Towards a Bootstrapping NLIDB System

Catalina Hallett and David Hardcastle

The Open University, Walton Hall
Milton Keynes, MK6 7AA, UK

Abstract. This paper presents the results of a feasibility study for a bootstrapping natural language database query interface which uses natural language generation (NLG) technology to address the interpretation problem faced by existing NLIDB systems. In particular we assess the feasibility of automatically acquiring the requisite semantic and linguistic resources for the NLG component using the database metadata and data content, a domain-specific ontology and a corpus of associated text documents, such as end-user manuals, for example.

1 Introduction

This paper presents the results of a feasibility study for bootstrapping a natural language database query interface which uses natural language generation (NLG) technology to address the interpretation problem faced by existing NLIDB systems. The query system presents the user with an interactive natural language text which can be extended and amended using context sensitive menus driven by Conceptual Authoring. Using NLG to allow the user to develop the query ensures accuracy and clarity. When the user submits the query the semantic representation is transformed into a valid SQL statement. A detailed discussion of the technology and an evaluation which showed the system to be a reliable and effective means for domain experts to pose complex queries to a relational database is presented by Hallett et al. [1].

While this approach delivers clear benefits, they come at a cost; domain expertise is required to construct the semantic resources, linguistic expertise is required to map this domain knowledge onto the language resources, and knowledge of the database structure is needed to map it onto a valid query structure. Our proposed solution is for the system to infer the resources and mappings required from a domain ontology, the database metadata and data content and a corpus of domain-specific texts. The feasibility study reported in this paper demonstrated that we can, in principle, infer the required resources from a simple, highly normalised database with well-formed lexical descriptors such as MS Northwind or Petstore. However, it also highlighted the need to couple metadata mining with analysis of external sources of information.

1.1 Related Work

Providing user-friendly query interfaces for casual and non-specialist users, which alleviate the need for programmatic knowledge is a central problem for the data

E. Kapetanios, V. Sugumaran, M. Spiliopoulou (Eds.): NLDB 2008, LNCS 5039, pp. 199–204, 2008.

querying community. Whether these interfaces are form-based, visual, or natural language-based, knowledge about the data source structure and content is essential to the construction of intuitive interfaces. Traditionally, natural language interfaces to databases (henceforth, NLIDB) work in two steps:

– query interpretation: a natural language query entered by the user is parsed into a logical representation
– query translation: the logical representation of a query is mapped to a database querying language

It is evident that the query interpretation process requires both extensive linguistic resources for understanding the query, whilst the query translation step requires semantic resources for mapping query terms to database entities. In early NLIDBs, these resources (such as semantic grammars and lexicons) were created through an extensive manual process, resulting in heavily database-dependent systems.

The issue of interface portability was first highlighted in the early 1980's, and the fact that database schemas could be used to acquire domain knowledge has been exploited in systems such as CO-OP [2] and INTELLECT [3]. These systems also demonstrated that a modular architecture might allow query interfaces to be ported without code changes. Although these systems made some use of the database schema to map query terms to database entities, porting the interface to a new database still required extensive reworking of the lexicon, although the introduction of generic linguistic front-ends, in which the query interpretation stage is independent of the underlying database (see [4]), reduced the impact. Current NLIDB systems employ a variety of machine learning techniques in order to infer semantic parsers automatically [5,6] or to improve syntactic parsers with domain specific information [7]. However, these systems require large sets of annotated SQL queries for training purposes. The PRECISE system [8] employs a semantic model in order to correct and enhance the performance of a statistical parser, and so requires far less training data. Customization of the semantic model remains an issue, and the system is restricted to a set of semantically tractable queries, which impairs coverage.

1.2 Query Interfaces Based on Conceptual Authoring

Conceptual Authoring (CA) using NLG [9] has been employed as an alternative to natural language input in order to remove the query interpretation step [1,10]. In querying systems based on CA, the query is constructed incrementally by the user, through successive interactions with a natural language text (termed *feedback text*). Changes to the feedback text directly reflect underlying changes to the semantic content of the query; so whilst the user is always presented with a natural language text, the query is always encoded in a structured internal representation. CA has been used successfully in building query interfaces [1,10] with evaluation showing positive user feedback and a clear preference over using SQL [1].

The feedback text shown in Figure 1 represents a simple SELECT query against the Orders table of the MS Northwind database, expressed by the SQL query in Figure 2. The words in square brackets are anchors and represent sites where the user can reconfigure the query; for example by changing a literal value, setting an aggregate function, removing a selection criterion or adding further criteria or ordering conditions.

List orders which
- were processed by [any employee]
- conisted of [total freight]
- were shipped between [1/4/97] and [3/31/98]
- [further criteria] ordered by [total freight]

Fig. 1. A sample feedback text query

```
SELECT Orders.EmployeeID, Sum(Orders.Freight) AS Shipping
FROM Orders
WHERE Orders.ShippedDate Between #4/1/1997# And #3/31/1998#
GROUP BY Orders.EmployeeID
ORDER BY Sum(Orders.Freight) DESC;
```

Fig. 2. The SQL produced by the query represented in Figure 1

2 Feasibility Study

In a previous attempt [11], we investigated the possibility of inferring some basic resources automatically, however this attempt did not reach far enough and it also resulted in relatively clumsy natural language queries. In this section we provide a high level summary of a recent feasibility study undertaken by the authors; a more complete discussion is presented in an auxiliary technical report [12].

The feasibility study focused on a prototype which is a modified version of a previous CA-based query interface [1]. We leave the task of inferring the domain ontology to others [13], and focus on the inferencing of the resources required by the NLG system. The prototype receives as input a model of the database semantics and a domain ontology, and it automatically generates some of the components and resources that in previous Conceptual Authoring querying systems were constructed manually, along with a module which translates the user-composed query into SQL . The resulting query system provides a user interface based on a feedback text (see Section 1.2) which is generated by the query system from a semantic graph. User interaction with the feedback text results in changes to the underlying semantic representation and a new feedback text is generated to update the display. When the query is run the underlying representation is converted to SQL using the model of the database semantics from which the query interface system was inferred.

Table 1. Evaluation results

	Petstore			Northwind		
	Actual	Identified	Accuracy	Actual	Identified	Accuracy
Entities	28	28	100%	49	49	100%
Relations	30	28	93.3%	61	61	100%
Entity lexical desc	28	22	78.5%	49	49	100%
Relation lexical desc	30	20	75%	61	33	54%
Entity part-of-speech	30	30	100%	49	49	100%
subcat frames	30	20	75%	61	33	54%

Although the system is supplied with a domain ontology it still needs to analyse the structure of the database to support the mapping from the semantic model to syntactic and lexical resources. The metadata analysis focused on the following elements as described in the SQL-92 Information Schema: Domain Descriptors (§4.7), Column Descriptors (§4.8), Table Descriptors (§4.9), and Table and Domain Integrity Constraints (§4.10). Since both sample databases are highly normalised a simplistic approach in which tables are identified as kernel entities and their columns are represented as properties was productive, and foreign key definitions sufficed to infer associations between kernel entities. In commercial databases, in the authors' experience, the system would need to locate inner associative entities, and although Query Expression metadata from derived tables and view definitions would aid this process it would be unlikely to prove sufficient. We propose to address this problem by looking at related metadata such as ERDs and ORM mappings.

The system also requires linguistic resources in order to represent the query as text. Some linguistic resources, such as the grammar and lexicon, are reusable, but the mappings from the ontology to the linguistic resources must be inferred. For example, the system needs to choose an appropriate lexicalization for each entity, and an appropriate syntactic frame and lexicalization for each association. It also requires domain-specific semantic and linguistic resources to manage literal values, spatial and temporal modifiers and sub-language jargon. In the case of the prototype, the identification of the lexical descriptors was made easier due to the fact that both databases use clear and reliable column naming conventions, a feature which could not be relied upon in a commercial database. In future versions of the system we intend to use related corpora, such as user manuals or domain-specific technical documentation, to support the inferencing of semantic and linguistic resources and the generation of mappings between the concepts in the domain ontology and the syntactic subcategorisations and lexical anchors required to express them.

We evaluated the prototype using two sample databases, MS Northwind and Petstore. The generated system had a coverage of 71.6% – 179 of 250 questions which could be asked of the database were supported by the interface. We assessed the system's ability to infer resources automatically by comparing the resources constructed by the generator with a manually constructed NLIDB system for each sample database. The results of this comparison are presented in

Table 1, and show that whilst resources that rely on database metadata can be identified quite accurately, linguistic resources which are identified using heuristics over textual metadata are less reliable.

3 Conclusions

Although the prototype made several simplifying assumptions about the nature of the database, it serves as a proof of concept showing that simple inferencing techniques can achieve results, and highlights the areas where more complex techniques are required (in particular, the identification of linguistic descriptions for relations). However, it is not the case that automatic inferencing will always be possible, and the limiting factors set out below may entail supervision, or may even mean that no resources can be inferenced at all. Our future research plans include extending the scope of our metadata analysis to include additional sources of information, as discussed above, to address these limitations.

Structure and Normalisation
Codd [14] proposes an extension to the relational model (RM/T) to align it with predicate logic. In this context he introduces the notion of ”property integrity”, under which each entity is represented by a single-valued E-relation and its dependent properties (or characteristics) are grouped into subsidiary P-relations. Inferring semantic dependencies from a database normalised to this extent (to 4NF) would be trivial; conversely, a data warehousing database with a flat structure and a large number of columns per relation might prove intractable.

Atomicity
Our task is made much easier if the entities within the database are represented atomically; of course in practice this will seldom be the case as the entities modelled by the database will have feature-based values which will be decomposed into column tuples. Whether or not the system can recognise these tuples will depend on a variety of incidental database-specific factors. There are various heuristics which we can throw at the problem, for example: using a domain ontology; scanning query expressions in derived tables or cached queries for common projections; analysing data content, and so on.

Nonetheless any relational database will almost certainly contain many non-atomic characteristic entities and, unless it is in 4NF, it is highly unlikely that they can all be recovered automatically. This is therefore an area where the system will require supervision if it is to function effectively.

Lexicalisation and Symbolic Values
The process of mapping entities onto concepts in the domain ontology will often involve string matching, either as part of the process of attempting to infer a semantic class or as a fallback strategy. In some instances the column names will be meaningful strings and there will also be string descriptions, in others there may be little lexical information available at all. Similarly, the data values may

be more or less tractable to the system; in particular if the data consists only of symbolic values or field codes then we can infer very little about its meaning.

Metadata Quality
In practice we cannot rely on the quality of metadata in the field. For example, we may encounter databases where foreign key information is not defined in the metadata, it is simply known to developers, where columns are mislabelled, for example due to merging of legacy data sets, where default values, unique constraints and referential constraints are not formally encoded, and so on. In such instances the system will be unable to infer the information required to build the query engine.

References

1. Hallett, C., Scott, D., Power, R.: Composing questions through conceptual authoring. Computational Linguistics (2007)
2. Kaplan, S.J.: Designing a portable natural language database query system. ACM Trans. Database Syst. 9(1), 1–19 (1984)
3. Harris, L.R.: The ROBOT system: Natural language processing applied to data base query. In: ACM 1978: Proceedings of the 1978 annual conference, pp. 165–172. ACM Press, New York (1978)
4. Alshawi, H.: The Core Language Engine. ACL-MIT Press Series in Natural Language Processing. MIT Press, Cambridge (1992)
5. Tang, L.R., Mooney, R.J.: Using multiple clause constructors in inductive logic programming for semantic parsing. In: EMCL 2001: Proceedings of the 12th European Conference on Machine Learning, London, UK, pp. 466–477. Springer, Heidelberg (2001)
6. He, Y., Young, S.: A data-driven spoken language understanding system. In: IEEE Workshop on Automatic Speech Recognition and Understanding (2003)
7. Kate, R.J., Mooney, R.J.: Using string-kernels for learning semantic parsers. In: Proceedings of ACL 2006, pp. 913–920 (2006)
8. Popescu, A.M., Etzioni, O., Kautz, H.: Towards a theory of natural language interfaces to databases. In: IUI 2003: Proceedings of the 8th international conference on Intelligent user interfaces, pp. 149–157. ACM Press, New York (2003)
9. Power, R., Scott, D.: Multilingual authoring using feedback texts. In: Proceedings of COLING-ACL 1998, Montreal, Canada, pp. 1053–1059 (1998)
10. Evans, R., Piwek, P., Cahill, L., Tipper, N.: Natural language processing in CLIME, a multilingual legal advisory system. Natural Language Engineering (2006)
11. Hallett, C.: Generic querying of relational databases using natural language generation techniques. In: Proceedings of the 4th International Natural Language Generation Conference (INLG 2006), Sydney, Australia, pp. 95–102 (2006)
12. Hallett, C., Hardcastle, D.: A feasibility study for a bootstrapping nlg-driven nlidb system. Technical Report TR2008/07, The Open University, Milton Keynes, UK (2008)
13. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches, pp. 146–171 (2005)
14. Codd, E.F.: Extending the database relational model to capture more meaning. ACM Trans. Database Syst. 4(4), 397–434 (1979)