally describe the meaning of high-level relations. For example, *split* is decomposed into one *becomes* relation and one or more *part-of* relations which changed their owner from the ancestor to the descendant (not necessarily all parts). Furthermore, the data model we assume is one that supports temporal constraints, i.e., every artifact or association in the data repository has been assigned its validity interval. The validity intervals must conform to a set of constraints such as the life span of a property must be during the life span of the corresponding property, the life spans of literals are within the entire available time line and others. With such a modeling, it is possible to construct the so-called evolution graph, i.e., a graph that illustrates the evolution of one or more concepts or entities through a series of different design artifacts in the data repository.

We have also developed a graph-navigation query language in order to traverse such evolution paths. The language allows users to formulate queries about the history of entities in both terms of causality and entity constituents. For instance, we can ask about the ancestors of some entity or its direct descendants. Efficient query evaluation becomes a crucial aspect of the system, since very often the transitive closure may have to be computed. Special indexing structures and bloom filters are employed to improve the query execution time.

The inverse problem of discovering the evolutionary operators from the available data is considered as well. We propose to analyze the re-allocations of the entity parts in order to infer possible evolutionary connections.

As an illustrative example, consider, for instance, the concept of Biotechnology whose meaning has undergone several changes throughout history. The notions of Selective Breeding, Fermentation and Hybridization existed from the ancient

times until now. In the 40s, however, they were combined with the new topic of Conventional Biotechnology, which was later on transformed into the current term of Biotechnology. Using the evolution framework we can explicitly model the concept of Conventional Biotechnology as a direct descendant of Selective Breeding, Hybridization and Fermentation, which then *evolved* to the modern notion of Biotechnology (through several intermediate evolution transformations). As a result, we can find the 'ancestors' of Biotechnology and retrieve all the concepts related to the term throughout history.

6. EXPLOITING TEXTUAL KNOWLEDGE

To facilitate the storage, location and processing of information on the web, semantic annotations are used. However, given the diversity of formats and domains, the web scale, and the high cost of human supervision, this task is not trivial and must be largely supported by automated tools.

In order to relate textual data structures of different granularity (e.g., whole documents, paragraphs, or word collocations) to conceptual categories of the domain schema, populated as specified in Section 2, we developed a toolset for customized text analysis. Our approach combines two different techniques, a classification method built on Support Vector Machines (SVNs) [13] and a semantic annotation method based on the Cerno framework [14]. The relations between the schema concepts are used as constraints for the analysis process. Accordingly, a document can be assigned zero or more topics and it can be provided with at most one theme and multiple terms belonging to this theme; the number of the terms is not limited. The annotations are stored in an XML file, that is then indexed and can be fetched by a search engine. In this way, semantic annotations convey information about the document theme and terms of interest, facilitating the user's work in finding relevant data. Thus, the information search is no more keyword-based, but semantic.

More specifically, our annotation approach first uses SVN-based classification models, preliminarily trained on manually classified data corpora, for generating theme annotations. As a result, each document is assigned one main category as found in the ontology. To assign annotations of those themes for which we may not have many training examples available, or of sub-themes of the main themes in case a theme taxonomy is defined, a greater extent of human attention is needed. For this purpose, we use a method based on the semi-automatic semantic annotation framework Cerno. This method generates theme annotations using a set of hand-crafted annotation rules. At the second annotation phase, a similar rule-based method is used to identify instances of the domain terms with a difference that the annotation rules are constructed automatically from the populated domain schema, as in Figure 3. The domain schema is first parsed by the Grammar Generator in order to extract keywords related to term instances, i.e., entities or concepts, and compile the annotation rules to their formal representation in Backus-Naur Form (BNF)-like syntax. Finally, the generated rules are passed

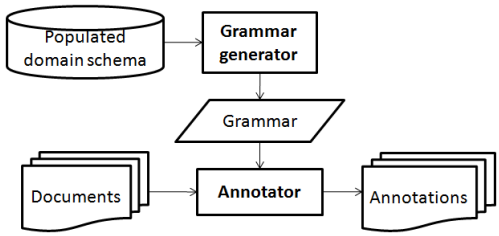

Fig. 3. Automatic annotation process

to the Annotator to produce term annotations. This annotation stage can result in identifying multiple entities or concept evaluation instances in a single document.

Once the schema-based annotations for the data structures of the target domains are generated, the mappings between their domain schemas are used to translate user queries in cross-domain information retrieval.

7. CONCLUSION

In this work we presented a number of solutions we have developed as an answer to the challenges faced when trying to achieve cross-discipline digital library interoperability. The core of each challenge reminisces those faced a decade ago by information integration systems, however, the novel reality required new, different and more advanced solutions.

References

[1] M. Lenzerini, "Data Integration: A Theoretical Perspective," in *PODS*, 2002, pp. 233–246.

[2] L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernandez, and R. Fagin, "Translating Web Data," in *VLDB*, Aug. 2002, pp. 598–609.

[3] M. Espinoza, A. Gómez-Pérez, and E. Mena, "Enriching an ontology with multilingual information," in *ESWC*, 2008, pp. 333–347.

[4] B. S. Lerner, "A Model for Compound Type Changes Encountered in Schema Evolution," *ACM TODS*, vol. 25, no. 1, pp. 83–127, Mar. 2000.

[5] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning, "Named entity recognition with character-level models," in *HLT-NAACL*, 2003.

[6] G. Paci, G. Pedrazzi, and R. Turra, "Wikipedia-based approach for linking ontology concepts to their realisations in text," in *LREC. To appear*, May 2010.

[7] X. Dong and A. Y. Halevy, "Indexing dataspaces," in *SIGMOD*, 2007.

[8] R. Fagin, L. M. Haas, M. Hernandez, R. J. Miller, L. Popa, and Y. Velegrakis, *Conceptual Modeling: Foundations and Applications by A. Borgida, V. Chaudhri, P. Giorgini, E. Yu*, chapter Clio: Schema Mapping Creation and Data Exchange, pp. 198–236, Springer, 2009.

[9] D. Srivastava and Y. Velegrakis, "Intensional Associations between Data and Metadata," in *SIGMOD*, 2007, pp. 401–412.

[10] A. Presa, Y. Velegrakis, F. Rizzolo, and S. Bykau, "Modeling associations through intensional attributes," in *ER*, 2009, pp. 315–330.

[11] Michael Gregory, "Aspects of varieties differentiation," *Journal of Linguistics*, vol. 3, pp. 177–197, 1967.

[12] F. Rizzolo, Y. Velegrakis, J. Mylopoulos, and S. Bykau, "Modeling concept evolution: a historical perspective," in *ER*, 2009, pp. 331–345.

[13] A. McCallum, *Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering*, 1998.

[14] N. Kiyavitskaya, N. Zeni, J. R. Cordy, L. Mich, and J. Mylopoulos, "Cerno: Light-weight tool support for semantic annotation of textual documents," *DKE*, vol. 68, pp. 1470–1492, 2009.