Intelligent Information Management, 2012, 4, ***-***
Published Online September 2012 (http://www.SciRP.org/journal/iim)

BINLI: An Ontology-Based Natural Language Interface for Multidimensional Data Analysis

José Saias[1], Paulo Quaresma[1], Pedro Salgueiro[2], Tiago Santos[2]
[1]Departamento de Informática – ECT, Universidade de Évora, Évora, Portugal
[2]Inductiva, Knowledge Technologies, Lda, Évora, Portugal
Email: jsaias@uevora.pt, pq@uevora.pt, pedro.salgueiro@inductiva.pt, tiago.santos@inductiva.pt

Received ********* 2012

ABSTRACT

Current technology facilitates access to the vast amount of information that is produced every day. Both individuals and companies are active consumers of data from the Web and other sources, and these data guide decision making. Due to the huge volume of data to be processed in a business context, managers rely on decision support systems to facilitate data analysis. OLAP tools are Business Intelligence solutions for multidimensional analysis of data, allowing the user to control the perspective and the degree of detail in each dimension of the analysis. A conventional OLAP system is configured to a set of analysis scenarios associated with multidimensional data cubes in the repository. To handle a more spontaneous query, not supported in these provided scenarios, one must have specialized technical skills in data analytics. This makes it very difficult for average users to be autonomous in analyzing their data, as they will always need the assistance of specialists. This article describes an ontology-based natural language interface whose goal is to simplify and make more flexible and intuitive the interaction between users and OLAP solutions. Instead of programming an MDX query, the user can freely write a question in his own human language. The system interprets this question by combining the requested information elements, and generates an answer from the OLAP repository.

Keywords: NLP; BI; Ontology; Question Answering

1. Introduction

Current technology facilitates access to the vast amount of information that is produced every day. A news article about a company's results can be read anywhere in the world, from the very moment it is made available. The amount of data potentially relevant for any topic, and the lack of time on a process where the information gathered on that topic is vital, lead to the adoption of automated techniques for searching and filtering information.

Both individuals and companies are active consumers of data from the Web and other sources, and these data guide decision making. An investor may decide to buy shares of a company based on the discovery of information favorable to that company. In the activity of a manager, to find that a group of customers shows a pattern that requires intervention, for instance, is a great achievement that depends on the access to relevant information in the shortest time possible. Due to the huge volume of data to be processed in a business context, managers rely on decision support systems to facilitate data analysis.

Online Analytical Processing (OLAP) tools are Business Intelligence (BI) solutions for multidimensional an alysis of data, allowing the user to control the perspective and the degree of detail in each dimension of the analysis. These systems are specialized tools for analysis and visualization of large volumes of business data, usually contained in a Data Warehouse (DW). A conventional OLAP system is configured to a set of analysis scenarios associated with multidimensional data cubes in the repository, such as the total amount of sales per month. To handle a more spontaneous query, not supported in these provided scenarios, one must have specialized technical skills in data analytics. Typically, the construction of such a new analysis scenario would require the implementation of queries in an interrogation language, like MDX[1]. This makes it very difficult for average users to be autonomous in analyzing their data, as they will always need the assistance of specialists. Instead of programming an MDX query, the average user would feel more comfortable asking the system for the information he wishes to see, using his own natural language. To view the amount of sales per quarter, the user would simply write "*What is the total amount of sales per quarter*?", as if he was asking it to a person.

This article describes an ontology-based Natural Language Interface (NLI) called BINLI, whose goal is to simplify and make more flexible and intuitive the interaction

[1]Multidimensional Expressions (MDX) is the most widely supported query language for reporting from multidimensional repositories.

between users and OLAP tools.

In the following section we present some recent publications related to the work we are developing. The model we propose is described in Section 3. In Section 4 we conclude by presenting some considerations about what we have achieved, and enumerating some aspects to consider for future work.

2. Related Work

A DW is "a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process" [1]. DW is used as storage resources for common OLAP systems. A multidimensional OLAP repository has cubes with many dimensions relevant to a particular domain. The dimensions are attributes associated with relevant perspectives for the analysis to be performed, such as a product category, or even the date or location. A dimension can have multiple levels organized into hierarchies, and each level may have one or more members. The date dimension is one of these cases, because it usually has a hierarchy of four levels: Year, Semester, Month and Day [2]. Members are possible values for a level, for example 2010 or 2011 would be members of Year level.

A measurement is a value taken from the intersection of all the dimensions, such as the value of a product *P* sale on a given day *D*, at the store *S*. Quantitative data is stored on a cube fact table. All the rows or measurements in a fact table must be at the same grain (level of detail). OLAP cubes are precalculated, in order to obtain a better query performance [3].

The work in [4] is an OLAP system for analysis and extraction of information on nursing records, starting from the development of a repository of multidimensional data. Authors have used several open source tools developed by Pentaho, a BI software company. That work is, however, a typical implementation of an OLAP solution, since it offers the user a set of fixed analysis scenarios, within which allows common operations such as drill-down, roll-up, slice and dice.

Conventional document retrieval systems are widely used in order to find documents from a set of keywords that describe the user's information needs. But a collection of documents may not be the most interesting type of response to situations that require rapid and specific results. Question Answering (QA) systems allow users to pose natural language questions [5], and instead of returning full documents they provide concise answers.

Kuchmann-Beauger and Aufaure have recently proposed a DW based QA system [6]. Their work is concerned with the semantic analysis of a question, looking for the data model objects that the query depends on, and then producing a data visualization result. They identify keywords or known terms in a natural language query, and link those elements to data model objects, which correspond to the OLAP repository dimensions, measures or other schema objects. Using a set of business sales and orders questions, the keyword direct match approach is not always sufficient. When no answer has been found, they rewrite the user question using a thesaurus-based transformation, by applying synonym expansion. This work also infers semantic closeness between terms based on web search results, for unknown words in the question.

While many studies start with some unstructured natural language elements and look for a structured result (such as a query), other works go in the opposite direction.

Ioannidis work [7] aims at translating structured data into natural language. The author concludes that text generation to produce a natural result is far from trivial, as identifying the right linguistic constructs is a complex task.

Conversation-based systems work with natural language, such as QA systems, but have the characteristic of generating a dialogue with the user, in order to resolve ambiguity in the interpretation of the question, or simply to collect his feedback on a given result. An example of conversation-based natural language interface to relational databases is described in [8]. Knowledge trees are used to structure the domain knowledge and to direct the conversational agent towards the goal of database query generation as required by natural language input.

Frost and Fortier proposed a denotational semantics model for natural language database queries [9], with explicit semantics for transitive verbs and negation. The GINLIDB system is a generic interactive NLI for databases [10] meant to facilitate the interaction with databases for common people who are not familiar with SQL syntax. It has two major components: the linguistic handling component and the SQL constructing component. That system accepts English language requests, which are interpreted and translated into SQL commands using a semantic-grammar technique and a knowledge base with the database schema.

In 2007, Li *i.e.* proposed an XML database interactive NLI [11] supporting aggregation, nesting, and value joins. English sentences are translated into XQuery expressions, by mapping grammatical proximity of natural language parsed tokens with the corresponding elements in the XML data to be retrieved.

3. Proposed Approach

The main idea is that an ordinary person, with no background in interrogation languages or programming, can interact with an OLAP system and perform queries in a spontaneous way. These queries are written in natural language, currently in Portuguese but the system is designed to be multilingual. Queries may be related to any subject or concept in the multidimensional repository, about which the user wants some information, and they

are not pre-computed. BINLI receives a question, freely written by the user, interprets it, and generates an answer from the multidimensional repository. We start by presenting the modules of the system and then we describe the methodology for processing the questions, presenting some illustrating examples.

Figure 1 shows the main components of the BINLI system and how they are interconnected. The user has a Web interface to enter his questions and to receive the system results. Question Analyzer is the most important component of the BINLI system. It applies language specific tools according to the idiom used in the inserted question. For now we consider Portuguese questions, but other languages such as English, Spanish or French can be supported as well, by integrating their respective dictionaries and parsers.

In this part BINLI behaves as a QA system with a defined domain: the DW content, the data in the OLAP engine multidimensional repository. It must determine what are the foci of interest for the user, which information is sought on them, and still detect possible restrictions to consider, as occurs in typical QA question analysis phase [5,12-14].

BINLI applies common procedures in NLP, *i.e.*, parsing and Named Entity Recognition (NER). The question text is parsed for morpho-syntactic analysis, in order to determine the main verb and the structure with the dependency relationships between the query terms. Some rules are also applied to handle special situations, detected via superficial text pattern analysis. After NLP techniques and the application of some rules, the system seeks a mapping between core terms in user question and OLAP schema objects. This is achieved by semantic similarity computation and semantic reasoning [15,16]. In the end of this phase, if a word or an expression admits several meanings, all these semantic paths are considered, resulting in a list of several possible interpretations for the question. There is a weight associated with each meaning of a query term, which is used as a sorting criterion when considering various interpretations.

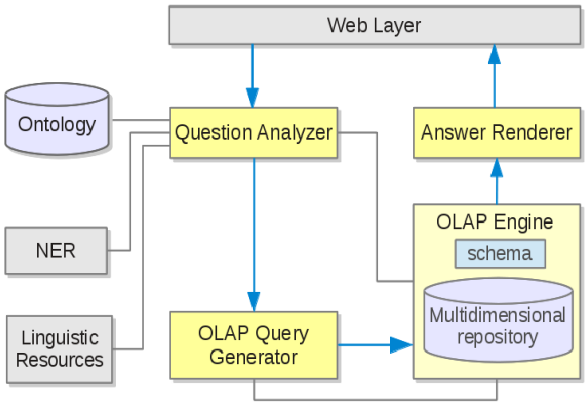

Figure 1. System Architecture.

The OLAP Query Generator is the module that will generate an MDX query for each question interpretation, according to the OLAP repository schema. If there are multiple interpretations, they are sorted in descending order of weight, and therefore the former is the more plausible. The generated OLAP query code for the first interpretation is then passed to the OLAP engine for execution. For this purpose, we chose the Pentaho Analysis Services Community Edition, also known as Mondrian[2], an OLAP server. Finally, the Answer Renderer component shows the result in the form of tables and charts using JPivot[3] or similar tools. To find the elements to display in response, the system searches for objects in the repository schema that is relevant to the query. The first approach is done by seeking expressions in the query text that are a direct match with OLAP schema objects. This correspondence is attempted only for query terms that the former step of the analysis has marked as possible links to the repository. If the match is successful, these are elements which will be sought on some properties or facts, in some repository table.

In addition to the direct match, the system also tries an indirect concordance between candidate terms and schema objects. Errors by exchange of two letters are common, while writing a query. When a candidate term, such as *sotre*, has no match in the scheme but the word *store* would have a match, the system considers this possible interpretation. We use the Levenshtein distance to determine if the similarity between the terms is acceptable.

The other approach for indirect relations is the test of semantic compatibility between the terms. This is where the support ontology plays an important role, allowing the analysis of the semantic relations *SynonymOf*, *MeronymOf*, *HyperonymOf*, *InstanceOf* and *AkA*. The ontogy includes the terminology from repository multidimensional cubes. In addition to the base name associated with each object in the DW schema, there is one or more alternative designations for each supported language. When applied to members in the OLAP schema, this technique has a similar effect to the application of semantic query expansion, in Information Retrieval, but here the scope is broadened with the ontology content, that may evolve.

When establishing the correspondence between query terms and OLAP objects, the weight on a direct match (100) is greater than the weight for indirect cases of Levenshtein error correction (80) and the ontology based semantic compatibility (90 for *AkA* relation; 85 for *SynonymOf* relation; 60 for others). The weight of a query interpretation is the average of the weights given to the interpretation of its terms. This allows interpretation ranking.

[2]http://mondrian.pentaho.
[3]http://jpivot.sourceforge.net

Consider the following question, written in Portuguese:"*Quais são as vendas de produtos por loja*?" (or "*What are product sales per store*?" in English). BINLI Question Analyzer will perform a linguistic analysis to the question text, discard the stop words, and it selects for candidate terms, to connect to the repository scheme, the words *produtos* (*product*), *vendas* (*sales*) and *loja* (*store*).

The module responsible for the generation of an OLAP query is then activated and produces the MDX code shown in **Figure 2**. In this simple example there is a direct link between query terms and objects in the repository schema. These OLAP schema object names appear underlined and highlighted with red color in **Figure 2**. They are an exact match of the relevant words we saw in the question.

Figure 3 shows the result that the system returns to the client through the Web interface. The Answer Renderer fetches the execution result from the OLAP engine and then generates the visual representation. The chart type is set according to the number of dimensions to display.

Let's take another example: "*Qual é o valor das vendas de mesas no mês passado por distirto*?" (in English: "*What is the value for table sales in last month per district*?"). In this case there is a spelling mistake in the word *distirto*. The existence of a valid term at a very short Levenshtein distance allows automatic correction of that expression into *distrito*.

The question refers to amounts for sales on *mesas* (*table products*). There is no direct correspondence between the term *mesas* and an object in the repository. However, the ontology includes this business domain terminology, and has a hint for the system: the term *mesa* is hyperonym of "*mesa de centro*" and "*mesa de cozinha*". As *msas* is the plural of *mesa*, the system performs an interpretation of that question term involving some concepts in question OLAP repository.

The text also includes the expression "*last month*". Such allusions are time constraints that the system has to solve. The user context includes the notion of location and time. In this case, the system calculates which is the month prior to the time of the question. **Figure 4** shows the result discriminating table sales values for each district. The bars omitted in the second group means the absence of values for those districts and for the desired month.

```
SELECT
NON EMPTY{ DISTINCT( HIERARCHIZE( {{
    [Store].[Name].Members}}))} ON COLUMNS,
NON EMPTY{ DISTINCT( HIERARCHIZE( CROSSJOIN({{
    [Product].CHILDREN}},{{
    [Measures].[TotalGross]}})))} ON ROWS
FROM [Sales]
```

Figure 2. MDX query for "*What are product sales by store*?"

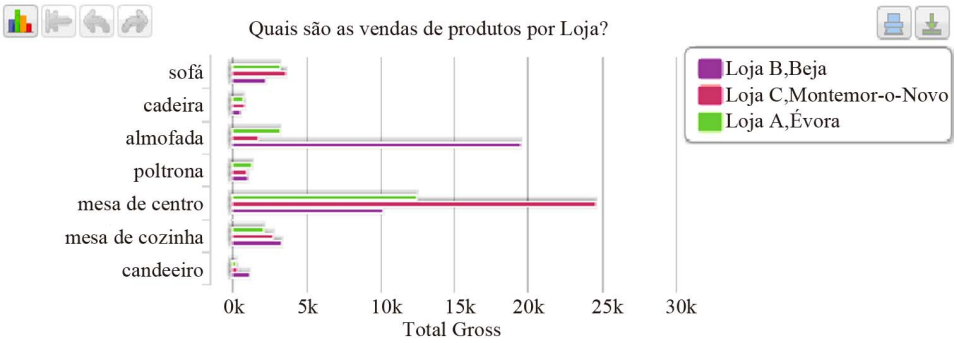

Figure 3. Result for "*What are product sales by store*?"

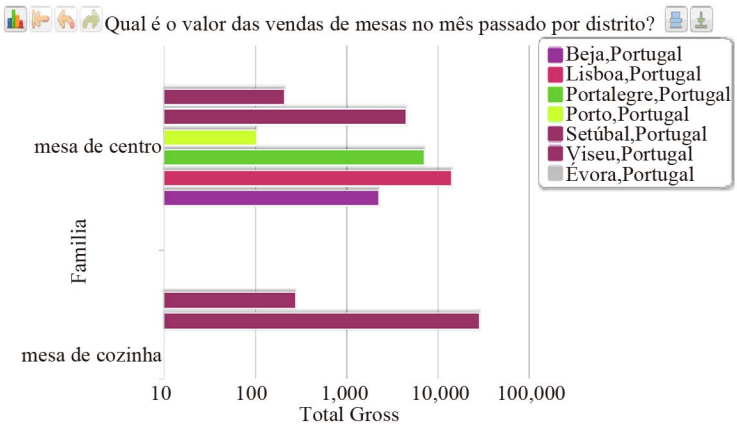

Figure 4. Result for "*sales in last month per the district*".

4. Conclusions and Future Work

In this article we presented BINLI, an ontology-based Natural Language Interface for multidimensional data analysis. The language currently supported by this system is Portuguese, but any of the most spoken western languages can be supported in the near future. The model has a generic architecture and is thought to be multilingual, and may also operate in cross-lingual mode with future improvements.

We developed a prototype system that is currently being evaluated by people in the management sector and independent from the development process. Our experiments show that the quality of the ontology is crucial to achieve a correct interpretation. The current phase of testing aims to feed the ontology with semantic elements that are gradually found, and that are usefull for the question understanding process.

An OLAP cube can have many dimensions, each having several hierarchies which describe the data under diverse views. There may be cubes with homonymous dimensions or hierarchies. This can be quite challenging when matching the words from the NL query to the terms found in the repository structure, producing many alternative interpretations. As in the work described in [6], our system produces a list of possible interpretations for a question. But instead of a thesaurus, the system described in this article has an ontology as the support knowledge base, which allows useful inference in processes of considerable semantic complexity.

The sorting criterion applied to the query interpretations needs to be revised. In addition to the interpretation weight formula, we may consider other weight values in individual term correspondence process depending on the semantic relationship. In this initial phase of the project, we focused on embracing the OLAP schema and domain terminology. We tried to minimize the cases where there was no response. To assess the impact of other interpretation weight formulas, we need further testing using queries chosen by BI experts and for which there are several plausible interpretations.

Another feature we need to improve is the presentation of alternative responses. When there are several possible interpretations for a query, it should be easy for the user to navigate through the results associated with each of these interpretations. We try to automatically provide the most plausible result for the user's need. Nevertheless, it is interesting to see alternative interpretations.

In this line of improvement, the user can select the interpretation that has to do with his interest when he asked the question. The system will analyze patterns of preference for that user and afterwards choose as first result a more appropriate interpretation, for that question category.

In order to improve the interpretation capability, the system can become interactive and ask the user for hints about eventual query terms that might not be automatically understood. The same can be done to reduce ambiguity. If a term *T* can be the name of a company but also the name of a city, the system can prompt the user to clarify the meaning of *T* before proceeding to calculate the result, rather than choosing automatically one of those interpretations.

Another important direction for future work is Text OLAP [17]: the loading of data from unstructured sources into the OLAP repository. A significant part of business data are unstructured or semi-structured documents, to which we can apply information extraction and NLP techniques. New data, obtained by the new approaches, may lead to the discovery of relevant information for the activity of the OLAP/BI user.

It is important to note that the system presented here is not intended to fully replace the mouse-based interfaces, with fixed and specific click or *drag & drop* operations. Instead, the goal is to complement these conventional interfaces with the introduction of a natural interface to provide a simple and flexible solution for non-expert users willing to use a BI system autonomously.

5. Acknowledgements

This research is partially supported by the QREN/PO Alentejo program, under the project ALENT-07-0202-FEDER-018599.

REFERENCES

[1] W. H. Inmon, "Building the Data Warehouse," QED Technical Publishing Group, Wellesley, 1992.

[2] R. Bouman and J. van Dongen, "Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL," Wiley Publishing, Inc., Hoboken, 2009.

[3] R. Kimball and M.Ross, "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling," 2nd Edition, Wiley Computer Publishing, New York, 2002.

[4] J. Silva and J. Saias, "Olap em âmbito hospitalar: Transformação de dados de enfermagem para análise multidimensional," In Actas das 2ªs Jornadas de Informática da Universidade de Évora - JIUE'2011, Évora, 2011, pp. 77-85.

[5] C. Monz, "From Document Retrieval to Question Answering," Ph.D. Thesis, University of Amsterdam, Amsterdam, 2003.

[6] N. Kuchmann-Beauger and M.-A. Aufaure, "A Natural Language Interface for Data Warehouse Question Answering," 16*th International Conference on Applications of Natural Language to Information Systems—NLDB* 2011, Alicante, 28-30 June 2011, pp. 201-208.

[7] Y. E. Ioannidis, "From Databases to Natural Language: The Unusual Direction," *Conference on Applications of Natural Language to Information Systems*, London, 24-27

June 2008, pp. 12-16.

[8] M. Owda, Z. Bandar and K. A. Crockett, "Conversation-Based Natural Language Interface to Relational Databases," *Proceedings of the* 2007 *IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology Workshops*, Silicon Valley, 5-12 November 2007, pp. 363-367.

[9] R. A. Frost and R. J. Fortier, "An Efficient Denotational Semantics for Natural Language Database Queries," 12*th International Conference on Applications of Natural Language to Information Systems*, Paris, 27-29 June 2007, pp. 12-24.

[10] F. A. El-Mouadib, Z. S. Zubi, A. A. Almagrous and I. El-Feghi, "Interactive Natural Language Interface," *WSEAS Transactions on Computers*, Vol. 8, No. 4, 2009, pp. 661-680.

[11] Y. Y. Li, H. H. Yang and H. V. Jagadish, "Nalix: A Generic Natural Language Search Environment for XML Data," *ACM Transactions on Database Systems*, Vol. 32, No. 4, 2007.

[12] H. Q. Hu, "A Study on Question Answering System Using Integrated Retrieval Method," Ph.D. Thesis, The University of Tokushima, Tokushima, 2006.

[13] C. Amaral, A. Cassan, H. Figueira, A. Martins, A. Mendes, P. Mendes, J. Pina and C. Pinto, "Priberam's Question Answering System in qa@clef 2008," *CLEF Workshop*, 2008, pp. 337-344.

[14] J. Saias and P. Quaresma, "The Senso Question Answering System at qa@clef 2008," Technical Report, CLEF Workshop, 2008.

[15] R. Thollot, N. Kuchmann-Beauger and M.-A. Aufaure, "Semantics and Usage Statistics for Multi-Dimensional Query Expansion," 17*th International Conference of Database Systems for Advanced Applications*, Busan, 15-18 April 2012, pp. 250-260.

[16] J. Saias and P. Quaresma, "Semantic Networks and Spreading Activation Process for QA Improvement on Text Answers," *Proceedings of the* 8*th Brazilian Symposium in Information and Human Language Technology—STIL*2011, Cuiabá, 24-26 October 2011.

[17] B.-K. Park and I.-Y. Song, "Incorporating Text Olap in Business Intelligence," *Business Intelligence Applications and the Web: Models, Systems and Technologies, IGI Global*, 2012, pp. 77-101.

