Automatic Relational Schema Extraction from Natural Language Requirements Specification Text

S. Geetha and G.S. Anandha Mala

JNTU Hyderabad andhra Pradesh, India
Head / CSE, St. Joseph’s College of Engineering, India

Abstract: Recent researches have been focussed greatly on automating the extraction of information from Natural language text and this requires a large amount of domain knowledge. This paper attempts to construct a schema that depicts the interrelationship between tables extracted from natural language requirements specification text. Initially the domain knowledge is extracted from requirements specification text. Following this the schema generator extracts the attributes from the class diagram generated by the domain knowledge extractor. By prioritizing the preferences of the attributes and adjectives present in the text, the key attributes (primary and foreign) are elicited using hand crafted rules. This is done in order to derive the schema for the tables present in the requirement specification text. The extracted schemas are then represented in XML and they are stored in a relational database. Finally the database constructed can be queried to acquire the domain knowledge which eases the job of software development.

Key words: Relational schema • Key attributes • Hand crafted rules • Natural Language Processing (NLP)

INTRODUCTION

In most of the applications, information is available in the form of NLT (Natural Language Text) rather than the structured database. Discovering knowledge from NLT is an important aspect of Knowledge Discovery Database (KDD). Similarly, text mining also makes productive use of NLT. It comprises the processes of structuring the input, extracting patterns and interpretation of output.

The process of software development focuses mainly on capturing the natural language requirements specification such as understanding the user needs and goals. Initially, software development starts with requirements capturing which is expected to provide a total understanding of the system which in turn describes the needs that are to be satisfied. If the requirements are not structured, then translation is needed.

So far, many algorithms have been proposed to semi-automate or automate the transformation of natural language SRS (Software Requirements Specification) to a structured one. In fact, the job of software development is to make this process easier and thus representing the database and acquiring knowledge by applying queries will become simpler.

However, translation of natural language SRS to usable software poses quite a lot of challenges. A number of issues need to be addressed in designing the relational schema of a database. Identifying the referential integrity constraints enforced on the database is a challenging task. Not only that, automatically discovering the semantic associations between the table schema elements, namely PK’s (Primary Key) and FK’s (Foreign Key) are also equally challenging. Thus a methodology for converting natural language SRS text into the relational database has to be developed. It must help in identifying the schema for the tables and the relationship corresponding to all the tables extracted from the requirements specification.

This paper introduces a transformation technique which helps to bridge the gap between the informal natural language requirements specification and the formal language specification so as to offer effective software solutions. The mapping rules and rule set approaches are used to identify PK’s and FK’s to construct the relational schema. The automatic construction of the relational schema from natural language SRS provides a platform to extract the information for specific domain.
The Rest of the Paper Is Organized as Follows:

- Related work is discussed in section 2
- Proposed methodology is explained in section 3
- Section 4 contains the procedure for results and discussion for evaluation of the different SRS
- Section 5 discusses the conclusion and future work

Related Work: The construction of a schema for a database is a crucial step in data analysis. Many researchers have attempted several methods to achieve the above task. Rohit Kate [1] extracted and presented the entities and their relationship in a sentence instead of text where a schema cannot be constructed. Yannis Sisman’s [2] proposed an algorithm called “Gordian” to discover composite key attribute from large datasets.

Inclusion dependency between single attribute and sequences of attributes in a given database was considered in constructing the schema by F. D. Marchi [3]. Jana Bauckmann [4] proposed a schema, which identifies Inclusion dependencies as specified precondition for foreign keys. For the generation of relational database schema, he used open Mms schema and parser to import the protein database. Identification of relationship between the columns, which explains the properties and characteristics of the values from the relational database, was studied by Meihui Zhang [5]. Though sample work has been done in constructing the schema for the analysis of database, many ways of building the schema for the database from NLT is still in the preliminary level. Significant work was done by Luis Tari [6], in proposing a method of information extraction from NLT followed by Xian-Yi Cheng [7] who worked on constructing the database based on, physical relation and generic relation.

The authors Divesh Srivastava [8] and Marco D. Adelfio [9] also have done extraordinary work in extracting the schema from complex database and metadata in tabular form, but they did not consider the relational schema construction from the NLT. G.S. Anandha Mala [10] proposed a method to extract Object Oriented Elements (OOE) from requirements specification text, Mfourga [11] extracted the entity relationship from the schema of the relational database. But this work did not give importance to construction of relational schema for the database. Michael J. Cafarella [12] proposed an algorithm TGen to discover the schema automatically for the extracted data, but he did not specify any information about the removal of data duplication and relationship among the columns. Jekub Piskorski [13] presented different NLP techniques to extract useful information from NLT.

These are some of the works which focussed on developing techniques to extract knowledge from Natural Language Text. The proposed system is built to construct the database by extracting the schema information that describes the logical structure of the tables in an automated manner, there by considering the natural language requirements specification.

MATERIALS AND METHODS

Identification of the PK / FK attribute is an elemental concern for different data management tasks such as extracting the relationship and queries. To build the database, discovering the keys and establishing the relationship between the tables using the keys to identify the relational schema is a crucial step in understanding and working with natural language SRS. The proposed methodology deals with automatic construction of the relational schema by identifying the key attributes to build the database from SRS using “rule based approach”. The Fig. 1 shown throws light on the system architecture of schema extractor.

Domain Knowledge Elicitor: The domain knowledge elicitor describes the user needs as input in the natural language SRS text. The problem statement splits the SRS into sentences to reduce the processing overheads. The PoS (Parts of Speech) tag designates each word of all the sentences and classifies the words as nouns, verbs, adjectives, etc. Brill is used for this purpose [14, 15]. Tagging of the words is necessary to chunk the words that form noun phrases or verb phrases, the noun and the verb phrases are classified based on simple phrasal grammar. When normalizing the sentence into Subject-Verb-Object (S-V-O) pattern, sometimes the subject and object happen to be pronouns and they have to be resolved to their respective noun phrases [16]. Then the sentence has to be interpreted into S-V-O pattern to map the words into OOE [3].

Generation of Class Diagram: The OOE namely classes, attributes, methods and relationships are identified based on simple rule based approach from the S-V-O pattern [3].
Fig. 1: System Architecture of schema generator

- Translating Nouns to classes
- Translating Noun-Noun to class property according to the position
- Translating the lexical verb of a non-personal noun to a method of this noun
- Translating S-V-O structure to a class diagram with the subject and object as classes, both sharing the verb as a candidate method

Schema Generator: This module extracts the schema using the OOE, which are identified by extracting the attributes from the class diagram. Then the PK and FK are identified using rule based approach. Later the schema is represented as XML and stored in the database.

Extraction of Attributes from Class Diagram: A relational schema represents the various properties of the tables and attributes (columns) within the tables.

Identification of Primary Key (PK): Identification of the PK attribute maintains the uniqueness of the table. Hence a rule based approach is proposed to identify the primary key attribute from the attributes of all the tables. The attributes play a vital role in forming the metadata of the database. The PK attributes should have the following properties:

- The instance of an entity must have a non-null value.
- Each instance must have a unique value.
- No change in the values.

This is achieved by applying the hand crafted rules for retrieving the PK attributes. The rules are as follows and sample is shown in Fig. 3.

Rule 1: When the sentence is in the form of “Subject+ Possessive verb + Adjective + Object”, then the object is a key attribute.

Rule 2: When the sentence is in the form of “Subject+ Possessive verb + Object” and if the object is prefixed or suffixed with set of predefined (training data set) words, then the object is a key attribute.

Rule 3: When the sentence is in the form of “Subject+ Possessive verb + Object”, if the object is in the training data set, then the object is considered as a key attribute.
Rule 4: By prioritizing the priority assigned to the objects of the form “Subject+ Possessive verb + Object1”, “Subject+ Possessive verb + Object2”, based on the relevance between the subject and object1, object2, the highest priority object is nominated for the key attribute.

The following are the sample sentences taken for identification of the PK attribute and observations found by applying the above rules.

Rules Applied to Identify the PK Attribute with Sample Sentences:

**Sample 1:** The/DT passenger/NNP have/VBP the/DT unique/JJ ssn/NN,/, name/NN,/, address/NN,/, age/NN,/, ticket_no/NN,/, journey_date/NN.

When Rule 1 is applied for this sample, it is observed that PK attribute is identified based on the possessive verb which follows any one of the object with the selective adjective maintained in the library.

**Sample 2:** The/DT aircraft/NN has/VBZ flight_no/NN,/, flight_name/NN,/, route/NN,/, take_off_date/NN,/, no_of_seats/NN,/, fare/NN.

Rule 2 is applied for this sample and it is observed that PK attribute is identified based on the possessive verb which follows any one of the object and is suffixed with the predefined Words in training data set.

**Sample 3:** The/DT ticket/NN has/VBZ ticket_no/NN,/, seat_no/NN,/, flight_no/NN,/, date /NN,/, source/NN and/CC destination/NN.

Rule 4 is applied for this sample. It is observed that PK attribute is identified based on the possessive verb which follows the object1, object2, etc. Initially the objects are assigned with equal priority value as 1. Later the object name is compared with the subject; if there is a similarity between them then priority value is increased to confirm one of the objects as PK attribute.

**Identification of the Foreign Key (FK):** As PK attribute of all the tables are identified using the rule sets, the next step is to identify the FK attributes. FK attribute can be a single attribute or combination of more than one attribute which forms relationships across the database tables. To be an FK attribute of the database tables, the attribute must satisfy the following:

- The FK/PK attribute name must be similar.
- The FK should cover almost all the PK.
- The FK should have cardinality 1:1, 1: N, M: N.

The system checks the PK attribute of all the tables. If the occurrence of the primary key attribute of all the tables matches with the attributes of other database tables, then assign it as foreign key attribute. This is shown in Fig. 4.

**Database Schema Definition:** Defining the database schema is very important for data analysis. The database schema is designed to store information about the tables extracted from SRS. The table includes a set of attributes and each attribute is associated with data type. The type of attribute is determined based on the attribute from which it derives its properties. So every attribute of the table is assigned with data types that contain a specific type or range of values. The data types considered to assign for the attributes of the table are number, character, date, etc. After distinguishing the key attributes (PK/FK) from all the tables, data type has to be assigned without violating the referential constraints. The following procedure is followed in order to assign the data type for each attribute.

- Assign date as data type for the attribute whose name has date as substring.
- If date is not the data type, then the attributes are assigned a character or number as the data type. In order to resolve these issues, a set of predefined data sets words used in the rule 2 are used for identifying the primary key attributes.
- Once type is assigned to the attribute, the database schema shows all the tables, the attributes in all the tables and the relationship between the attributes of one table with other.
XML (eXtensible Markup Language) Representation and
Relational DB: The tool has been developed to convert
the relational schema extracted from the SRS into XML,
which is an intermediate representation to be imported
into the database. The data which is getting stored in the
XML has to follow the structure as XSD (XML Schema
Definition). The XML database can be constructed by
considering the attributes of all the tables and its
constraints. The code snippets for the different
representation of the elements in a DB are shown in
Fig. 5. Then a simple java library called Jackcess, is used
to write the XML schema definition into MS Access
database which shows all the database tables including
key attributes, non-key attributes, data type and its size.

RESULTS AND DISCUSSION

The entire work (tool) was implemented using java
and it was validated using around 50 samples of 100 lines
approximately. The proposed approach creates a tool to
generate relational schema automatically from the input
SRS. The core idea is to identify the dependency of the
attributes between the tables based on rule based
approach. The process of finding the relationship
between the attributes starts with the requirements
extractor which interprets the SRS into S-V-O to map the
word into OOE. The identified S-V-O patterns are used to
classify OOE as class, attributes, methods and
relationships using hand crafted rules. The crafted rules,
priority and training data set are used to identify the PK
attributes. With the use of PK attributes, FK attributes are
identified based on the similarity of the attributes between
the tables and the data type is assigned to all the
attributes.

The construction of the database schema is shown
in Fig. 6. The database schema to be converted into XML
is imported into relational database without violating the
integrity constraints shown in Fig. 7.

The tool should not miss to identify the PK and FK
attributes. But approximately 12.92 % and 10.61 %
additional primary key and foreign key attribute are
identified by the schema generator based on true
positives and false positives. The tools are tested for the
following domains like Airline ticket reservation, banking
system, retailing system and so on.

Database MS - Access: Fig. 8 shows accuracy of key
attribute identification over 10 sample documents.
Accuracy is measured as

\[
\text{Accuracy} = \frac{\text{Number of True Positives} + \text{Number of True Negatives}}{\text{Number of True Positives} + \text{Number of False Positives} + \text{Number of False Negatives} + \text{Number of True Negatives}}
\]
Fig. 6: Automatic generation of database schema

Fig. 7: Automatic generation of table schema in the relational Database MS - Access

Fig. 8: Results obtained on long complex document. [Document length not exceeds 500 complex sentences and sentence length not exceeds 25-30 words.]
CONCLUSION

This approach presents an idea to restructure the natural language text into structured information. Natural language processing techniques and set of rules are used to extract the domain knowledge from the natural language SRS. The system compactly constructs the relational schema by identifying the class diagram, schema for the tables and relationship between the tables using primary key / foreign key attributes. Thus a set of hand crafted rules are presented to identify the key attributes to construct the relational schema. Then the relational database is automatically constructed from the relational schema which is finally converted into XML. The user can query and acquire domain knowledge from the relational database which is built from natural language SRS. The developed tool is found to extract the schema efficiently which is indicated by performance measures like True Positives and False positives. This method could be extended to identify the relationship between the databases.

REFERENCES


