J Intell Inf Syst
DOI 10.1007/s10844-014-0351-2

A framework for enriching Data Warehouse analysis with Question Answering systems

Antonio Ferrández · Alejandro Maté · Jesús Peral · Juan Trujillo · Elisa De Gregorio · Marie-Aude Aufaure

Received: 12 November 2013 / Revised: 16 November 2014 / Accepted: 9 December 2014

Abstract Business Intelligence (BI) applications allow their users to query, understand, and analyze existing data within their organizations in order to acquire useful knowledge, thus making better strategic decisions. The core of BI applications is a Data Warehouse (DW), which integrates several heterogeneous structured data sources in a common repository of data. However, there is a common agreement in that the next generation of BI applications should consider data not only from their internal data sources, but also data from different external sources (e.g. Big Data, blogs, social networks, etc.), where relevant update information from competitors may provide crucial information in order to take the right decisions. This external data is usually obtained through traditional Web search engines, with a significant effort from users in analyzing the returned information and in incorporating this information into the BI application. In this paper, we propose to integrate the DW internal structured data, with the external unstructured data obtained with Question Answering (QA) techniques. The integration is achieved seamlessly through the presentation of the data returned by the DW and the QA systems into dashboards that allow the user to handle both types of data. Moreover, the QA results are stored in a persistent way through a new DW repository in order to facilitate comparison of the obtained results with different questions or even the same question with different dates.

A. Ferrández · A. Maté · J. Peral (✉) · J. Trujillo · E. De Gregorio
Department Languages and Information Systems, University of Alicante, Carretera San Vicente S/N,
Alicante 03080, Spain
e-mail: jperal@dlsi.ua.es

A. Ferrández
e-mail: antonio@dlsi.ua.es

A. Maté
e-mail: amate@dlsi.ua.es

J. Trujillo
e-mail: jtrujillo@dlsi.ua.es

E. De Gregorio
e-mail: edg12@alu.ua.es

M.-A. Aufaure
MAS Lab Ecole Centrale Paris, Grande Voie des Vignes, 92290 Châtenay-Malabry, France
e-mail: marie-aude.aufaure@ecp.fr

Published online: 23 December 2014

Keywords Business Intelligence · Data Warehouse · Question Answering · Information Extraction · Information Retrieval

1 Introduction and motivation

Nowadays, the available information, mainly through the Web, is progressively increasing. According to the 2011 Gartner Group report (Gartner Group report 2011), worldwide information volume is growing annually at a minimum rate of 59% annually. Thus, the information that could be potentially used by a company is progressively increasing. This information is accessible from any computer, and an important percentage of this information is unstructured and textual, such as the one generated by Social Networks (e.g. Twitter or Facebook). The structured data is predetermined, well defined, and usually managed by traditional Business Intelligence (BI) applications, based on a Data Warehouse (DW), which is a repository of historical data gathered from the heterogeneous operational databases of an organization (Inmon 2005; Kimball and Ross 2002). The main benefit of a DW system is that it provides a common data model for all the company data of interest regardless of their source, in order to facilitate the report and analysis of the internal data of an organization. However, there is a wide consensus in that the internal data of organizations to take right decisions is not enough, even more in current highly dynamic and changing markets where information from competitors and clients/users is extremely relevant for these decisions. Thus, the main disadvantage of traditional DW architectures is that they cannot deal with unstructured data (Rieger et al. 2000). Currently, these unstructured data are of a high relevance in order to be able to make more accurate decisions, since the BI applications would empower their functionality by considering both data from inside the company (e.g. the reports or emails from the staff stored in the company intranet) and outside (e.g. the Webs of the company competitors) (Trujillo and Maté 2012). For example, let us consider a scenario where an enterprise needs to compare its product prices (internal structured DW data) with those of the competence (external unstructured data obtained from the Web) for making new promotions.

So far, many attempts to integrate a corporate DW of structured data with unstructured data have been reported (Badia 2006; Henrich and Morgenroth 2003; McCabe et al. 2000; Pérez-Martínez 2007; Pérez-Martínez et al. 2008a, b, 2009; Priebe and Pernul 2003a, b; Qu et al. 2007; Rieger et al. 2000). They are mainly based on systems that use Natural Language Processing (NLP) techniques to access the unstructured data in order to extract the relevant information of them but they do not reach a full integration of structured and unstructured data as our proposal manages.

In this paper, we propose to integrate the DW internal structured data, with the external unstructured data obtained with Question Answering (QA)[1] systems. We start with a question or query in Natural Language (NL) posed by the decision maker, who also identifies the sources where to search the required information. We distinguish between *queries* and *questions* in order to highlight that a query refers to a request of data to the DW system, whereas a question requests data to the QA system. The former are likely to be much more rich and complex than simple questions, which may force to divide the query into several questions. The questions are analyzed by the Distributor/Integrator service of the framework and are passed to the corresponding node (e.g. the QA node to access external data or the DW node to access internal data). Then, each node processes the question in an autonomous way

[1] Question Answering systems represent the potential future of Web search engines because QA returns specific answers as well as documents. It supposes the combination of IR and IE techniques.

on its corresponding sources. Once the system receives all the results from the nodes, like internal DW, Web services or API's, it is capable of integrating and showing a dashboard to the user that allows him/her to take the right decision. Finally, let us add that we also take advantage of our unique well-checked hybrid method to build data warehouses by considering (i) user's requirements and (ii) data sources, thereby guarantying that the query posed on the DW will return the correct data required by the decision maker (Mazón and Trujillo 2008; Mazón et al. 2007).

The paper is structured as follows. In Section 2, we summarize the most relevant related work regarding combining traditional DWs with unstructured data. In Section 3, we introduce our framework for analyzing and integrating different data sources into a common dashboard. In Section 4, and in order to clarify our proposal, we introduce the case study that will be evaluated in Section 5, where we provide deep detail on the evaluation of the application of our proposal. We conclude the paper with the summary of our main contributions and our directions for future works.

2 Related work

Several attempts to integrate search of structured and unstructured data have arisen, where a DW and an Information Retrieval (IR)[2] system are connected, such as the work presented in (Rieger et al. 2000) and (Henrich and Morgenroth 2003). However, as it is claimed in the work presented in (McCabe et al. 2000), those efforts do not take advantage of the hierarchical nature of structured data nor of classification hierarchies in the text, so they implement an IR system based on a multidimensional database. Specifically, they focus on the use of OLAP techniques as an approach to multidimensional IR, where the document collection is categorized by location and time. In this way, they can handle more complex queries, like retrieving the documents with the terms “financial crisis” published during the first quarter of 1998 in New York, and then drilling down to obtain those documents published in July 1998.

In (Priebe and Pernul 2003a, b), authors propose an architecture that introduces a communication bus where both systems publish their output. Each system picks up this output and uses it to show related information. For example, the query context of a DW access is used by an IR system in order to provide the user with related documents found in the organization's document management system. In order to solve the problem of the heterogeneity of both systems, they propose to use ontological concept mapping (e.g. the DW system uses “owner” for what is called “author” within the document metadata). They use an ontology for the integration, but it is only oriented to communicate both applications in enterprise knowledge portals. In this way, they handle queries like “sales of certain audio electronics products within the four quarters of 1998”.

In (LaBrie and St. Louis 2005), an alternative mechanism for IR (“dynamic hierarchies” based upon a recognition paradigm) that overcome many of the limitations inherent in traditional keyword searching is proposed. This IR approach was used in BI applications but no integration between both applications was made.

In (Pérez-Martínez 2007; Pérez-Martínez et al. 2008a), authors provide a framework for the integration of a corporate warehouse of structured data with a warehouse of text-rich XML documents, resulting in what authors call a contextualized warehouse. These works are based

[2] Information Retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. This activity is currently quite popularized by the Web search engines as Google.

on applying IR techniques to select the context of analysis from the document warehouses. In (Pérez-Martínez et al. 2009), authors formalize a multidimensional model containing a new dimension for the returned documents. To the best of our knowledge, these papers are the most complete ones in combining and considering structured and unstructured data in a common DW architecture.

Regarding Information Extraction (IE),[3] (Losiewicz et al. 2000) surveys applications of data mining techniques to large text collections, including IR from text collections, IE to obtain data from individual texts, DW for the extracted data. In (Badia 2006), different IE-based (as well as IR) methods for integrating documents and databases are discussed. Specifically, the author proposes IE as the right technology to substitute IR, which fills the slots of a set of predefined templates that determines the information that is searched in the collection of documents. In (Bhide et al. 2008), authors claim that there exist BI products like QL2 (QL2, 2013) and IBM Business Insights Workbench (BIW) (IBM. Business insights workbench, 2013) that try to derive context from the unstructured data by using various IE and clustering techniques. However, no business intelligence product has tried to exploit context available in the structured data of the enterprise in order to allow us a seamless analysis of both structured and unstructured data fully integrated, in a consolidated manner. They propose the use of IE techniques to a specific task of linking common entities in a relational database and unstructured data.

With regard to work on the integration of DW and Question Answering (QA) systems, in (Qu et al. 2007), a scheme about a DW design based on data mining techniques was put forward in order to overcome the defects of current Chinese QA systems. In (Roussinov and Robles-Flores 2004), authors explored the feasibility of a completely trainable approach to automated QA on the Web for the purpose of business intelligence and other practical applications. They introduce an entirely self-learning approach based on patterns that do not involve any linguistic resources. In (Lim et al. 2009), the authors present a study of comparative and evaluative queries in the domain of Business Intelligence. These queries are conveniently processed by using a semantic interpretation of comparative expressions and converting them to quantifiable criteria, in order to obtain better results in a QA system for this domain. In our previous work of (Ferrández and Peral 2010), we analyzed the main benefits of integrating QA systems with traditional DW systems in order to be able to complete internal data with precise returned answers from QA systems, instead of returning whole documents provided by IR systems.

Several work on NL questions to query the Semantic Web have been carried out, like Aqualog (Lopez et al. 2005), SQUALL (Ferré 2012) or FREyA (Damljanovic et al. 2012), which use SPARQL for querying knowledge bases built in RDF. In PANTO (Wang et al. 2007) and Querix (Kaufmann et al. 2006), they accept generic NL questions and outputs SPARQL queries.

2.1 Contributions of our proposal to previous work

We overcome the data integration problems identified in previous work through the following four contributions. Contribution 1 is that we use QA in order to access to the unstructured data. We consider QA more suitable than only IR because the integration of whole documents

[3] Information Extraction is the task of automatically extracting specific structured information from unstructured and/or semi-structured machine-readable documents. A typical application of IE is to scan a set of documents written in a natural language and populate a database with the information extracted (e.g. the name of products and their prices).

returned by IR is weaker and less useful to the decision maker, since the information provided by QA is much more specific, and thus, can be integrated seamlessly into DW cubes. Moreover, we consider QA more suitable than IE because of the QA flexibility to afford any kind of question, and not only a set of predefined templates.

With regard to contribution 2, we deal with the weak point about the lack of full integration between systems that access the unstructured data (e.g. QA), whether it is external or internal, and the ones that access the structured data (DW). In this way, we allow the decision maker to compare both the internal data of a DW and the data gathered from the Web. This aim is managed by our proposed framework that completes the whole flow of data.

In contribution 3, we have improved the interaction with the user through: (i) the outputs of the nodes are fully integrated and presented to the user in a friendly dashboard (Eckerson 2007), which allows the decision maker to immediately compare internal data of a company against competitors; (ii) our NL interface (Llopis and Ferrández 2012) outdoes previous work by its full portability to different DW systems; and by its query-authoring services. These services dramatically improve the system usability allowing the decision maker to early detect errors in the question by automatically distinguishing between linguistic (e.g. when the grammar in the interface cannot parse a question) and conceptual (e.g. entity-property mismatch, data type mismatch, etc.) failures.

Finally, in contribution 4, we have proved and evaluated the feasibility of our approach on the case scenario of an enterprise's marketing department that needs to compare its product prices with those of the competence for making new promotions. These competitors' prices are obtained from the Web through the QA system. Therefore, from the initial request of data of "What is the price of the Canon products in the sales period?", our proposal can obtain the cube from the enterprise's DW, and the QA database with the competitors' prices, where both results are integrated into a dashboard that immediately allows the user to analyze and compare them. Moreover, it can transform the initial DW query into the set of questions formed by the products present in the DW scheme, such as "What is the price of the Canon Pixma in the sales period?"

3 Our business intelligence framework

In our framework (Fig. 1), we can distinguish two phases: (i) the system setup and (ii) the running phase, which are detailed in the next two subsections.

The setup phase prepares the source nodes, where the required information will be searched, by creating the corresponding ontologies. It is important to emphasize that several

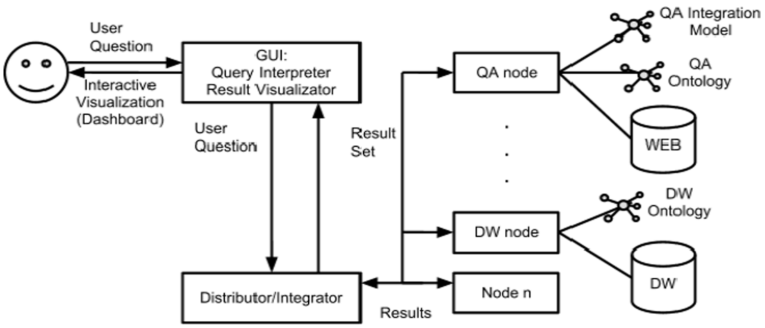


Fig. 1 Framework to access/integrate structured/unstructured and internal/external data

DW, QA or Big Data source nodes can be connected, each one with its own implementation, model and domain (e.g. we can connect a QA node specialized in electronic products as well as a QA node specialized in legal domains). These ontologies are created just the first time that the source node is connected in our framework.

In the running phase, the user or decision maker (i) poses a NL question through the GUI (Graphical User Interface) element and (ii) selects the sources to be searched (e.g. in a specific database or DW, or in a specific QA domain). The GUI element passes the NL question to the Distributor/Integrator element that also sends it to the set of specialized nodes (e.g. the DW and QA nodes). Each specialized node disposes of the proper interface in order to process adequately the NL question and to produce the suitable output information. Then, the Distributor/Integrator coordinates the running of each specialized node, gathering the output of these nodes in order to send the fused information to the GUI element. Finally, the GUI is responsible for displaying the results as a dashboard, that integrates both external and internal data.

This paper complements our approach to access different sources shown in (Maté et al. 2012a (i) by reaching the full integration of unstructured and structured information through the ontologies and (ii) by displaying the data integration with a dashboard. In () the authors describe an approach based on the MapReduce strategy (Dean and Ghemawat 2008) where the query is divided and distributed to different nodes and then it integrates the results; this approach allows to maintain the internal structure of the different nodes, allowing to add or remove the nodes in a seamlessly way. A similar proposal is (Abelló et al. 2011) where the authors present a framework for create cubes using MapReduce; this proposal differs from ours, where we consider the cube with the OLAP server a single node. For more information on theoretical foundation see (Gray et al. 1997).

3.1 Setup phase

In this phase, the specialized source nodes, both DW and QA, are prepared just the first time that they are connected to our framework, in order to integrate them in the global system. In each QA node, we create (i) its QA integration model and (ii) its QA ontology; whereas in each DW node we create its DW ontology that describes the DW scheme, which will allow its integration with the QA nodes through a semi-automatic mapping process that detects connections between the QA and DW ontologies.

QA node (i) The QA integration model contains information about the answer that is returned to the Distributor/Integrator element in order to be integrated with the data returned by the DW node. For example, Fig. 2 depicts a QA integration model that contains the answer (as a noun phrase and as a string of fixed size), the expected answer type (e.g. the "economic" type for the question "What is the price of the Canon products

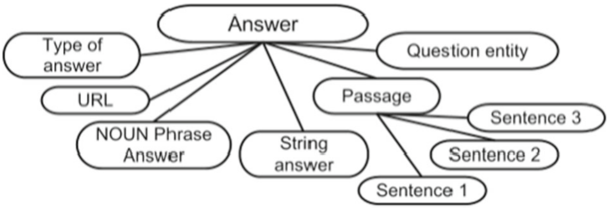


Fig. 2 QA integration model

in the sales period?"), the entities detected in the question (e.g. "Canon product" as an "object-electronic product"), the URL or document that contains the answer and the passage or answer context (i.e. the surrounding text around the answer, with which the user can decide whether the answer is correct for its purposes without reading the whole document). The QA integration model can vary in different QA systems. For example, a QA system can return an answer context of three sentences (such as the one depicted in Fig. 2), whereas other QA systems can return only a fixed number of words around the answer.

QA node (ii) The QA ontology contains information about the set of answer types considered in the QA system. For example, Fig. 3 depicts an excerpt of an answer ontology, where a set of WordNet top concepts (e.g. object or person) are used with some extensions (e.g. economic or percentage type in the numeric type).

DW node The DW ontology (Santoso et al. 2010) is created, which will allow us to analyze an integrated view of data. The ontology relates the tables and attributes considered as the internal data. In Fig. 4, an excerpt of a DW ontology is shown.

QA and DW ontology mapping. Finally, a semi-automatic mapping process is carried out in order to detect connections between the QA and DW ontologies (Wang et al. 2007) (see Fig. 5):

(a) We detect equivalent classes/properties in both ontologies. Firstly, the exact matches between the two ontologies are retrieved (e.g. in Fig. 5 the equivalent classes "Day, Month, Year" are detected since they appear in Figs. 3 and 4). After that, the remaining concepts are matched using the information of the lexical-semantic resources used in QA (WordNet, lexicons, dictionaries, glossaries, etc.) and prompting the user to confirm the match. For example, in Fig. 5, the equivalence is found: between the classes "Electronic Product" and "Object" thanks to the hyperonym WordNet relation between "product" and "object". Similarly, the equivalent property Price in DW vs. Economic in QA is established;

(b) We add new subclasses –extracted from the DW ontology– in the QA ontology (e.g. "Electronic Product" in DW, which is added to the object answer type, because of the mentioned WordNet hyperonym relation between "object" and "product");

(c) We enrich the lexical-semantic resources used in QA with instances from the DW ontology (see Fig. 6). In the Figure, the enrichment of WordNet can be seen, where the instances of electronic products stored in the DW (Asus P5KPL-AM EPU, etc.) are added to the lexical resource. In this way, questions about these new instances can be treated by the system.

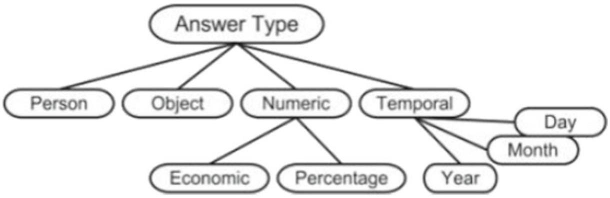


Fig. 3 QA ontology

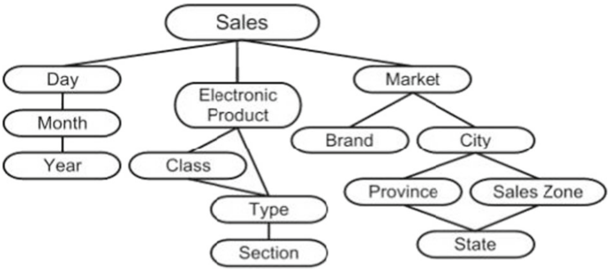


Fig. 4 DW ontology

3.2 Running phase

The GUI element Firstly, the GUI element receives the NL request of data through our NL interface (Llopis and Ferrández 2012), which thanks to its query-authoring services improves the system usability allowing the decision maker to early detect errors in questions by automatically distinguishing between linguistic (e.g. errors due to lexical or syntactic mistakes) and conceptual failures (e.g. errors due to the lack of an specific relation between tables in the DW). Secondly, the decision maker selects the sources to be searched for the required information.

Then the Distributor/Integrator performs a coordinator role by distributing the NL request of data to each DW and QA node; and by receiving and creating an integrated view of the data returned from all nodes.

The DW node The NL query is transformed into a MultiDimensional eXpression (MDX), which can be interpreted by the OLAP engine. This transformation is performed by combining NL processing tasks with schema matching techniques (Maté et al. 2012b; Rahm and Bernstein 2001). First, the system analyzes the NL query. The analysis aims to match the main concepts involved in the query with those in the DW schema. The mapping is performed first by retrieving the exact matches from a Business Dictionary (Maté et al. 2012b). Then, the remaining concepts are matched with those in the DW schema by means of expansion using the DW Ontology (Fig. 4) and WordNet (Fig. 6). Finally, the query is reformulated as a valid controlled language expression (Maté et al. 2012b). If a word is not found in the Business Dictionary and cannot be matched against the schema, the user will be prompted to introduce a match. For example, consider the query "What is the price of Canon products in the sales

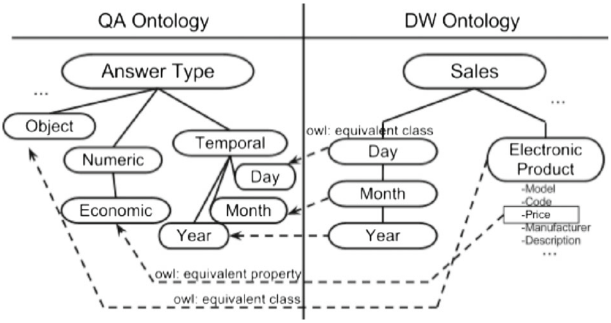
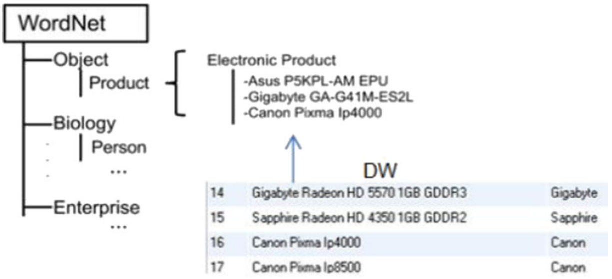


Fig. 5 Mapping between subsets of QA ontology and DW ontology



14	Gigabyte Radeon HD 5570 1GB GDDR3	Gigabyte
15	Sapphire Radeon HD 4350 1GB GDDR2	Sapphire
16	Canon Pixma Ip4000	Canon
17	Canon Pixma Ip8500	Canon

Fig. 6 Enriching QA lexical-semantic resources with knowledge from the DW

period?" The main concepts involved are "price", "Canon", "products", and "sales period". The first concept, "price", matches with the attribute "Price" of the "Electronic Product" level in Fig. 5. Next, "Canon" cannot be matched to any element in the schema, thus it is expanded by means of WordNet and identified as an instance of "Electronic Product Manufacturer" (see Fig. 6). Afterwards, "product" is found in the Business Dictionary as a synonym of "Electronic Product". Finally, "sales period" is not found in the dictionary nor using the expanded search. Thus, the user is prompted to introduce a formal definition for the word or modify the query. In case that the user introduces "with month equal to January or month equal to July" as a definition for the "sales period", as a result, the initial query is transformed into the controlled language sentence "price of Electronic Product with manufacturer equal to Canon and (with month equal to January or month equal to July). This sentence can then be interpreted by a controlled language grammar similar to the one in (Maté et al. 2012b) that transforms sentences into MDX queries. As a result, the DW node returns a cube which contains the information specified by the NL query, which can be navigated using the traditional OLAP operations, such as roll-up or drill-down.

The QA node The NL question is internally processed through a set of NLP tools (e.g. POS-taggers or partial parsing) in order to detect the type of the answer to be searched (e.g. for the previously mentioned question "What is the price of the Canon products in the sales period?", given the "economic" answer type, it supposes that the searched information consists of a numeric string followed by a currency symbol such as € or $), as well as the most salience entities in the question (e.g. "Canon products" as an entity of object type). After that, the processed answer is posed to an Information Retrieval tool in order to obtain the set of documents that is more likely to contain the answer. These documents are analyzed in order to extract a set of answers sorted by the probability of correction certainty. The extraction process is specialized for each answer type. For example, in the case of the "economic" type, for the previously mentioned question, several patterns are used: a) "Canon Pixma price: 240 €"; b) "Table of prices…Canon Pixma…240€". Finally, the set of answers extracted by the QA system is stored in a database (Stanojevic and Vraneš 2012; Kerui et al. 2011) with the structure defined in the QA integration model (see Fig. 2) as it is explained in the following step.

The integration of the results Once the running of each DW and QA node is finished, the Distributor/Integrator element creates an integrated view of the data returned from both nodes. In order to integrate the results from both the QA and the DW without storing the information directly into the DW, a transformation must be made. DWs represent information in a multidimensional manner, whereas QA retrieves information in a table format. Therefore, we apply the following process. First, we lower the dimensionality of the DW information retrieved by transforming the DW cube into a table (i.e. flattening process). This process is formalized as follows:

Let $C = \{M, D\}$ be a cube where M is a set of measures represented by the cube and D is a set of dimensions that determine the coordinates of the cube. A Relation R containing the equivalent information can be obtained by the following process. For each level selected L_j in dimension $d_i \in D$, a column is created in R. Afterwards, the columns corresponding to the measures $m_n \in M$ are created. Finally, R is populated by a set of tuples $n_1 \ldots n_n$ where the domain of each column $c_j=\{L_j\}$ for the columns corresponding to the dimensions and $c_n=\{m_n\}$ for the columns corresponding to the measures. A similar result can be obtained in current BI tools by pivoting all dimensions to one side of the pivot table.

After that, we have obtained a compatible representation of the DW data and a set of union points (that we have called connections and are identified by means of the ontological mappings as it is depicted in Fig. 5). In the next step, the user filters the QA results and selects those elements that the decision maker considers relevant to be joined to the flattened DW cube through the union points in a resulting table created on the fly: DW⋈QA (where the symbol ⋈ indicates the natural join between the two tables). Therefore, the DW system is not altered in any way, keeping the data clean and avoiding being affected by inaccuracies in the information retrieved by the QA system.

Finally, the dashboard (feeding on the mentioned joined table) shows both data from inside the company and the competitors. Moreover, these connections points would allow the automatic generation of new questions, such as the questions about the specific electronic products stored in the DW (e.g. “What is the price of the Canon Pixma in the sales period?”), which facilitates to focus only on the products sold by the user.

Repository of questions Our approach stores the QA results in a persistent way through a new DW repository. This repository is created from the QA integration model (Fig. 2) and a generic set of dimensions. The logical design has four dimensions: *Date*, contains the information about when the question was made; *Query*, with the NL question; *Fields*, with the QA integration model fields and the union points; and one degenerated dimension with *ID*, that links with the specific NL question and the QA rows obtained in a concrete date. The fact table of this repository has the elements retrieved after the matching phase. The purpose of this repository is double: on the one hand, the external data obtained through the QA system are stored in a permanent way in order to have a historical file with relevant data to the different questions, overcoming the intrinsic dynamic character of the external information (e.g. the Webs of the enterprise’s competitors); on the other hand, a comparison of the obtained results with different questions or even the same question with different dates can be made.

Advantages of our proposal The main advantages of this integration of results are: (1) the decision maker can browse all the information (passage, context, precise answer, etc.) about every tuple of the QA database so the user does not need to explore the whole document; (2) the user can delete the incorrect tuples returned by the QA node; (3) new questions can be automatically generated from the instances stored in the DW taking into account the ontology integration and the detected question entities; and (4) the connections between the QA and DW ontologies have been detected in order to facilitate the data integration.

Finally, it is important to emphasize the modularity and scalability of our framework. It is independent of the DW and the QA systems specifically used, because the integration of these systems is carried out by the detected connection points between the respective ontologies, thereby having a more integrated and scalable view of internal and external data. Furthermore, several QA nodes can be used and, subsequently, several QA databases are shown to the user in the dashboard. Moreover, the user can easily store different questions and results (DW cube and QA database), allowing the user to save time in the access and analysis of external information.

4 A case scenario

4.1 The case scenario description

After introducing the system architecture, we illustrate the application of our framework, and later we will evaluate it through the following case scenario: an enterprise's marketing department wants to analyze sales to identify possible features useful for making new promotions. The corresponding model for this scenario, shown in Fig. 7, is based on a UML profile for modeling DWs presented in (Luján-Mora et al. 2006). DW models structure data according to a multidimensional space, where events of interest for an analyst (e.g., *sales*, *treatments of patients…*) are represented as facts which are associated with cells or points in the multidimensional space, and which are described in terms of a set of measures. These measures can be analyzed by means of dimensions which specify different ways the data can be viewed, aggregated or sorted (e.g. according to *time*, *store*, *customer*, etc.). Importantly, dimensions are organized as hierarchies of levels, which are of paramount importance in BI systems in order to empower data analysis by aggregating data at different levels of detail.

Our case scenario models the electronic products bought by customers in different markets throughout the country (Sales fact class). This fact contains several properties which are Fact

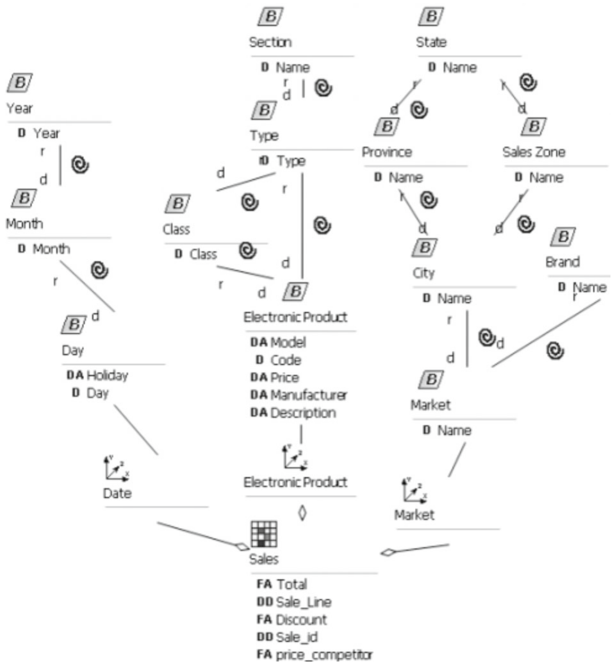


Fig. 7 Excerpt of the multidimensional model for our case scenario on Electronic Product Sales

Attributes (FA): Total, Discount, etc. These properties are measures that can be analyzed according to several aspects as the products (Electronic Product) which were bought or the market (Market) where they were bought and the associated Date. The fact also contains two Degenerated Dimensions (DD). These dimensions are important to differentiate each product bought in a single sale record, but do not provide any additional information. Therefore, these dimensions do not have an associated hierarchy. The rest of the dimensions present one or more hierarchies, either one or multiple aggregation paths. The products can be aggregated to the class level only on certain products, since not all of them have a class. On the other hand, the market dimension presents alternative hierarchies, and can be aggregated either by cities or by the brand associated to the market.

Given this UML model, users (the decision makers) can request a set of queries to retrieve useful information from the system. For instance, they are probably interested in getting the sales zones with most sales. Many other queries can be similarly defined to support the decision making process. However, the allowed queries are constrained by the information contained in the schema in such a way that other important information may be missed. For example, the following scenario is likely to happen: the company wants to maximize benefits by selling products just a bit cheaper than its competitors, offering interesting promotions (i.e. if you find this product cheaper, we give your money back), and they want to analyze their sales according to the rival markets and their prices. Normally, the company has not any internal report about the present prices of every competitor. However, it is likely to obtain this information from the Web.

4.2 The application of our proposal on the case scenario

Let us apply our framework detailed in section 3 to this case scenario supposing that the following user's NL request of data is formulated: "What is the price of the Canon products in the sales period?"

Setup phase. QA and DW node With regard to the system setup phase, on the one hand, in the QA node, the QA integration model and the QA ontology of answer types are generated in Figs. 2 and 3 respectively. As it can be seen in these Figures, the QA integration model specifies: the answer type, the entities detected in the question, the URL or document identifier, the noun phrase and the passage (formed by three sentences) that contains the answer. On the other hand, in the DW node the DW ontology is created (Fig. 4).

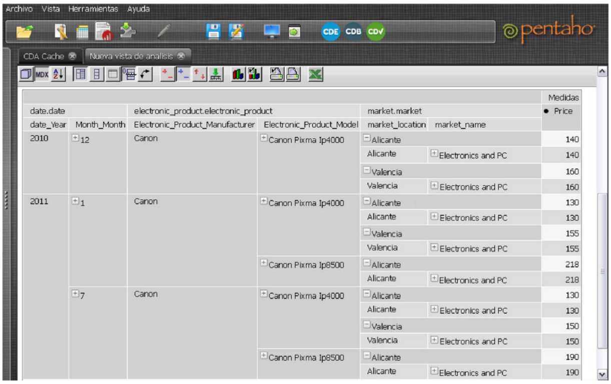
Setup phase. QA and DW ontology mapping Next, the connections between the DW and QA ontologies are detected. In Figs. 5 and 6 can be seen: (a) two equivalent classes in both ontologies (date vs. temporal and electronic product vs. object) and an equivalent property (price vs. economic); (b) two new subclasses are added in the QA ontology: electronic product and market; (c) the lexical-semantic resource used in QA is enriched with the set of markets or electronic products stored in the DW.

Running phase. The GUI and Distributor/Integrator element In the running phase, the GUI element receives the NL request of data, which is distributed to each specialized node by the Distributor/Integrator element.

Running phase. The DW node In the DW node, the NL query is transformed into MDX as presented in section 3.2., and the cube shown in Fig. 8 is returned. In this scenario the following MDX query is obtained:

```
with
member [Measures].[Price] as
[electronic_product].[Electronic_Product_Model]
.CurrentMember
.PROPERTIES("Electronic_Product_Price")
select NON EMPTY {[Measures].[Price]} ON COLUMNS,
NON EMPTY Crossjoin({[date].[2010].[12], [date].[2011].[1],
[date].[2011].[7]}, Crossjoin({[electronic_product].[Canon].
Children},
[market].[market_location].Members)) ON ROWS
from [sales]
```

Running phase. The QA node In the QA node, the NL question is processed, and its output is structured as the QA integration model specifies. It returns "economic" type as the answer type according to the QA ontology; the question string "Canon product" as an entity of object-electronic product type; and "the sales period" as an entity of temporal type. Both entities are used to trace and restrict possible right entity solutions of economic type (e.g. when the document contains the noun phrase "sales period"). Then, the set of answers extracted by the QA system is stored in the database shown in Fig. 9, in which the first column ("w") means the confidence of the QA system in this answer (this value ranges between 0 and 1); the second one means the string answer that is extracted from the fourth column that means the noun phrase that contains the answer (e.g., the "218.97_€" price entity is extracted from the noun phrase solution in "218.97_€ con IVA Canon Código de producto" (*Canon 218.97_€ with VAT product code*) thanks to the pattern "Number + Currency"); the third one means the QA system internal code of the Web page; the following three columns represent the passage in which the solution appears. The passage is formed by three sentences, where the sentence 2 contains the answer. In this way, the user has a context to decide whether the answer is right: the text around the solution, as well as the link to the corresponding URL to access the whole document. Therefore the user can filter this QA database by deleting the wrong extracted information. The last two columns mean the question entities extracted in the document by means of a name entity tagger, which can be used as connection points in the integration phase. For example, the "canon pixma 4000/8500" product description is extracted from the passage in "cdr tray canon pixma 4000/8500 …" thanks to the pattern "[Canon] + following modifiers"; or the temporal entity that is extracted from the date that may appear on the document (as it occurs in Fig. 9) or



						Medidas
date.date		electronic_product.electronic_product		market.market		● Price
date_Year	Month_Month	Electronic_Product_Manufacturer	Electronic_Product_Model	market_location	market_name	
2010	⊞12	Canon	⊞Canon Pixma Ip4000	⊟Alicante		140
				Alicante	⊞Electronics and PC	140
				⊟Valencia		160
				Valencia	⊞Electronics and PC	160
2011	⊞1	Canon	⊞Canon Pixma Ip4000	⊟Alicante		130
				Alicante	⊞Electronics and PC	130
				⊟Valencia		155
				Valencia	⊞Electronics and PC	155
			⊞Canon Pixma Ip8500	⊟Alicante		218
				Alicante	⊞Electronics and PC	218
	⊞7	Canon	⊞Canon Pixma Ip4000	⊟Alicante		130
				Alicante	⊞Electronics and PC	130
				⊟Valencia		150
				Valencia	⊞Electronics and PC	150
			⊞Canon Pixma Ip8500	⊟Alicante		190
				Alicante	⊞Electronics and PC	190

Fig. 8 Cube retrieved from the DW

from the date of the URL when the web document was last modified. Finally, we should highlight that the QA table facilitates the user to easily correct the results, for example, normalizing extracted prices to include taxes.

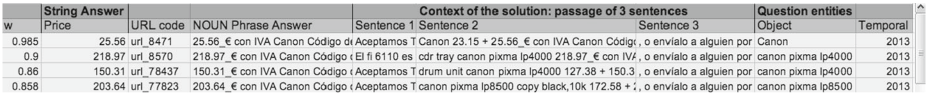
Running phase. The integration of the results The DW cube and the QA database are sent to the Distributor/Integrator element, which merges the different results and sends them to the GUI element. The merge is performed in our scenario as follows. The results obtained from the DW node are obtained in a cube (Fig. 8) that is flattened, obtaining a set of tuples that contain the relevant columns to the query posed, including "Electronic_product_model", "market_location", "market_name", "month_month" and "price". Then, these results are joined with the information recovered from the QA system (Fig. 9). Both results are joined by means of the candidate union points identified in the ontology (see Fig. 5) and selected by the user. The result is a table created on the fly (Fig. 10) that contains the natural join (⋈) between the flattened DW cube and the QA result. By default, the natural join is only carried out with the best result of the QA database and this information is initially shown at the dashboard.

For example, in Fig. 10, using the connection "Electronic_product_model"-"Object", each DW row is joined with the best QA result whose object query entity matches; in the example, the model Canon Pixma Ip4000 is shown with the price (218.97) and answer confidence of 0.9. The Figure also shows another model of Canon product, Canon Pixma Ip8500, and its corresponding price after the match. In case of not matching between these union points, as the one that occurs between DW "Canon Pixma Ip4000" and QA "canon pixma 4000/8500", our current proposal allows the user to perform a cross join in order to combine each row from the DW table with each row from the QA table. In future work, we plan to suggest to the user possible matches according to semantic matching and edit distances between entities.

If other connections were established, like "month_month", every "Electronic_product_model" and "month_month" in the DW will be joined with their equivalent QA results.

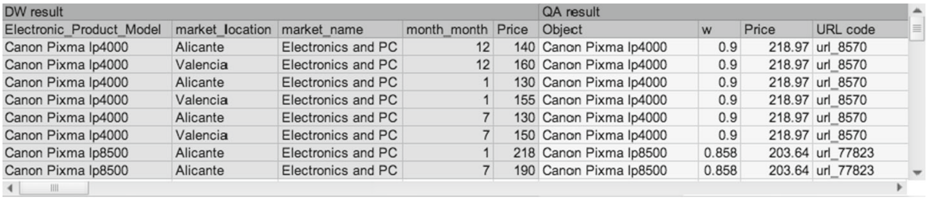
After creating the joined table, the integrated results can be viewed in the dashboard (see Fig. 11). At the top of figure, the user can select the rows to analyze (e.g. in this Figure, the user has selected the first six rows). Additionally, the dashboard allows the user configure the chart fields, such as the X axis as the column "month_month", the title of the chart as "Canon Pixma Ip4000 comparison", the filter column (DW.market_location) and how many QA results will be joined (in this Figure the system joins with the first five QA results sorted by "QA.w"). In the example, the DW.Price and the QA.Price are depicted because the price is the main extracting aim of the query.

Repository of questions The QA database is stored in a persistent way through the new DW repository as well as the date when the question was made, and the NL question. In order to avoid information redundancy, the DW extracted cube is not stored because this information would be easily extracted again whenever the decision maker runs the same query. That is to say, we only stores in the repository of questions, the dynamic external information.



	String Answer			Context of the solution: passage of 3 sentences			Question entities	
w	Price	URL code	NOUN Phrase Answer	Sentence 1	Sentence 2	Sentence 3	Object	Temporal
0.985	25.56	url_8471	25.56_€ con IVA Canon Código de	Aceptamos T	Canon 23.15 + 25.56_€ con IVA Canon Códig	, o envíalo a alguien por	Canon	2013
0.9	218.97	url_8570	218.97_€ con IVA Canon Código	El fi 6110 es	cdr tray canon pixma Ip4000 218.97_€ con IV	, o envíalo a alguien por	canon pixma Ip4000	2013
0.86	150.31	url_78437	150.31_€ con IVA Canon Código	Aceptamos T	drum unit canon pixma Ip4000 127.38 + 150.3	, o envíalo a alguien por	canon pixma Ip4000	2013
0.858	203.64	url_77823	203.64_€ con IVA Canon Código	Aceptamos T	canon pixma Ip8500 copy black,10k 172.58 +	, o envíalo a alguien por	canon pixma Ip8500	2013

Fig. 9 QA database for the question "What is the price of the Canon products in the sales period?"



DW result					QA result			
Electronic_Product_Model	market_location	market_name	month_month	Price	Object	w	Price	URL code
Canon Pixma Ip4000	Alicante	Electronics and PC	12	140	Canon Pixma Ip4000	0.9	218.97	url_8570
Canon Pixma Ip4000	Valencia	Electronics and PC	12	160	Canon Pixma Ip4000	0.9	218.97	url_8570
Canon Pixma Ip4000	Alicante	Electronics and PC	1	130	Canon Pixma Ip4000	0.9	218.97	url_8570
Canon Pixma Ip4000	Valencia	Electronics and PC	1	155	Canon Pixma Ip4000	0.9	218.97	url_8570
Canon Pixma Ip4000	Alicante	Electronics and PC	7	130	Canon Pixma Ip4000	0.9	218.97	url_8570
Canon Pixma Ip4000	Valencia	Electronics and PC	7	150	Canon Pixma Ip4000	0.9	218.97	url_8570
Canon Pixma Ip8500	Alicante	Electronics and PC	1	218	Canon Pixma Ip8500	0.858	203.64	url_77823
Canon Pixma Ip8500	Alicante	Electronics and PC	7	190	Canon Pixma Ip8500	0.858	203.64	url_77823

Fig. 10 Result of the join operation between the DW and the QA results

5 Evaluation

5.1 Description of the QA system

The QA system used for this experiment is called AliQAn, with which we have participated in several CLEF[4] competitions in both monolingual (Roger et al. 2009) and cross-lingual tasks (Ferrández et al. 2009). AliQAn consists of two phases: the indexation and the search phase. The first one is carried out in an off-line mode previous to the search phase, where its main aim is to prepare all the information required for the subsequent phase, in order to speed up as much as possible the searching process. There are two independent indexations, one for the QA process, and another for the IR process. The first indexation involves Natural Language Processing tools in order to reach a better understanding of the documents (e.g. a morphological analyzer such as Maco+[5] or TreeTagger,[6] a shallow parser such as SUPAR (Ferrández et al. 1999) and a Word Sense Disambiguation, WSD, algorithm (Ferrández et al. 2006) that is applied on WordNet/EuroWordNet,[7] EWN). The second indexation is used for the IR tool that filters the quantity of text on which the QA process is applied (AliQAn uses the IR-n system (Llopis et al. 2003)).

With regard to the search phase, it is accomplished in three sequential modules: (1) Question Analysis (2) Selection of relevant passages (3) Extraction of the answer. Module 1 uses the same NLP tools as in the indexation phase (Maco+, SUPAR, WSD and EWN) with the aim of reaching a syntactic analysis of the question, and eliciting its Syntactic Blocks (SBs). These SBs are matched with a set of syntactic-semantic question patterns designed for the detection of the expected answer type and the identification of the main SBs of the question. The answer type is classified into a taxonomy based on WordNet Based-Types and EuroWordNet Top-Concepts. AliQAn's taxonomy consists of the following categories: person, profession, group, object, place city, place country, place capital, place, abbreviation, event, numerical economic, numerical age, numerical measure, numerical period, numerical percentage, numerical quantity, temporal year, temporal month, temporal date and definition. Each taxonomy class stands for the type of information that the answer needs to contain in order to become a candidate answer (e.g. for the "person" type, a proper noun will be required, or for the "temporal" type, a date will be required). The main SBs of the question are used in Module 2 in order to extract the passages[8] of text on which Module 3 will search for the answer. For example, the CLEF 2006 question "Which country did Iraq invade in 1990?" is matched by the pattern "[WHICH] [synonym of COUNTRY] […]", where the "place" answer-type is

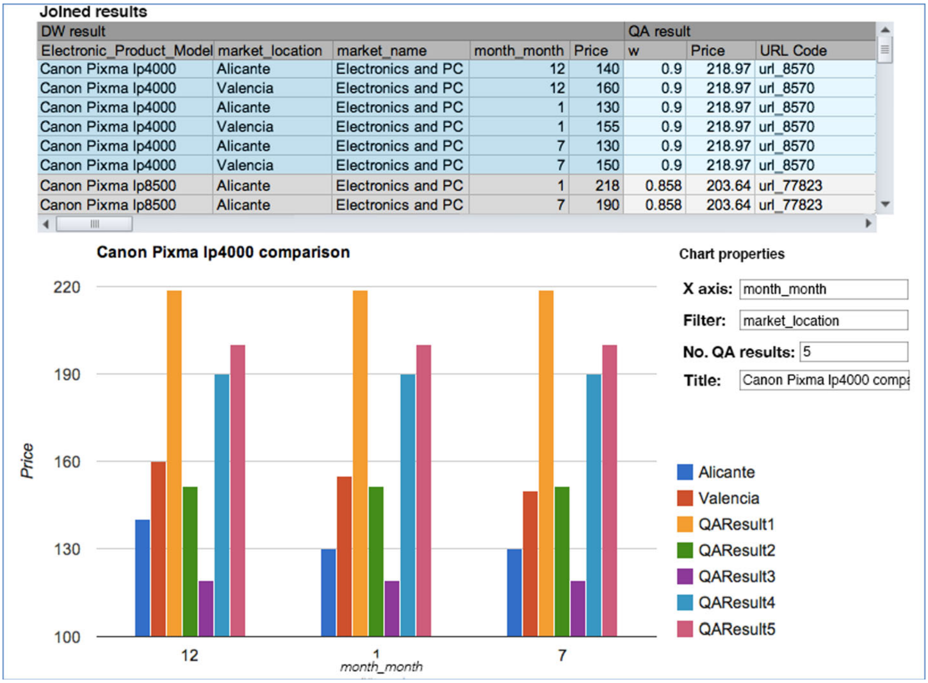
[4] http://www.clef-initiative.eu// (visited on 24th of March, 2013).

[5] http://nlp.lsi.upc.edu/freeling/ (visited on 24th of March, 2013).

[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/ (visited on 24th of March, 2013).

[7] http://www.wordnet-online.com (visited on 24th of March, 2013).

[8] Each passage is formed by a number of consecutive sentences in the document. In this case, the IR-n system (our passage retrieval tool) returns the most relevant passage formed by eight consecutive sentences.



Joined results

DW result					QA result		
Electronic_Product_Model	market_location	market_name	month_month	Price	w	Price	URL Code
Canon Pixma Ip4000	Alicante	Electronics and PC	12	140	0.9	218.97	url_8570
Canon Pixma Ip4000	Valencia	Electronics and PC	12	160	0.9	218.97	url_8570
Canon Pixma Ip4000	Alicante	Electronics and PC	1	130	0.9	218.97	url_8570
Canon Pixma Ip4000	Valencia	Electronics and PC	1	155	0.9	218.97	url_8570
Canon Pixma Ip4000	Alicante	Electronics and PC	7	130	0.9	218.97	url_8570
Canon Pixma Ip4000	Valencia	Electronics and PC	7	150	0.9	218.97	url_8570
Canon Pixma Ip8500	Alicante	Electronics and PC	1	218	0.858	203.64	url_77823
Canon Pixma Ip8500	Alicante	Electronics and PC	7	190	0.858	203.64	url_77823

Chart properties

X axis: month_month

Filter: market_location

No. QA results: 5

Title: Canon Pixma Ip4000 compa

Fig. 11 Dashboard presented to the user

assigned, so a proper noun is required in the answer, with a semantic preference to the hyponyms of “country” in WordNet. Finally, the following SBs are used in Module 2: “[Iraq] [to invade] [in 1990]”, in order to select the most relevant passages between all the documents. You can notice that the SB “country” is not used in Module 2 because it is not usual to find a country description in the form of “the country of Kuwait”. Module 3 also uses a set of syntactic-semantic answer patterns to search for the correct answer. For example, for the question “What is the brightest star visible in the universe?”, AliQAn extracts “Sirius” from the following sentence: “All stars shine but none do it like Sirius, the brightest star in the night sky”, although a complete matching is not reached between the SBs of the question and those of the sentence.

5.2 Experiment results on the electronic product sales scenario

This experiment is run on the case scenario of Electronic Product Sales that was previously detailed. With regard to the information extracted from the Web, a set of 97,799 Web pages was obtained from the following URLs:

http://www.pccomponentes.com/
http://www.softworld.es/
http://www.dell.es/
http://www.mequedouno.com/

The initial NL request of data is “What is the price of the Canon products in the sales period?”, whose evaluation results are analyzed below for each phase of our proposal:

- Setup phase:

  - The QA integration model and ontology are obtained properly as it is presented in Figs. 2 and 3. We have used the Web Ontology Language (OWL) following W3C Recommendations (Dean and Schreiber 2004; Patel-Schneider et al. 2004). We have used Protégé 4 (ontology editing environment) to create all the ontologies of our proposal (http://protege.stanford.edu/).
  - The DW ontology is obtained similarly using OWL and Protégé 4. Specifically the DW server in our experiment is configured to use the open-source BI platform called Pentaho. Pentaho provides the necessary OLAP capabilities by means of the Mondrian OLAP server. The OLAP server is connected to a MySQL Server 5.6 DBMS that stores the data for the analysis. Since our approach transforms the input into a MDX query, it can be sent directly to the OLAP server, without performing modifications in the platform.
  - With regard to the semi-automatic QA and DW ontology mapping, our evaluation results achieve a 100 % of precision in the detection of equivalent classes and properties in both ontologies for the exact matches (e.g. the "day, month, year" classes). Therefore, we have not detected the necessity of applying techniques to disambiguate word senses, that is to say, situations in which there is a different meaning in spite of the exact matching. In the remaining cases, the precision decreases to 73 % because the mapping is obtained from the lexical-semantic resources (e.g. WordNet). The analyzed errors show that three different situations produce them. In the first kind of errors, the user that confirms the match considers that the automatically assigned class is wrongly mapped. For example, the "market" DW class is automatically mapped to the "location" QA class instead of the "group" QA class, because of the hyperonym relation ambiguity that takes us to decide between "location", "group", and "object". The second one occurs when the user considers that there are several mapping points. For example, the "manufacturer" DW class is automatically mapped to the "group" QA class because of the WordNet hyperonym relation: "occupation – human – group", but it also could be mapped to the "person" QA class because of the "human" WordNet concept. The third error situation comes from problems produced by the wrongly normalization process to obtain the lemma of each class and property, which result in missing matches. The normalization tool should be improved and adapted to the case scenario (e.g. for the "sale_id" DW property). Moreover, we should remark the necessity of a syntactic analysis in order to obtain the head of the phrase. For example, the "sales zone" DW class is automatically mapped to the "location" QA class because our system has chosen the hyperonym relations of the head "zone", instead of those of the modificator "sales" (which would return the "economic" QA class).

- Running phase:

  - The GUI element properly receives the NL request of data through our NL interface, and the Distributor/Integrator distributes the NL request to each DW and QA node. We have evaluated the NL interface through an experiment in which a set of ten users wrote fifty queries per user to evaluate how using query-authoring services improves the overall usability of the system, by enabling early detection of query errors. These users were completely new to the system and they did not have any previous knowledge about the underlying domain. We gave them an initial description of the

DW, without schema representation or concrete entity/property names, and let them query the system in an exploratory way. During this process, users are very likely to introduce mistakes in most of the queries they come up with for the first time. We captured traces for all of these queries and recorded in which stage of the parsing process they were raised. Our results indicate that, from the set of fifty input queries per user, the 89,7 % of them contained errors, from which the 79,9 % of these wrong queries could be detected before they were being executed against the DW. The results of this experiment shown that while an important amount of errors (23 %) are due to lexical errors (usually things like typos), and 26 % of them correspond to syntactic errors (mostly ill-formed sentences in the English language), most of the errors are due to semantic errors (51 %). In order to help minimizing the probability of having lexical errors in a query, the system provides auto-completion for entities and properties, and also auto-correction of typos based on distance-editing algorithms.

- The DW node receives the NL question that is transformed into a MDX. The transformation process is performed in a two-step process. First, the engine determines the entities involved in the question and their correspondence to data warehouse concepts with the aid of the ontology. Then, the engine tests if the resulting question is well formulated and can be translated into a query. After that, if there is no error, the engine translates the set of concepts identified into an MDX query. The ability of our system to answer the different questions posed by the user is dependent on the degree that users' information requirements are covered. In order to ensure that the data warehouse is capable of answering all the desired questions, we design it using a hybrid DW development approach (Mazón and Trujillo 2008). By following this approach, we can trace all the requirements down to the data stored in the data warehouse. We verified that all the requirements posed by the users were covered, thus obtaining a 100 % coverage in the set of questions posed by users. However, it should be noted that any future query unrelated to current information requirements would require an extension of the data warehouse and its associated ontology. Nevertheless, the transformation engine would not require any modifications, as it relies on the DW ontology for the addition of new concepts. Performance wise, we tested the implementation by posing several queries that required extracting the information of over 100.000 entries, and more than 900 products in 10 markets. All the queries posed obtained the result in under 10 s, such as the query presented in section 4.2 that returned a cube with 7 columns and 18,519 rows in 4 s.
- The QA node receives the NL request of data of "What is the price of the Canon products in the sales period?", which is classified by AliQAn as "numerical economic" type. This type means that the possible answer should be of lexical type "number" followed or preceded by a currency symbol (e.g. € or £). The running time depends on the length of the query. In this case, the results are returned in 2 s. With regard to the results obtained on the previously mentioned corpus of 97,799 Web pages, AliQAn obtained a Mean Reciprocal Rank (MRR[9]) of 0.33. In the previous participations of AliQAn in CLEF between 2003 and 2008, there were 11 questions of economic type, where AliQAn obtained a MRR of 0.45. This lower MRR obtained on this corpus is due to a number of reasons. Firstly, the conversion of the Web pages into text should be improved, mainly in the processing of tables in order to link each dimension of the

[9] MRR means the inverse of the rank of the first correct answer. For example, MRR=1 if the first returned document contains the answer for the query, MRR=1/2 if the first returned document that contains a correct answer is in the second position, and so on.

table. Secondly, the AliQAn system has been designed for the CLEF competitions, but it requires a deeper adaptation to the Electronic Product Sales scenario, through the inclusion of domain resources (e.g. an ontology of electronic products), and the adaptation of the patterns to extract an answer in this domain. An excerpt of the results extracted is shown in Fig. 9, in which it is observed a high confidence in each answer (see column 1). This confidence value is higher for the first solution because it completely matches with the question entity "Canon". The remaining solutions present lower confidence values because of the presence of more details of the model Pixma (e.g. "canon pixma 4000/8500"), which does not assure the convenience of the answer.

- The integration of the results is performed by means of the ontological mappings. Thus, errors in the classification of entities or in their representation (i.e. typographical errors, low quality information) translate into rows that are not correctly matched with the information stored in the DW, since no corresponding counterpart is found. While in our experiments the error rate was relatively low, we argue that electronic products domain is a technical one and, thus, the information managed is usually more accurate than in open domains. Performance wise, the integration introduced an overhead in the process since the system has to wait for all the nodes to finish its queries, and then, perform the integration and show the results to the user. The tests show that this delay was not meaningful, and most of the time was spent by I/O in the DW node. Finally, the repository of questions is properly generated from the QA results in a persistent way through a new DW repository.

6 Conclusions and future research

In this paper, we have proposed a full framework with the aim to integrate the internal structured data of an enterprise, with external unstructured data. This framework has been tested on an Electronic Product Sales scenario, in which the enterprise's marketing department wants to analyze sales to identify possible features useful for making new promotions by accessing and acquiring external data from the Web competitors. In this case scenario, the advantages of our proposal have been shown. Specifically, a set of 97,799 Web pages of electronic products have been crawled and accessed by a Question Answering (QA) system on a specific question. This question has been also posed to a DW system with the internal information of the enterprise, and the information returned by both the QA and the DW systems has been presented to the user through a dashboard that helps the decision makers to compare instantaneously internal figures with figures from competitors, thereby allowing taking quick strategic decisions based on richer data. Moreover, the QA results are stored in a persistent way through a new DW repository in order to facilitate comparison of the obtained results with different questions or even the same question with different dates.

Our proposal differs from previous work because we are using a QA system instead of an Information Retrieval (IR), which is more suitable because the information provided by QA is much more structured and can be integrated seamlessly with DW cubes. We consider QA more suitable than Information Extraction (IE) because of the QA flexibility to afford any kind of question, and not only a set of predefined templates. Therefore, the integration is facilitated by the specific information returned by QA and by the ontologies generated from the QA and the DW systems that completes the whole flow of data.

As future work, we plan to prove our framework with new questions and case scenarios, where new QA and DW systems will be integrated in order to check the modularity of our proposal. Moreover, we will study how the different steps of our framework can be better automated, for example, the mapping process between QA and DW ontologies. Another issue to improve in the future is the question analysis in the Distributor/Integrator element, in order to automatically detect the sources to be searched for the required information; and automatically split the question to be passed to each specific node (e.g. when a more complex query is posed such as “What are the price and discount of the Canon products?”, it must be split into two QA questions such as “What is the price of the Canon products?” and “What is the discount of the Canon products?”). A medium-term future work is to adapt this framework to a NOSQL server (e.g. Hadoop) and take advantage from the MapReduce algorithm to process more complex data sources.

Acknowledgments This paper has been partially supported by the MESOLAP (TIN2010-14860), GEODAS-BI (TIN2012-37493-C03-03), LEGOLANG-UAGE (TIN2012-31224) and DIIM2.0 (PROMETEOII/2014/001) projects from the Spanish Ministry of Education and Competitivity. Alejandro Maté is funded by the Generalitat Valenciana under an ACIF grant (ACIF/2010/298).

References

Abelló, A., Ferrarons, J., Romero, O. (2011). Building cubes with MapReduce. In Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP (pp. 17–24).

Badia, A. (2006). Text warehousing: Present and future. In Processing and Managing Complex Data for Decision Support. In J. Darmont and O. Boussaïd, (Eds.) (pp. 96–121). Idea Group Publishing.

Bhide, M., Chakravarthy, V., Gupta, A., Gupta, H., Mohania, M., Puniyani, K., Roy, P., Roy, S., Sengar, V. (2008). Enhanced Business Intelligence using EROCS. In Proceedings of ICDE 2008 (pp. 1616–1619).

Damljanovic, D., Agatonovic, M., Cunningham, H. (2012). FREyA: An interactive way of querying Linked Data using natural language. In The Semantic Web: ESWC 2011 Workshops (pp. 125–138).

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107–113.

Dean, M., & Schreiber, G. (2004). OWL Web Ontology Language Reference. W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-ref-20040210/ (visited on 24th of March, 2013).

Eckerson, W. (2007). Dashboard or scorecard: which should you use? [Online]. Available: http://www.tdan.com/view-articles/4674 (visited on 24th of March, 2013).

Ferrández, A., & Peral, J. (2010). The benefits of the interaction between data warehouses and question answering. EDBT/ICDT Workshops 2010, Article No. 15,(pp. 1–8).

Ferrández, A., Palomar, M., & Moreno, L. (1999). An empirical approach to Spanish anaphora resolution. *Machine Translation, 14*(3/4), 191–216.

Ferrández, S., Roger, S., Ferrández, A., López-Moreno, P. (2006). A New Proposal of Word Sense Disambiguation for Nouns on a Question Answering System. Advances in Natural Language Processing. Research in Computing Science (pp). 83–92.

Ferrández, S., Toral, A., Ferrández, O., Ferrández, A., & Muñoz, R. (2009). Exploiting Wikipedia and EuroWordNet to solve cross–lingual question answering. *Information Sciences, 179*(20), 3473–3488.

Ferré, S. (2012). SQUALL: A Controlled Natural Language for Querying and Updating RDF Graphs. Controlled Natural Language (pp. 11–25).

Gartner Group report. (2011). Gartner Says Solving ‘Big Data’ Challenge Involves More Than Just Managing Volumes of Data. [Online]. Available: http://web.archive.org/web/20110710043533/http://www.gartner.com/it/page.jsp?id=1731916 (visited on 24th of March, 2013).

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pello, F., & Pirahesh, H. (1997). Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery, 1*(1), 29–53.

Henrich, A., & Morgenroth, K. (2003). Supporting Collaborative Software Development by Context-Aware Information Retrieval Facilities. In Proceedings of the DEXA 2003 Workshop on Web Based Collaboration (WBC 2003) (pp. 249–253).

IBM. Business insights workbench. [Online]. Available: http://domino.watson.ibm.com/comm/research.nsf/pages/r.servcomp.innovation2.html (visited on 24th of March, 2013).

Inmon, W. (2005). Building the data warehouse. Ed: Wiley publishing.

Kaufmann, E., Bernstein, A., Zumstein, R. (2006). Querix: A natural language interface to query ontologies based on clarification dialogs. In 5th International Semantic Web Conference (ISWC 2006) (pp. 980–981).

Kerui, C., Wanli, Z., Fengling, H., Yongheng, C., & Ying, W. (2011). Data extraction and annotation based on domain-specific ontology evolution for deep web. *Computer Science and Information Systems, 8*(3), 673–692.

Kimball, R., & Ross, M. (2002). The data warehouse toolkit: the complete guide to dimensional modelling, Ed: Wiley publishing.

LaBrie, R. C., & St. Louis, R. D. (2005). Dynamic hierarchies for business intelligence Information retrieval. *Internationl Journal of Internet and Enterprise Management 2005, 3*(1), 3–23.

Lim, N.R.T., Saint-Dizier, P. Gay, B., Roxas, R.E. (2009). A preliminary study of comparative and evaluative questions for business intelligence. International Symposium on Natural Language Processing, SNLP'09 (pp. 35–41).

Llopis, M., & Ferrández, A. (2012). How to make a natural language interface to query databases accessible to everyone: an example. *Computer Standards & Interfaces*. doi:10.1016/j.csi.2012.09.005.

Llopis, F., Vicedo, J. L., & Ferrández, A. (2003). IR-n system at CLEF-2002. *LNCS, 2785*, 291–300.

Lopez, V., Pasin, M., Motta, E. (2005). Aqualog: An ontology-portable question answering system for the semantic web. The Semantic Web: Research and Applications (pp.135-166).

Losiewicz, P., Oard, D., & Kostoff, R. (2000). Textual data mining to support science and technology management. *Journal of Intelligent Information Systems, 15*(2), 99–119.

Luján-Mora, S., Trujillo, J., & Song, I. (2006). A UML profile for multidimensional modeling in data warehouses. *Data and Knowledge Engineering, 59*(3), 725–769.

Maté, A., Llorens, H., de Gregorio, E. (2012). An Integrated Multidimensional Modeling Approach to Access Big Data in Business Intelligence Platforms. ER'12 Proceedings of the 2012 international conference on Advances in Conceptual Modeling (pp.111-120).

Maté, A., Trujillo, J., Mylopoulos, J. (2012). Conceptualizing and specifying key performance indicators in business strategy models. 31st International Conference on Conceptual Modeling (ER) (pp. 282–291).

Mazón, J. N., & Trujillo, J. (2008). An MDA approach for the development of data warehouses. *Decision Support Systems, 45*(1), 41–58.

Mazón, J. N., Trujillo, J., & Lechtenbörger, J. (2007). Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. *Data and Knowledge Engineering, 63*(3), 725–751.

McCabe, M. C., Lee, J., Chowdhury, A., Grossman, D., Frieder, O. (2000). On the design and evaluation of a multi-dimensional approach to information retrieval. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 363–365).

Patel-Schneider, PF., Hayes, P., Horrocks, I. (2004). OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-semantics-20040210/ (visited on 24th of March, 2013).

Pérez-Martínez, J.M. (2007). Contextualizing a Data Warehouse with Documents. Ph. D. Thesis.

Pérez-Martínez, J. M., Berlanga, R., Aramburu, M. J., & Pedersen, T. B. (2008a). Contextualizing data warehouses with documents. *Decision Support Systems, 45*(1), 77–94.

Pérez-Martínez, J. M., Berlanga, R., Aramburu, M. J., & Pedersen, T. B. (2008b). Integrating data warehouses with web data: a survey. *IEEE Transactions on Knowledge Data Engineering, 20*(7), 940–955.

Pérez-Martínez, J. M., Berlanga, R., & Aramburu, M. J. (2009). A relevance model for a data warehouse contextualized with documents. *Information Processing Management, 45*(3), 356–367.

Priebe, T., & Pernul, G. (2003a). Towards integrative enterprise knowledge portals. In Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM'03) (pp. 216–223).

Priebe, T., & Pernul, G. (2003b). Ontology-based Integration of OLAP and Information Retrieval. In Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03) (pp. 610–614).

QL2. Real-time web data solutions for better business intelligence. [Online]. Available: http://www.ql2.com/ (visited on 24th of March, 2013).

Qu, S., Wang, Q., Liu, K., Zou, Y. (2007). Data Warehouse Design for Chinese Intelligent Question Answering System Based on Data Mining. In Proceedings of the 2nd International Conference on Innovative Computing, Information and Control (ICICIC 2007) (pp. 180–183).

Rahm, E., & Bernstein, P. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal, 10*(4), 334–350.

Rieger, B., Kleber, A., von Maur, E. (2000). Metadatabased Integration of Qualitative and Quantitative Information Resources Approaching Knowledge Management. In Proceedings of the 8th European Conference of Information Systems (pp. 372–378).

Roger, S., Vila, K., Ferrández, A., Pardiño, M., Gómez, J. M., Puchol-Blasco, M., & Peral, J. (2009). Using AliQAn in Monolingual QA@CLEF 2008. *LNCS, 5706*, 333–336.

Roussinov, D., & Robles-Flores, J. A. (2004). Web question answering: technology and business applications. *Proceedings of the Tenth Americas Conference on Information Systems, 3*(1), 46–62.

Santoso, H., Haw, S., & Abdul-Mehdi, Z. T. (2010). Ontology extraction from relational database: concept hierarchy as background knowledge. *Knowledge-Based Systems, 24*(3), 457–464.

Stanojevic, M., & Vraneš, S. (2012). Representation of texts in structured form. *Computer Science and Information Systems, 9*(1), 23–47.

Trujillo, J., & Maté, A. (2012). Business intelligence 2.0: a general overview. *Lecture Notes in Business Information Processing, 96*(1), 98–116.

Wang, C., Xiong, M., Zhou, Q., Yu, Y. (2007). Panto: A portable natural language interface to ontologies. In Proceedings of the 4th European Semantic Web Conference (pp.473-487).