Integration of Data Warehouse and Unstructured Business Documents

Ahmad Abdullah Alqarni and Eric Pardede.
Department of Computer Science and Computer Engineering,
La Trobe University,
Melbourne, Australia.
Email: [aalqarni@students., E.Pardede@]latrobe.edu.au

Abstract—The profusion of unstructured data forced organizations to manage and take advantage of such data especially in the decision making process. The feasibility of integrating or mapping unstructured data to a data warehouse is becoming significant to bridge this gap and take the full potential of these data. In this paper, we propose a multi-layer schema for mapping structured data stored in a data warehouse and unstructured data in business-related documents. The multi-layer schema facilitates the mapping between the two different data. Linguistically correlated data is identified using WordNet to enable the integration between both data sources. We also propose a generic XML schema for business-related unstructured documents to assist the mapping. The use WordNet to identify the matching result is promising in the absence of schema-instance and without the need to domain specific knowledge.

Keywords-unstructured document; data warehouse; data integration; XML schema matching; schema mapping.

I. INTRODUCTION

Data plays an important rule for developing decisions in daily life situations including businesses. Consequently, data can lead decision makers to take major and momentous decisions. Most of such data is coming from structured repositories and eventually fed to a data warehouse. This is due to the fact that data warehouses have full capacities power in integrating, accessing, and analyzing data.

This situation was accepted in the past due to the limited resources, limited capacity of unstructured data, and lacking proper techniques to manage unstructured data. However, the proliferation of unstructured data has dramatically exceeds the amount of structured data. In fact, in the last years there has been a dramatic increase in the volume of unstructured data, due to the increased usage of internet and the huge number of individuals and organizations which create and store such data. On business organizations domain only, large portion of organization stored unstructured data in document format. It leads to the significance of such data particularly in the process of decision making. In this paper, we propose a multi-layer schema for mapping structured data stored in a data warehouse with unstructured data in business-related documents. The multi-layer schema task is to facilitate the mapping between the two different data by identifying correlated data and creating the connection between them using XML schema matching. This can be achieved by modeling a generic XML Schema for unstructured documents and match it with the logical level (XML Schema) of corporates data warehouses.

Our motivation is enable data warehouse users fetching more detailed information from unstructured data. The mapping will lead to discover important unstructured data that data warehouse incapable to provide, and guarantee that only valuable unstructured data is mapped.

This paper is organized as follows, background of the issue and related work are presented in Section 2. Section 3 presents our approach in details. Implementation and evaluation are summarized in Section 4 with the findings. Finally, conclusion and future work are presented in Section 5.

II. BACKGROUND

Unstructured documents such as emails, business documents, invoices, contracts, government agency reports, and customers’ feedback are usually not included in data warehouses due the nature of plain text format and the lack of schema definition. In addition, data stored in electronic formats such as office documents may also be considered as unstructured documents. Furthermore, some valuable document that non-compliance with an organization’s schemas or data represented in different standards to an organization will fall in the category of unstructured data. As a result to that, significant portion of data is excluded from the analysis and the process of decision making. According to a recent study in [1], it is estimated that around 80% of organizations data is located in text documents such as, webpages, user manual, technical reports. In the last few years, the amount of unstructured data has been increased dramatically.

A. Related Works

Several approaches in the vein of finding and integrating valuable data existed in plain text documents had been proposed in [2-8]. Clearly, the primary target for the approaches is the integration of unstructured data to structured database or repository. These approaches fall in the category of undirected integration, because unstructured data is firstly integrated to structured database before it can be fed to a data warehouse. However, this procedure is very cumbersome due to time and efforts that required achieving it. We will briefly discuss some of these approaches, their limitations, and its relation to this research paper.

Approaches in [3, 5, 7] rely on existing structured resources to find rules or clues that effectively enhance the...
extraction process by guiding the extraction system. Basically, the accuracy of information extraction (IE) tools is insufficient and consequently the yield of this extraction process will be error-prone. Hence, rules extracted from database using current data mining approaches in [3, 5] can guide and enhance as well the results of IE systems in the extraction process of finding valuable data in unstructured data. Although these approaches focus on structuring text into structure data, it was limited on mining interesting relationship. Approach in [7], takes advantage of structured, semi-structured, and domain ontologies data sources to improve IE systems results. These approaches suffer from serious limitations. First they are unsupervised, which make them prone to error. Furthermore, combining structured and unstructured data in a single repository may cause large flow of data for decision makers.

It is our aim to target only valuable strategic data in text documents and map it to the corporate data warehouse data. We aim to keep different data in separate repositories and facilitate only the mapping between them. This is important to diminish the flow of data and allow the key people to decide whether the corporate data is enough or further correlated unstructured data is needed.

Other approaches [9, 10] find substantial unstructured data first and then integrated with corporate data warehouses in a new type of data warehouse called Contextualized data warehouse (R-Cubes) without the need to go through traditional databases. The principal task of R-Cube is to maintain a record of the facts and their corresponding contexts as defined in documents. This can be achieved by focusing on unstructured XML documents format. Once the R-Cube is built, the analyst (user) needs to supply sequence of keywords. Then, R-Cube system performs the analysis by finding documents that contain facts related to the supplied context. These approaches have a number of limitations. For example, [9] focuses on metadata, while in [10], the user is assumed to be an information retrieval (IR) expert to be able to supply the context of analysis.

Our approach is different in the sense of performing the mapping prior to the analysis by utilizing data warehouse schema. Once the mapping is done, the user will be able to determine if further data from related unstructured documents is required. Also, user in our case is not required to provide context of the analysis.

Mapping unstructured data techniques in [11, 12] eliminate the needs for the physical integration of the different and heterogeneous resources. The mapping provides a unified way of viewing different types of data. Therefore, the mapping process is considered a substantial part in the process of finding and establishing the relationship between data in different repositories. Approach in [11] discusses the virtual integration between data warehouse and unstructured data existed in documents through an enterprise knowledge portal. The base of this integration is a global RDF schema ontology, which stores both resources metadata. This mechanism assists the mapping between both metadata structures towards facilitating the communication between the components of systems. However, one major disadvantage in this approach is the need to perform IR on unstructured document with every single analysis, which requires great effort and wastes limited organizations resources. Furthermore, this approach only limited to document management system that support RDF-based metadata storage.

Similarly, the main objective of [12] is to map data warehouse with geographical information system (GIS). The outcome of such mapping leads to a great improvement to the decision support quality. The most serious limitation of this approach is that it only focuses on the integration of GIS data with data warehouse. Our approach on the other hand, focuses on integrating business-related unstructured data to improve the decision making quality.

B. Modeling Data Warehouse

Before we propose the mapping of data warehouse with unstructured documents, we need to ensure that both have similar logical representation. We use XML Schema as the logical model. Modeling relational data warehouse with XML Schema is based on Bouassaid et al. works in [13]. In this paper a methodology called X-Warehousing has been proposed, which creates a logical level to data warehouse and then populate the physical level of this data warehouse with XML documents. Authors use of multidimensional conceptual level (MCM) to easily achieve the multidimensional structure representation of data warehouse.

Relational data warehouse model is generally a description of a fact table linked to sets of dimensions. Two obvious examples of this model are star and snowflake schemas. The fundamental difference between both schemas is that a star schema composed of single fact table and a set of independent dimension tables, while snowflake schema is identical with the addition of the hierarchical representation of each dimension. In the following sections we will present the modeling mechanism used in [13] to transform star and snowflake schema data warehouse to XML Schema model.

1) Model Star Schema with XML: Star schema is represented by a fact table F and set of dimensions D {D s, \(1 \leq s \leq m\)}. F table have m measures attributes \{F.M i, 1 \leq i \leq n\}. Whereas, each D s tables have a set of n s attributes \{D s.A j, 1 \leq j \leq n s\}. Thus, XML star schema that supposed to represents star schema (F,D) is considered XML star schema where:

- F represents the XML root element in the XML schema.
- \(\forall i \in \{1,\ldots,n\}\), F.M i, represents an XML attribute inside XML root element.
- \(\forall i \in \{1,\ldots,m\}\), D s, represents an XML element of XML root element as long it is connected to the fact table.
- \(\forall i \in \{1,\ldots,m\}\), and \(\forall j \in \{1,\ldots,n s\}\), D s.A j represents an XML attribute within XML element D s.

2) Model Snowflake Schema with XML: Since the snowflake dimensions is represented in a hierarchal way and XML formalism accept to include multi-level of sub-elements in single XML tag, authors use this feature to
represent dimensions hierarchy using XML. Hence, XML snowflake equivalent of conceptual snowflake schema can be represented as follows:

Let (F, HD) is a snowflake schema, where F represents a fact table with m measures attributes \{F.M_i, 1 \leq i \leq n\} and HD = \{HD_s, 1 \leq s \leq m\} that represents a set of m hierarchies. Thus, the XML snowflake schema of (F, HD) is considered as an XML snowflake schema where:

- F represents XML root element in the XML schema.
- \(\forall i \in \{1, \ldots, n\}\), F.M_i represents an XML attributes inside the XML root element.
- \(\forall s \in \{1, \ldots, m\}\), HD_s represents XML dimensions hierarchies as long as it is connected to the fact table. In other word it represents a sub-element to the XML root element.

C. Semantic Matching Overview

Schema matching is used to identify the corresponding elements of two schemas based on semantic relations [14]. Several approaches for semantic similarities have been proposed, such as linguistic-based metric, which use string matching to compute semantic similarity of two elements' names or descriptions. Examples of such measures are Jaccard and Cosine [14]. Other approaches proposed the use of linguistic taxonomy such as, WordNet lexical database in order to attain more accuracy and less semantic ambiguity [14]. We adopt WordNet to accomplish the task of schema matching in our proposal as named matcher mechanism.

WordNet was presented by Miller et. al [15]. It is an online dictionary that supports semantic and syntactical meaning and matching. Words relations such as synonyms, hyponyms, and antonym are used in the computation of semantic hierarchy level. These levels represented groups of words named synsets. In addition, nouns and verbs are organized in hierarchies of is-a relations. Other relations are also provided such as has-part. Furthermore, definition of each concept is represented by short gloss which may contain usage example. All such information can be used when creating measures of relevance.

WordNet::Similarity [16], is a free open source software package that facilitates measuring the semantic similarity and relatedness between pair concepts. It includes six similarity measures, and three measures of relatedness. The implementation of these measures is in a Perl module, which accepts two inputs (concepts) and returns a similarity numeric value, which determines the degree of similarity or relatedness. The compared concepts are processed by a steaming algorithm [17]. Due to time and space constraints only two WordNet measures Lin [18] and Wu and Palmer [19] will be discussed and used in this paper.

Lin’s, measures information context augmentation of the lowest common subsumer of the two concepts (LCS) with the sum of information content of both concepts \(c_1, c_2\). Whereas, Wu and Palmer measure (\text{wup}), relies on depths of the two synsets in WordNet with the depth of the LCS. Equation (1) and (2) respectively, show how both approaches scale as in [20, 21].

\[
\text{Sim}_\text{lin}(c_1, c_2) = \frac{2 \times IC(\text{LCS}(c_1, c_2))}{IC(c_1)+IC(c_2)} \tag{1}
\]

\[
\text{Sim}_\text{wup}(c_1, c_2) = \log \left( \frac{2 \times \text{depth}(\text{LCS}(c_1, c_2))}{\text{depth}(c_1)+\text{depth}(c_2)} \right) \tag{2}
\]

III. PROPOSED FRAMEWORK

In this section, we describe approach to facilitate the mapping between data warehouse and unstructured documents. We use a multi-layer data schema that consists of three layers namely data warehouse layer, mapping layer, and unstructured documents layer. The top layer is schema extracted from existing data warehouses. The bottom layer is schema extracted from unstructured documents. The middle layer is the generated schema that is derived from mapping the previous two layers. The core part of our integration methodology is to use XML Schema as a content representation for both top and bottom layer. Then by using XML schema-based matching mechanisms mapping parts can be identified. Fig. 1 depicts the overall structure of the proposed Multi-Layer Schema.

A. Generic XML Schema for Unstructured Documents

In this section, we will process the bottom layer of our multi-layer schema shown in Fig. 1. Our aim is to create a generic structure for variety documents in this layer. For this paper, we focused on several business-related document types such as, invoices, contracts and catalogues. By analyzing these documents we observe that some attributes are common in different type of documents. Furthermore, there are general patterns for each type of documents where many attribute appeared in almost all documents of this genre.

Based on the above observation, we can create a template for a generic schema that represents unstructured documents. This has a greater importance in our proposal, in order to automate the mapping of such data with a data warehouse with an unsupervised mechanism. This generic schema will represent a variety of different documents and this due to the unique structure of the schema.
Basically, the overall structure of the generic schema is composed of generic parts and several specific parts as shown below. The generic part will represent all common attributes, whereas specific parts will represent common attributes in each document type. For instance, the name and data of invoice documents may considered as common attributes that may also existed in other different type of documents. Whereas “invoice id” and “customer name” attributes, may considered specific to invoice type documents.

```xml
<SCHEMA>
  <GENERIC>
    <!--This part will represent all attributes that are common for all type of documents.-->  </GENERIC>
  <TYPE1>
    <!--This part will represent all attributes that are common for type 1 only (e.g. invoices).-->  </TYPE1>
  <TYPE2>
    <!--This part will represent all attributes that are common for type 2 only.-->  </TYPE2>
  <TYPE3>
    <!--This part will represent all attributes that are common for type 3 only.-->  </TYPE3>
  <GENERIC>
    <!--This part will represent attributes common for type 1 only (e.g. invoices).-->  </GENERIC>
  <GENERIC>
    <!--This part will represent attributes common for type 2 only.-->  </GENERIC>
  <GENERIC>
    <!--This part will represent attributes common for type 3 only.-->  </GENERIC>
 <SCHEMA>
```

In this paper, we have identified and represent in the generic schema, three types of business documents: invoices, contracts and catalogues.

The generic schema (GS) that represents several types of documents is a combination of m generic attributes GA= \{GA_i, 1 \leq i \leq m\} and n type attributes TA=\{TA_j, 1 \leq j \leq n\}, where:

- \( \forall i \in \{1, \ldots, m\} \), \( CA_i \), represents a common attribute that exists in almost all documents.
- \( \forall j \in \{1, \ldots, n\} \), \( CSTA_j \), represents common attributes that only exist within the same type of documents.
- \( \forall i \in \{1, \ldots, m\} \), and \( \forall j \in \{1, \ldots, n\} \), \( GA_i \subset CSTA_j \).
- \( \forall m \in \{1, \ldots, m\} \), and \( \forall n \in \{1, \ldots, n\} \), \( CSTA_{mn} \subset TA \).

- \( \forall i \in \{1, \ldots, m\} \), and \( \forall j \in \{1, \ldots, n\} \), \( CSTA_{ij} \subset TA \).

B. Schema Matching Method

To match XML schemas from data warehouse and from the unstructured documents, our method runs a tokenization process. After that we use stemming algorithm in [17] to reduce a word to its stem or root form. Then elements names of both schemas can be checked for possible semantic matching. We choose Lin and Wu and Palmer measures because they can show similarity and dissimilarity by showing score one to zero. Most values produced by other WordNet measures do not have defined upper bound. Furthermore, Lin and Wu and Palmer have different approaches to calculate the similarity. Lin is an information content measure and Wu and Palmer is a path measure.

Due to the fact that all WordNet measures produce numeric value as a way to calculate the similarity, predefined thresholds must be used as a cutoff between semantically similar and dissimilar elements. However, there is no standard threshold for all WordNet measures, due to the different behavior of each measure. This is admissible because each WordNet metric calculates the similarity by a different approach. There are few literatures [22-24] that suggested an optimal threshold for Lin and Wu and Palmer measures. According to [22] Lin optimal threshold is 0.9. Whereas, in [23], authors suggested 0.89 for Lin and 0.96 for wup as an optimal thresholds. However, in [24], authors argued that Lin accuracy is drop-off constantly when lowering the threshold. It appears that 0.9 for Lin and 0.7 for wup were good in terms quality and quantity data. Hence, after testing the above mentioned thresholds we decide to use 0.89 and 0.94 for lin, wup respectively, to perform the semantic matching between the two schemas’ elements.

IV. IMPLEMENTATION AND EVALUATION

A prototype for capturing the semantic similarities between two schemas is implemented using NetBeans IDE. This prototype relies on a Java based version of WordNet::Similarity in [25] called JWS. The matching prototype accepts two schemas and use elements and attributes of each schema to capture and output the semantically matching elements. The prototype then calculates the semantic similarity using Lin and Wu and Palmer measures. The prototype can be easily adapted to use another WordNet measures. Then, by comparing the semantic value with predefined thresholds the prototype can conclude whether two elements are semantically related or not.

We match our generated schema from unstructured documents with five different data warehouse schemas found in [26-29]. The aim is to demonstrate how the proposed generic schema is best represented to integrate with any business related data warehouse. Fig. 2 depicts the size of matching elements between these data warehouses’ schemas and our generic unstructured documents XML schema. The first data warehouse is a star schema in [26] for sales of products geographically. It is clear that using threshold value of 0.89 for wup is unsuitable. This makes wup scale permissive and consequently lead to match two elements with less degree of similarity. For instance, wup with threshold 0.89 considered “Family” and “Organization Name” similar with score of 0.917, which demonstrates the need to increase the threshold value. Lin measure however, was unchanged.

![Figure 2. Matched Elements between Data Warehouse Schemas and Generic Unstructured Documents Schema.](image-url)
The second data warehouse schema that has been matched with the generic unstructured document schema is another data warehouse for products sales [30]. The difference between \textsl{lin} and \textsl{wup} with threshold of 0.94 was not significant. However, \textsl{wup} with threshold 0.89 still behave permissively. For example “Product” and “Total” are similar based on 0.89 threshold.

The third data warehouse schema used in the evaluation is web log snowflake schema in [27]. The matching results this time show a slight difference between \textsl{lin} and \textsl{wup} at threshold value 0.94. However, \textsl{wup} asserts that threshold of 0.89 is not appropriate by considering the poor matching results. For example, “name” and “signature” are considered similar. \textsl{Lin} in this result shows the same number of matching using both thresholds.

The forth data warehouse in [28] is star schema for E-commerce. The result of matching with the generic schema shows that result of matching by \textsl{lin} and \textsl{wup} with threshold of 0.94 is still marginal. On the other hand using threshold of 0.89 for both measures shows considerable difference.

The fifth and final data warehouse in [29] is a star schema of parts sales with different suppliers in different time schema. The result of the matching with the proposed generic schema indicates yet again that \textsl{lin} and \textsl{wup} metrics are almost similar with 0.94 as a threshold. This clearly indicates that decreasing the \textsl{wup} threshold would lead to a drop-off in the accuracy of matching. This is due to the fact that the matching at this threshold accepts low similarity degree of matching, which results in poor matching results.

It is observed that \textsl{lin} measure does not significantly change with threshold’s change. It is due to the fact that \textsl{lin} is based on information content obtained from sense–tagged corpora. While the outcome of \textsl{wup} is significantly varies since it is a path measure. Therefore, \textsl{lin} will be used to demonstrate the need to different partitions inside the generic schema.

Matching all data warehouses with each partition of the generic schema in Fig. 3, indicates that the generic partition is the least section that contributes to the matching. Hence, excluding it may reduce the time required to process the matching using WordNet.

Based on the evaluation outcome, the scalability of processing time if we use the overall generic schema is rather poor. However, using particular partition result in a linear increase as appeared in Fig. 4 and 5. Hence, the processing time can be reduced if there is predetermined knowledge of the nature of a data warehouse and thus selects the related partitions instead of the entire generic schema.

V. CONCLUSION AND FUTURE WORK

This paper proposes an approach to facilitate integration of data warehouses with unstructured documents. We propose a new generic XML schema model for various types of unstructured documents for this integration purpose. The new generic schema is designed into separate partitions grouped by the type of the documents. In order to achieve the integration, we use linguistic matching mechanism using WordNet dictionary to find possible similarities between elements’ names of both schemes. The outcome of our paper shows that semantic matching is essential as an initial process to calculate the degree of similarity between data warehouse schema and our generic schema.

Based on our initial study, we found that relying only on WordNet for the integration incurs high processing time. Therefore, for the future work we will implement structure
similarity together with the linguistic similarity reduces the time as well as to increase the accuracy of matching.

REFERENCES