Academic Science

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Issue 10
October 2015

Techniques to automatically generate Entity Relationship Diagram

Ronak Dedhia
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai University, Mumbai, India

Atish Jain
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai University, Mumbai, India

Prof. Khushali Deulkar
Dept. of Computer Engineering
D.J. Sanghvi College of Engineering
Mumbai University, Mumbai, India

Abstract— An Entity-Relationship Model (ERM) plays a central role in developing the structure of business systems. Due to abstract nature and high level of conceptual data, ER modeling is an overwhelming task for system analysts. With advancements in Artificial Intelligence, researchers have been trying to automate the process of generating ER diagram from problem statement in natural language such as English. This paper focuses on a technique to combine both syntactic analysis and semantic heuristics for extracting the major components such as entity, attribute, and relationof Entity-Relationship diagram. This can be done by evaluating various rules discovered by prior work and combining them effectively resulting into an efficient NLP engine.

Index Terms—Entity-Relationship diagrams, ERD, database design, NLP, Artificial Intelligence, requirement analysis, semantic heuristics, natural language.

I. INTRODUCTION

IN almost any organization for carrying out a business at a large scale, databases are an integral part. Database schema plays an important role in developing the structure of business systems. Database modelling can be an overwhelming task because of its conceptual nature and technicality. Artificial Intelligence (AI) with its highly specialized and expert techniques has become an essential part of technology industry. AI has provided solutions to some of the most challenging problems in Computer Science. So why not let AI handle the entire complex process of database design, beginning with a simple input text from the user to the generation of Entity Relation (ER) diagram or multi level schemas.

Natural Language Processing (NLP) is one of the central goals of AI. NLP can be used to achieve automation for generating ER diagram. Lot of research has been done in application of structural analysis for generation of ER diagram. Thus structural analysis along with semantic heuristics can be utilized to get ER components such as entity or attributes from natural language text. In this paper, we explore the various techniques used by the existing tools, for extracting the requirement specification from the problem statement in English language.

II. LITERATURE REVIEW

This section explains the concept of ER modeling and also review the previous works that apply NLP to databases. Some approaches are fully automatic, whereas others have human intervention at some point, making them semi-automatic.

A. Overview of ER Model

The first phase in designing a database application is requirement analysis. It helps us understand all the important data that must be stored in the database. This information is then conceptualized into high-level description of data. This is done by designing the Entity-Relationship model. An ER model can be thought of as a blueprint of data which will help us understand the complexities of a functional system. ER models facilitate interaction among system analysts, designers, application programmers and end users.

The main components of ER model are entity, attributes, relations, and cardinality. In real world modeling, an entity represents a distinguishable business object. A collection of entities containing more than one property is called an entity set. Examples of entities are “Employee”, “Shop”. It is represented using rectangular box in the ER diagram. An attribute describes a property or characteristic of an entity. For example: name, age, address can be attributes of the entity employee. Attributes are represented using ellipse in the ER diagram. A relation between entities is represented by diamonds in the ER diagrams. For example, a teacher teaches many students. Here “teaches” implies a relation between entities teacher and student. Entities can be related in one-to-one or one-to–many relation. This is said to be cardinality of one given entity in relation to another.For

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Issue 10
October 2015

example, a single teacher can teach many students, so the cardinality is one-to-many.

B. Related Work

Entity-Relation modeling was first developed by Peter Chen [1]. He presented rules to generate conceptual model elements from the given problem statement. Table 1 presents the rules of thumb, proposed by Peter Chen, for mapping natural language descriptions in ER models. We know that English sentence has eight parts of speech (POS) namely, Nouns, Pronouns, Verbs, Adverbs, Conjunctions, Prepositions and Interjections. Thus, basic rules use these POS for tagging with ER diagram components. As basic ER diagram was limited, advanced concepts such as specialization and generalization were added to solve complex problem statements. It was then, called, Extended Entity-Relation diagram (EER).

Parts of Speech	*ER component*
Proper Noun	Entity
Common Noun	Entity Type
Adjective	Attribute for entity
Transitive Verb	Relationship between entities
Intransitive Verb	Attribute type
Adverb	Primary Key/ Attribute for Relation
"There are X in Y"	Relation between Y and X as 'has'
"The X of Y is Z and Z is proper noun"	Y and Z are entities and X is relation between them;
"The X of Y is Z and Z is not proper noun"	X is an attribute of Y. Y is an entity and Z represents a value
Gerund	Relationship converted into entity type
Noun followed by another noun	Both noun are attributes

Table 1: Rules of thumb for mapping natural language in ER models

Abbott [2] proposed a method for analyzing requirements from problem statement in natural language. It was further developed by Booch [3,4] by introducing object-oriented methodology for designing Unified Modeling Language (UML) models. However, both methods were done mechanically and not automated. Their method required knowledge of world and understanding of problem statement. LOLITA [19] abbreviated as Large-scale Object-based Language Interactor Translator Analyzer. LOLITA is developed to output object models automatically using NLP. In this approach, nouns were tagged as objects and linkswere used to find relation between those objects. Although better than previous mechanical methods, LOLITA could identify only classes and could not extract objects in different Natural language specification.

DMG [5] is a semi-automatic technique which maintains various rules and heuristics in its knowledge base. From the natural language text, a parsing algorithm understands the grammar and represents each individual word as a lexicon. Further by using heuristics, a relationship is set up between linguistic input and design knowledge. If there is any particular word missing from the lexicon or if there is some ambiguity in the mapping of input rules then, in such cases, human intervention is required, making the process semi-automatic. Linguistic structure so formed is then converted into ER diagram. DMG proposed significant heuristics for transforming natural language in ER models and also Extended ER models, however, this technique has never been developed for practical use.

CM Builder [6] was developed to support requirement analysis of software development using NLP techniques. It build's an integrated discourse model which is depicted in a semantic network. UML model is generated automatically from the semantic network. However, its limitations are some lexical analysis. For example, Limitation of attracting post modifiers such as relative clauses and prepositional phrases. Also CM builder has static knowledge base. Thus, it becomes difficult to update and be adaptive to different problem cases.

N. Omar et al [18] conducted a study to extract semantic knowledge, from the natural language problem statement to produce ER models. They show that, using semantic lexical knowledge along with syntactic heuristics, yields more accurate and precise results. Semantic analysis helps us solve wider range of problems such as anaphoric references or nominalization. More expressions can be added by understanding the results of parsing. Semantic roles are conceptual notions that can be used for depicting certain arguments of verbs [9]. Some semantic roles are AGENT, THEME, GOAL, SOURCE, etc. For example, "The employee (AGENT) has been appointed (THEME) as the new manager (GOAL). Thus, depending on context and the semantic cues, the subject and object are given separate roles. Each heuristic is

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Issue 10
October 2015

assigned a weight, based on common sense, to provide optimal solution to complex problems. The value of weights represent confidence level for an event to be true. For examples, if HE2 (heuristic rule) is assigned a weight of 0.6 then it means the heuristic will produce true results 60% of the time. This technique of assigning heuristic weights in training sets evolves as we develop the data set of problems. This developed system which combines both semantic and syntactic rules for ER diagram generation can be used for modeling intelligent tutor systems.

Meziane [16] uses semi-automatic approach for obtaining ER diagram from natural language specifications. It converts the natural English language input into Logical Form Language (LFL). These logic forms are used as basis for identifying entities, attributes and relationship between them. Using heuristics, suitable degrees are assigned for identified relationships. This approach is highly dependent on the quantifiers for identifying the degree of relations. Two definite and indefinite existential quantifiers are 'the' and 'a'.

MacDonell [17] designed a tool that assist's the system analyst in selecting and verifying terms which are relevant to the project. The main reason for such a tool is 40% to 60% of database modeling errors, which are made while finding the components for the model. This verification is done with the help of dictionary, which, at the time of implementation had about 32000 entries and 79 rules. To further remove disambiguate in certain complex problems, the dictionary and rule set can be expanded. The burden of theanalysis,requiring the system analyst to carry out parsing, selection and relatingobjects of interest from specification documents – can be transferred at least in fractions to a toolset that is able to perform these tasks intelligently and automatically.

E-R generator [5] is an alternate rule based tool. The E-R generator is based on two types of rules: specific rules associated to semantics of words in sentences, and generic rules that differentiate entities and relationships on the premise of the logical form of the sentence. The knowledge representation structures are developed by the Natural Language Understand (NLU) framework which utilizes a semantic interpretation approach. For the connection of attributes and determination of anaphoric references, the framework requires user intervention keeping in mind the end goal.

The English language input has some indicators such as initial capitalized letters which represent entities such as name, person, place, etc. However, it is very difficult to identify components in other languages such as Arabic, Chinese, and Hindi. A tool based on knowledge base to produce ER model from German Language input is Dialogue tool [10]. Manika Nanda [11] developed a framework which can identify named entities in Hindi language input text. For this system, she built her own database which is bilingual. Here bilingual means maintaining a database in both languages, i.e. Hindi and English. Database contains mainly three columns- words, their Hindi transliteration and entities such as name, organization and place with respective sub-categories.

Circe [14], is a web-based system for providing natural language requirement gathering, elicitation, selection and validation. The requirements are supplemented by dictionary to specify all the systems and domain specific terms used in input. Here the NLP engine has relevant knowledge about requirement specifications of ER model obtained by a-priori algorithm.Table 2 is a comparison table of all tools and techniques discussed.

Table 2 given below is a comparison table of all tools and techniques discussed.

Table 2. Comparison table of tools/techniques

Sr. No.	*Tools/ Design Developed*	*Advantages*	*Limitations*
1	ER Diagram, Peter Chen [1]	Basic rules for mapping of ER models (rules of thumb)	Insufficient rules for systems having semi-structured data
2	Abbott &Booch [2] [3]	Brings in Object Oriented Methodology for designing data models	Had to be done mechanically, overwhelming and not automatic
3	LOLITA, L. Mich [19]	Automatically analyzed requirement specification from input text in natural language	Only limited to extracting classes and could not extract objects
4	CM Builder, H. Harmain [6]	Better than LOLITA and generates UML networks using semantic	Cannot attract post modifiers like relative clauses and prepositional

		network	phrases
5	Novel approach by Meziane [16]	Converts natural input into Logical Form Language (LFL) and use of quantifiers for determining degree of relation	Cannot handle structured objects such as tables
6	MacDonell [17]	Assisted analysis process for identifying ER comonents using 79 rules and 320000 entries in knowledge base	Static knowledge base and its expansion can be a burden
7	Entity recognizer framework, Manika Nanda [11]	Technique that can identify components of ER model using Hindi language input text	Database maintenance for two language transliterations is time consuming and mechanical
8	Circe, V. Ambriola [14]	Web-based system for validation selection of ER components using a-priori algorithm	Cannot be used for input in languages other than English such as Arabic, Chinese.
9	ER-Converter, N. Omar [18]	Makes use of semantic roles such as 'AGENT','SOURCE','GOAL' for semantic lexical knowledge	Requires human intervention for resolving conflict in assigning of heuristic weights

III. THE NLP ENGINE

A set of NLP modules were built for LaSIE (Large Scale Information Extraction) system at Sheffield. [7,8] It has been efficient in processing words in natural language. NLP system has three main processing stages.

A. Lexical Preprocessing

This step consists of four phases: tokenizing, splitting, tagging and chunking or analyzing. The general input to the preprocessor is a text file, in natural language text, bearing the description of the problem. The output is a set of charts, one for each sentence, which aids the parser to understand the complexity of problem statement by applying predefined rules.

1) Tokenization:

This phase separates plain text file into tokens. They are also called as lexicons. For example separate words, punctuation identifying numbers, and so on. Usually a 'space' can be used as a qualifier to identify individual words.

2) Sentences Segmentation:

This phase identifies the sentence boundaries. The entire problem statement is broken down into individual sentences. Usually this is done by analyzing the punctuation mark, period (.). The individual sentences formed are then analyzed to extract the requirement specifications.

3) POS tagger:

In this phase, we attach tags to the separated tokens. There are two ways in NLP, rule-based and stochastic taggers. In stochastic tagger a lot of statistical data has to be captured from conceptual information. This data is usually a table for trigram statistics, indicating all tags i.e. taga, tagb, and tagc and also probability that taga follows tagb and tagc.

However, rule based tagger by Eric Brill captures information in less than eighty rules. This makes for an unambiguous clear tagger. Contextual information is stored in much more compact and understandable format. Also a rule based tagger automatically learns it rules and can perform very well. Thus, it should be used over stochastic tagger for ER diagram generation.

4) Chunking and Morphological Analysis:

Once tagging of POS is done, we find the root and suffix of each word in each sentence. This means the word is examined for singular/plural, tenses, etc. For example, a noun like "students" will be interpreted as "student + s". Verbs in continuous tense such as "teaching" will be interpreted as "teach + ing". These roots and suffixes are then fed into the parser.

B. Parsing and Heuristic Classification

Parsing process is easy for a human but difficult for a software. In this process we decide which word relates to a component of an ER diagram. Natural language

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Issue 10
October 2015

has multiple possible analysis due to its ambiguous grammar. Parsing determines parse tree of a given sentencewherein, a group of words is transformed into structures that depict how the asentence's units relate to each other. [12]

This transformed structure, highlights the different fundamental parts of a sentence such as subject,

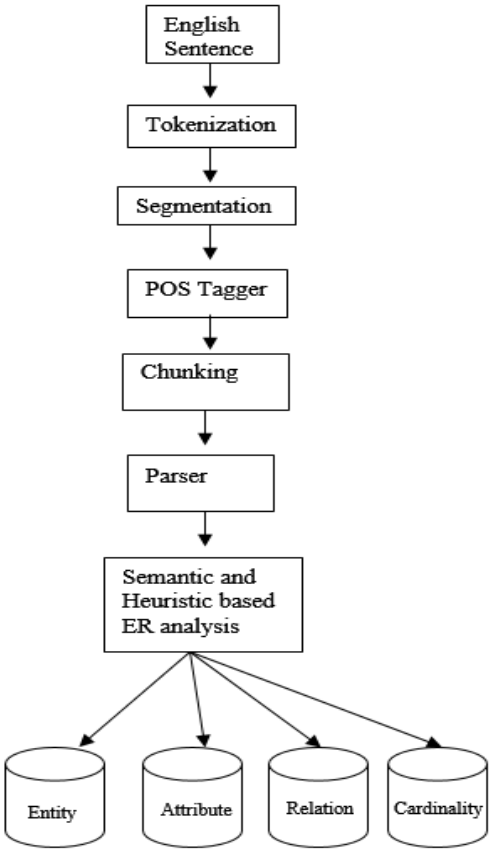

Figure 1.Structure of NLP Engine

object, and so on. It uses the predefined rules of grammar (mentioned in table 1) to classify the POS Tags. For example, nouns can be interpreted as entities or attributes, whereas verbs can act as a relation between them. The cardinality of components can be extracted from adjectives so rules for determining different components like entity, attributes, etc. are given in table 1. There are two different types of parsers that can be used. The Memory Based Shallow Parser (MBSP) and the bottom up chart parser. [12] The enhanced versions of these parsers can be used can be used for parsing. Programming languages like Prolog or LISP can be used to implement these parsers [13].

Once the parsing is done the parsed text is fed to the NLP engine for identifying data modeling elements. Semantic and syntactic heuristics are applied and depending on the confidence level, weights are assigned to the heuristics. These assigned weights have values ranging from 0 to 0.9. They are updated automatically depending on precision, for correctly identifying elements. Thususing a data set optimized for specified NLP engine is theorized to produce a more accurate output.

C. Generation of final ER diagram

Once all the components of ER diagram are identified by the system, a manual check by human user should be done. Attributes are attachedto their corresponding entities and cardinalities are assigned to relations between the entities. Thus we get final Entity-Relation diagram. Figure 1 shows the working structure of the NLP engine.

D. Evaluation Criteria:

One of the methods used for evaluating information extraction systems is the Message Understanding Conferences (MUC) evaluation, which can be resolved into recall and precision. [9] Percentage of all correct answers produced by the system is called recall while the answers correctly identified by system isprecision. For an ideal system, the recall and precision percentages should be 100%. Many other evaluation criteria can be developed to judge the system such as Over-generated components, undegenerated components or wrongly-attached entities, are some of the evaluation parameters devised by Nazlia et al in [18]. For an ideal system, all the new parameters should be 0%.

IV. CONCLUSION AND FUTURE WORK

In this paper, we described the approach towards generating ER diagram automatically using Natural Language Processing technique. First we saw systems like CM builderandLOLITA which were used to extract requirement specification for conceptual data modeling. Technique by MacDonell pointed out importance of automated assistance required by designers in selecting and verifying components. ER-Converter, a tool by N. Omar brought in the use of heuristics weights for better accuracy and precision. Thus, a system which uses both semantic as well as syntactic heuristics (NLP Engine) would help us attain maximum precision and recall. Also system should be tolerant to accepting input in

multiple languages such as Hindi, English, Chinese or Arabic.

Further research should be done to improve the NLP engine using neural networks and advanced algorithms such as a-priori or Support vector Machine (SVM).

REFERENCES

[1] Chen, Peter Pin-Shan. "English sentence structure and entity-relationship diagrams." Information Sciences 29.2 (1983): 127-149.

[2] Abbott, Russell J. "Program design by informal English descriptions." Communications of the ACM 26.11 (1983): 882-894.

[3] Grady Booch : Object-Oriented Development. IEEE Transactions on Software Engineering. VOL. SE-12, NO. 2 , February 1986.

[4] Booch, Grady. The unified modeling language user guide. Pearson Education India, 2005.

[5] Tjoa, A. Min, and Linda Berger. "Transformation of requirement specifications expressed in natural language into an EER model." Entity-Relationship Approach—ER'93. Springer Berlin Heidelberg, 1994. 206-217.

[6] Harmain, Harmain M., and R. Gaizauskas. "CM-Builder: an automated NL-based CASE tool." Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on. IEEE, 2000.

[7] Gaizauskas, Robert, et al. "University of Sheffield: description of the LaSIE system as used for MUC-6." Proceedings of the 6th conference on Message understanding. Association for Computational Linguistics, 1995.

[8] Humphreys, Kevin, et al. "University of Sheffield: Description of the LaSIE-II system as used for MUC-7." Proceedings of the Seventh Message Understanding Conferences (MUC-7). 1998.

[9] Omar, N., P. Hanna, and P. Mc Kevitt. "Semantic analysis in the automation of ER modelling through natural language processing." Computing & Informatics, 2006. ICOCI'06. International Conference on. IEEE, 2006.

[10] Buchholz, Edith, et al. "Applying a natural language dialogue tool for designing databases." Proceedings of the First International Workshop on Applications of Natural Language to Databases. 15p. 1995.

[11] Nanda, Manika. "The Named Entity Recognizer Framework." International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN (2014): 2349-2163.

[12] Btoush, Eman S., and Mustafa M. Hammad. "Generating ER Diagrams from Requirement Specifications Based On Natural Language Processing." International Journal of Database Theory & Application 8.2 (2015).

[13] Gazdar, Gerald, and Chris Mellish. "Natural Language Processing in {LISP}." (1989).

[14] Ambriola, Vincenzo, and Vincenzo Gervasi. "Processing natural language requirements." Automated Software Engineering, 1997. Proceedings. 12th IEEE International Conference. IEEE, 1997.

[15] Gomez, Fernando, Carlos Segami, and Carl Delaune. "A system for the semiautomatic generation of ER models from natural language specifications." Data & Knowledge Engineering 29.1 (1999): 57-81.

[16] Meziane, Farid, and Sunil Vadera. "Obtaining ER diagrams semi-automatically from natural language specifications." (2004): 638-642.

[17] MacDonell, Stephen G., Kyongho Min, and Andy M. Connor. "Autonomous requirements specification processing using natural language processing." arXiv preprint arXiv:1407.6099 (2014).

[18] Omar, N. Heuristics-Based Entity Relationship Modelling Through Natural Language Processing. PhD Thesis (2004). University of Ulster, UK.

[19] Mich, Luisa. "NL-OOPS: from natural language to object oriented requirements using the natural language processing system LOLITA." Natural language engineering 2.02 (1996): 161-187.