Object-Relational Database Based Category Data Model for Natural Language Interface to Database

Avinash J. Agrawal

Shri Ramdeobaba Kamla Nehru Engineering College
Nagpur-440013 (INDIA) Ph. No. 91+ 9422830245
avinashgondia@gmail.com

Dr. O. G. Kakde

Visvesvaraya National Institute of Technology
Nagpur(INDIA)
ogkakde@vnit.ac.in

Abstract

Domain specific question answering technique allows users to use natural language to express their queries so that users need not have the knowledge about the structures of the information source. For such an application relational model is not suitable as it is not a natural way to represent real world knowledge. Using relational model for representing information source results in scattered relations of data about the real world objects. In this paper an effective category model is presented to organize information according to their contents based on object-relational database. For the discussion of the category model railway domain is used in the paper.

Keywords

Natural Language Processing, Question Answering, Object-Relational Database Model, Natural Language Question Patterns.

1. Introduction

Domain specific Question Answering Techniques [1] of natural language processing makes the information access easy from the remote data source. In this method sources of information can be divided into application domains. For each domain a system is designed separately which will map unstructured queries with actual database query. After firing this query, system will get the desired information from the source. This information is then passed to the user in an appropriate format. Railway inquiry is one the appropriate application domain for such system [1]. In Railways Inquiry often the people wants to get short information in quick time. Here searching of such short information on web is not efficient and a common user does not have knowledge of actual database query. In such a scenario Question Answering that answers user query without any human intervention can prove to be very useful [2]. By using such system answers for repeated question can be provided accurately and quickly. This system can be implemented for other application domain as well where a common person from a complex information source requires frequent access.

However the development of such system is a challenging task since a long time. The main reasons are portability [3] that is caused by the limitation of domain knowledge representation and accuracy [4] that is caused by too much dependency on domain knowledge rather than

DOI : 10.5121/ijaia.2011.2103

language knowledge. In this paper an effective conceptual model to organize data according to their contents based on object-relational databases is presented and also Natural language interface to database is described.

2. Relational vs. ORDB in NLIDB

Although database query language, such as SQL, is very powerful for data access, real database users (end user) do not know how to use them. Almost every database application exploits some kind of interactive interfaces. However the development of user interfaces is still ad-hoc [6] and often uses some predefined form-based style. This greatly limits what users can do with the data in databases. How can users express their queries in a semantic way is an important research topic. A lot of research on question answering kind of user interface has been done based on relational model [5]. However the relational model is not a natural way to represent real world knowledge. Using relational model for representing information source results in scattered relations of data about the real world objects due to the complex process of normalization. Object-Relational Database [7] is an emerging concept and have been considered as the next great wave in the database community due to its naturalness. ORDB preserves the semantic of data stored which helps in drawing conclusion from incomplete input information. Also this method takes all advantages of object orientation like reusability, extensibility, portability etc. However presently the applications are difficult to develop in ORDB , as there is not a well-defined conceptual model for such kind of database design that plays the same role as ER or EER model [8] for the relational model. To solve these problems a category model along with Natural language interface to database has been proposed which is explained in next sections. Also how this model is useful in mapping users query with database query is explained.

3. Category Model

Entity relationship(ER) model is widely used as conceptual model for relational data which describes entities involved and relationship between them. Similarly to model object relational data category model can be used. The three important constituents of category model are Objects, Category Hierarchy and Relationships.

3.1. Objects and Attribute

In the category model, an object represents the real world physical and conceptual entities. For example in railway domain, a specific Train or Station is an example of an object. Objects have attributes through which they are related to each other. An object attribute is a named property that describes a value held by the object. There are four kinds of attribute values in the category model:

1. Atomic values such as strings (‘Mumbai Howrah Mail’, ’Delhi’ ,………..), integer (10,12, ……..), real (61.2, 100.1, ……).

2. Tuple values such as Seats Available (AC2-60, AC3-120,SL-780), Fare (AC2, 1000), Stoppage (Superfast, {Mumbai, Nashik, Nagpur}) .

3. Object identifier (OID). OID is the property of an object, which distinguishes it from all others and is used to references the objects. OID and objects have the immutable relationship. In other words, one OID represents one object and one object has one OID.

4. Set values such as { }, { Mumbai , Nashik , Nagpur }, { Stoppage (Superfast, { Mumbai, Nashik, Nagpur }) }.

The variety of attribute values allows objects in the category model to naturally simulate the properties of real world entities.

3.2. Category Hierarchy

Category in the category model represents a collection of objects with common attributes. Categories are identified naturally. For example, Train and Station are categories. There is an Instance-of relationship between objects and categories. Figure 1 illustrates the relationship between categories in the upper layer and objects in the lower layer.

A category may have the hierarchical structures. That means one category can have subcategories or super-categories when the groups of objects they denote are subsets or superset of the corresponding groups. The super-category holds common attributes; the sub-categories inherit the attributes of their super-category and introduce additional attributes.

Figure 1 also shows a category hierarchy in the upper layer where category Train has three sub-categories Rajdhani, Express and Passenger. Category Express has sub-categories Super Fast and Non Super.

Besides inheritance, our category model also supports polymorphism as in object-oriented data models [9]. Polymorphism is critical because it allows categories derived from a super-category to be used where the super-category is expected. For example, attribute Seasonal train member of the category Seasonal trains needs to reference the objects in the categories Train, Rajdhani, Express, Passenger, Super fast and Non super. With polymorphism, we can simply make attribute Seasonal train refer to the category Train that is the super-category of the categories above. Thus, any object related to Train's sub-categories could be referenced. Without polymorphism, it would be difficult to specify this.

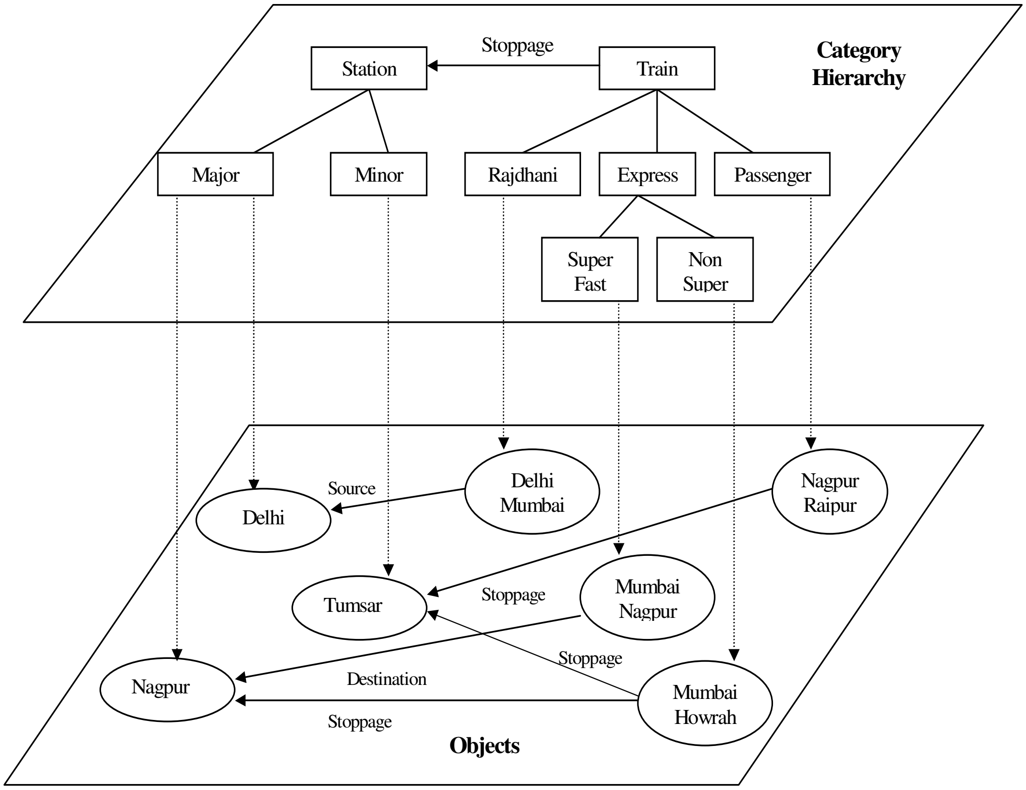

Figure1. Example Category Hierarchy and Object Relationships

3.3.Relationship in Category Model

Unlike the ER model that represents relationship by relationship types, in the category model relationships are represented by category attributes, In particular, if there are two categories C1 to C2 with relationship R1, relationship R1 has two traversal paths, that is, the path from C1 to C2 and the path from C2 to C1. To represent relationship R1, we define a pair of attributes Ar1 and Ar2 on categories C1 and C2 respectively. Ar1 references the objects in category C2 by the OID type, and Ar2 presents the other direction of the traversal path of the relationship R1. We use attribute with OID or set of OID types to represent the relationships between two categories: an OID type represents the one to one relationships and a set of OID types represents the one to many or many to many relationships.

In the category model, there can be categories whose existence depends on others categories. For example, category Station is related to City by means of a dependency relationship. We call Category City a parent category and Station a dependent category. It is important to note that the dependency relationship we present herein is not equal to the Aggregation or Part-off relationship [10] in the object data model. Some aggregation relationships are dependency relationships and some are not. Dependency relationships are the key to semantic integrity checking. In other words, dependency relationships allow tracking and solving inconsistencies in the category model.

4. The 3 Layer Architecture of NLIDB and Category model

Generally, there are three levels of schemas involved in a Natural Language Interface to Database, which are the user's linguistic schema, the conceptual schema and the actual data schema [11]. The task of an NLIDB is to map the user's linguistic schema to the actual data schema. However the distance between them is very far, because the actual data schema does not contain any semantic information and the linguistic schema is flexible and varied. An intermediate representation, the conceptual model is introduced in most systems. The linguistic schema is first mapped to the unambiguous conceptual schema and then makes the actual data schema be defined from the conceptual schema.

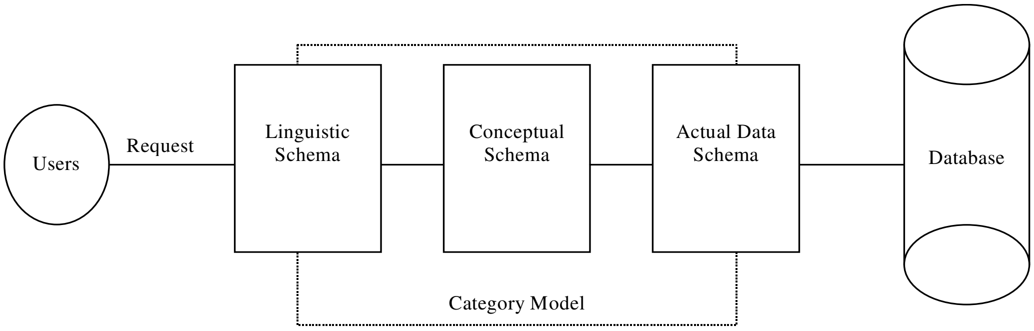

Figure.2 Three Layer Architecture of NLIDB

Figure 2 illustrates how these schemas relate to each other. Although the ER model is also available to express a conceptual schema, it is developed for database design and cannot express complete semantics. The category data model not only has semantics to cover linguistic schema but also can be easily mapped to the object-relational data schema. So, it is more powerful than the ER model in term of semantic expression.

5. Mapping Queries

The concept of category and category relationship are fundamental in the category model. Utilizing these concepts, we can build several natural language patterns [12]. Since the category model considers real world applications as a set of categories, the function of these patterns is to query objects in the category. The natural language patterns can be abstracted as:

[Selected Attribute] + Topic Category + Conditions.

The topic category refers to the theme of the query. For example, the query “Find the time, seat availability of super fast Express at Delhi that runs from Delhi to Mumbai” has the topic category Express train. We call the specific attributes of a topic category as selected attributes. For example, time and Seat availability are selected attributes. The selected attributes are optional elements. When they are omitted the query includes all attribute of the topic category. Conditions refer to the elements in the query modifying a topic category. For example, Delhi Station and Delhi to Mumbai are conditions of the topic category Express train in the considered query. We call Station a Condition Category, since it is a category modifying a topic category. Condition categories are connected by prepositions such as in and at. We call this preposition phrase a category phrase. The implicit relation among these conditions is and. The conditions for the topic category come from three aspects:

i) The values of the condition categories. For example, in the query above, “Delhi” is the value of the condition category Station.

ii) The Value of the attributes of the topic category. For example, “Delhi to Mumbai” is the value of attribute Source and Destination.

iii) The value of the topic category. Consider the query “Train from Delhi to Mumbai”, the value “Delhi” and “Mumbai” are the condition of the topic category Train.

The query language used in object-relational database is the ORSQL that is a dialect of SQL-3. The SQL statements have the familiar structure of SELECT-FROM – WHERE - GROUP BY – HAVING - SORT BY and also have the features of supporting inheritance, reference, dereference, nested-table etc. There are three clauses that are fundamental to construct ORSQL queries:

SELECT CLAUSE, FROM CLAUSE and WHERE CLAUSE.

The translation of the selected attributes to the attributes of a table can be handled by the alias match approach. However, getting attributes’ name is not the sufficient condition to obtain the SELECT clause, because the attributes’ representation in ORSQL queries does not only depend on the attributes’ name. It also depends on the attribute types. The representation of different types in ORSQL is different.

The FROM CLAUSE in ORSQL may contain several related table joins. Actually, this feature of the FROM CLAUSE raises one of the most difficult and most fundamental problems in a natural language interface [13].

The WHERE clause refers to the conditions of the query. After getting the three clauses, the final query can be generated by the formulation: SELECT CLAUSE + FROM CLAUSE + WHERE CLAUSE.

6. Related Work

Natural language interface to database is not a new topic, research is going on since a long time many systems are suggested and still it is an open problem. NLIDB System developed in recent past are:

6.1. PRECISE

PRECISE [14] is a system developed at the University of Washington in 2005. It translates English questions into SQL query. It introduces the idea of semantically tractable sentences which are sentences that can be translated to a unique semantic interpretation. It analyzes given inputs by implementing a graph matching approach (Maxflow algorithm) resulting in possible mappings. While it is able to achieve high accuracy in semantically tractable questions, the system compensates for the gain in accuracy at the cost of recall. It adopts a heuristic based approach, the system suffers from the problem of handling nested structures.

6.2. WASP

WASP [15] is developed at the University of Texas in 2006. It requires no prior-knowledge of the syntax, because the whole learning process is done using statistical machine translation techniques. The system is based solely on the analysis of a sentence and its possible query translation, and the database part is therefore left untouched. There is a lot of information that can be extracted from a database, such as the lexical notation, the structure, and the relations within. Not using this knowledge prevents WASP to achieve better performances. The second problem is that the system requires a large amount of annotated corpora before it can be used, and building such corpora requires a large amount of work.

Almost all the systems developed so far suffer from low success rate because of low usage of language knowledge. Also these systems are too much dependent on database which affects portability. All these systems used relational database and so far no system is proposed using ORDB as information source.

7. Conclusion

For question answering using natural language to get information from database for realistic application like railways inquiry, some natural way of representing source information is required. In this paper, the category model that allows modeling information in more natural ways has been described. Also the paper discuses about how this data modeling approach supports the translation of a general natural language query pattern into its equivalent database query.

In future detailed analysis of complex natural language query patterns specifically related to the target domain is expected. And also the techniques to resolve the issues in generating database queries for these complex patterns using category models needs to be addressed.

8. REFERENCES

[1] A. J. Agrawal, "Using Domain Specific Question Answering Technique for Automatic Railways Inquiry on Mobile Phone", In Proceeding of the International Conference on Information Technology : New Generation -08(ITNG-08), Las Vegas, USA, 7-9 April 2008.

[2] A. J. Agrawal, M. B. Chandak, "Mobile Interface for Domain Specific Machine Translation Using Short Messaging Service". In Proceeding of the International conference ITNG-07, Page 957, Las Vagas, USA, April 2007.

[3] X. Meng and S. Wang, "NChiql: The Chinese Natural Language Interface to Databases", Proceeding of 12th International conference , DEXA-2001, Munich, Germany, 3-5 September 2001.

[4] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates, "Modern Natural Language Interfaces to Databases : Composing Statistical Parsing with Semantic Tractability", 20th international conference on Computational Linguistics COLING-2004,Geneva, Switzerland, 2004, 141.

[5] Minock, Michel, “Where are the killer application of restricted domain question answering”, In proceeding of the IJCAI Workshop on Knowledge Reasoning in Question Answering, Edinbergh, Scotland, page 4, 2005.

[6] T. Griffiths, J. McKirdy, G. Forrester, N. W. Paton, J. B. Kennedy, Peter J. Barclay, R. Cooper, A. Carole and P.D. Gray, “Exploiting Model-based Techniques for User Interfaces to Database”, VDB, 1998, 21-46.

[7] W. Kim, “ Object – Oriented Database : Definition and Research Direction”, IEEE Transactions on Knowledge and Data Engineering, 1990, 2(3):317-341.

[8] P. Chen, “The Entity Relationship Model: Toward a Unified View of Data”, ACM Transaction on DB System, 1976, 1(1):9-36.

[9] G. Gottlob, M. Schreff and B. Rock, “Extending Object-Oriented Systems with Roles”, ACM Transactions on Information Systems, 1996, 14(3):268-296.

[10] A. Olive, D. Costal and M. Sancho, ”Entity Evolution in ISA Hierarchies”, Proceeding of 18th Internationl Conference on Conceptual Modeling, Paris, France, 15-18 November 1999, 62-80.

[11] W.W. Chu and F. Meng, “Databse Query Formation from Natural Language using Semantic Modeling and Statistical Keyword Meaning Disambiguation”, Technical Report 990003, 1999, 16.

[12] W. S. Luk., “Building Natural Language Interface to an ER Database”, Proceeding of the Eighth International Conference on Entity-Relationship Approach, Toronto, Canada, 18-20 October 1989, 345-360.

[13] E. Brill, “Pattern Based Disambiguation for Natural Language Processing”, Proceedings of the 19th International Conference on very Large Data Bases, Dublin, Ireland, 1993, pages 39-51.

[14] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates, “Modern Natural Language Interfaces to Databases : Composing Statistical Parsing with Semantic Tractability”, COLING (2004)

[15] Yuk Wah Wong, “Learning for Semantic Parsing Using Statistical Machine Translation Techniques”, Technical Report UT-AI-05-323, University of Texas at Austin, Artificial Intelligence Lab, Oct 2005.

Avinash J. Agrawal received Bachelor of Engineering Degree in Computer Technology from Nagpur University, India and Master of Technology degree in Computer Technology from National Institute of Technology, Raipur, India in 1998 and 2005 respectively. He is currently pursuing Ph.D. from Visvesvaraya National Institute of Technology, Nagpur. His research area is Natural Language Processing and Databases. He is having 12 years of teaching experience. Presently he is Assistant Professor in Shri Ramdeobaba Kamla Nehru Engineering College, Nagpur. He is the author of seven research papers in International and National Journal, Conferences.

Dr. O. G. Kakde received Bachelor of Engineering degree in Electronics and Power Engineering from Visvesvaraya National Institute of Technology (formerly Visvesvaraya Regional College of Engineering), Nagpur, India and Master of Technology degree in Computer Science and Engineering from Indian Institute of Technology, Mumbai, India in 1986 and 1989 respectively. He received Ph.D. from Visvesvaraya National Institute of Technology, Nagpur, India in 2004. His research interest includes theory of computer science, language processor, image processing, and genetic algorithms. He is having over 22 years of teaching and research experience. Presently he is Professor and Dean, Research and Development at Visvesvaraya National Institute of Technology, Nagpur, India. He is the author or co-author of more than thirty scientific publications in international journals, international conferences, and national conferences. He also authored five books on data structures, theory of computer science, and compilers. He is the life member of Institution of Engineers, India. He also worked as the reviewer for international and national journals, international conferences, and national conferences and seminars.