International Journal of Database Theory and Application
Vol.8, No.2 (2015), pp.61-70
http://dx.doi.org/10.14257/ijdta.2015.8.2.07

Generating ER Diagrams from Requirement Specifications Based On Natural Language Processing

Eman S. Btoush and Mustafa M. Hammad

Department Of Computer Science,
Mutah University, Jordan
{emanbtoush26, mustafa.mutah}@gmail.com

Abstract

An Entity Relationship (ER) data model is a high level conceptual model that describes information as entities, attributes, and relationships. Entity relationship modeling designed to facilitate database design. The abstract nature of Entity Relationship diagrams can be discouraging task to both designers and student alike. This paper deals with the problem of extracting ER elements from natural language specifications using Natural Language Processing (NLP). The approach provides the opportunity of using natural language documents as a source of knowledge for generating ER data model. The structural approach is used to parse specification syntactically based a predefined set of on heuristics rules. Extracted words with its Part Of Speech (POS) mapped into entities, attributes and relationships, which are the basic elements of ER diagrams.

Keywords: *ER, Entity-relationship diagram, NLP, natural language, user requirement analysis*

1. Introduction

Recent researches have been focused on automating the extraction of information from natural language text using Natural Language Processing (NLP), which requires large amount of domain knowledge [1]. NLP is a field in computer science and linguistics that is related to Artificial Intelligence (AI) and Computational Linguistics (CL). Generally, NLP employed to automatically convert information stored in natural language to a machine understandable format. The main goal of NLP is to extract knowledge from unstructured data that are highly ambiguous with complex grammars to be processed [2]. Natural language processing is a field of increasing importance with growing applications such as search, machine translation, and general human-computer interaction [3].

Entity Relationship (ER) models have played a central role in systems specification, analysis and development. Moreover, ER models are used to control and monitor system's databases. In ER modeling, a system's data is modeled as a set of entities, which composed of a set of attributes, with their relationships. However, obtaining entity relationship models from a system's specifications may be a lengthy and time consuming. This paper focuses on systematic transformation of natural language descriptions to a data model.

This paper proposes an approach that uses natural language processing to extract ER elements. The approach begins by using NLP techniques to translate user specifications to words with its Part Of Speech (POS). Parsing process is proposed and a set of syntactic heuristics rules are applied to identifying entities, attributes and relationships of the target system. The rest of the paper is organized as follows; related works presented in Section 2. An overview of the proposed approach is introduced in Section 3. Section 4 shows an extraction example followed by the work's limitations. Conclusion and future works are presented in Section 6.

ISSN: 2005-4270 IJDTA

2. Related Work

This section provides a brief summary on data modeling that apply the concept of ER model and reviews previous work of applying natural language processing to databases. Drawing ER diagram is very important step in relational database design. Commercial products for ER models representation have been developed including Tech's ER Studio, Microsoft's Visio and Dia [4] .Also many research focus on develop and implement tools that draw ER diagram according to different methodologies. The ER model is represented by ER diagrams which show how data will be represented and organized in the various components. Peter Chen [5] presented 11 rules to generate conceptual model elements (entity types and relationship types) from structured sentence. Later on, Extended ER Diagram (ERD) was presented by adding new concepts like generalization and specialization [6].

Abbot [7] used heuristics for the generating ER model. Parsing techniques used in [8] [9]. In [10] CM-Builder approach used natural language processing techniques to analyze software requirements texts. CM-Builder approach build an integrated discourse model of the processed text, ER components are defined using tagging and parsing technique. Limitations in CM-Builder include some linguistic analysis. For example, attachment of post modifiers (prepositional phrases and relative clauses) is limited. Other limitations include state of knowledge bases which are static and not easily to update or adaptive [11].

In [12] an approach of generating ER elements automatically from natural language specifications using a heuristics-based approach is proposed. Semantic heuristics applied as strategy for obtain ER elements including entities, attributes and relationships. Author implies that syntactic heuristics produced good results in identifying the relevant and correct results of the ER elements. DMG [13] is a rule based design tool use heuristics approach to extract information from natural language. DMG proposed a large number of heuristics rules in both syntactic and semantic heuristics. The DMG has to interact with the user in case of ambiguous input [9].

ER generator [12] is a rule-based system that generates ER models from natural language specifications. The ER generator consists of specific rules and generic rules. The structures of knowledge representation are constructed by understanding of a natural language which uses semantic approach. The system needs assistance from the user in order to resolve ambiguities problems. In our ER generator system user help is also needed.

Large -scale Object-based Language Interactor, translator and Analyzer (LOLITA) [13] is NLP system that generates an object model automatically from natural language specification .considering nouns as objects, links used to find relationships between objects. There is no distinguishing between attributes, objects and classes. This approach is limited to identify classes and cannot extract objects in different NL specification.

A method to automatically generate a conceptual model from the system's textual descriptions presented by (A. Montes *et al.*, 2008) [14]. The requirements model is analyzed in order to establish the static structure (conceptual model) and dynamic structure (sequence diagrams, state diagrams) of the future system. In [15] ER diagram generated from free text. Natural language processing techniques was applied as first step, domain ontology applied to improve the performance of identification process. Their tool introduces a semi-automated process.

3. The Approach

Entities, attributes, and relationships are the basic elements of ER models. An entity is an object that exists in the real world and it is distinguishable from other objects. An entity type is a collection of similar entities with its own attributes. Entity type's

attributes show details structure about entities data and can be derived from adjectives and adverbs. Therefore, nouns in system's requirements can be identified as entities. A relationship is an association among two or more entities. Relationships can be derived from verbs. Key constraint in a relationship is represents by cardinality.

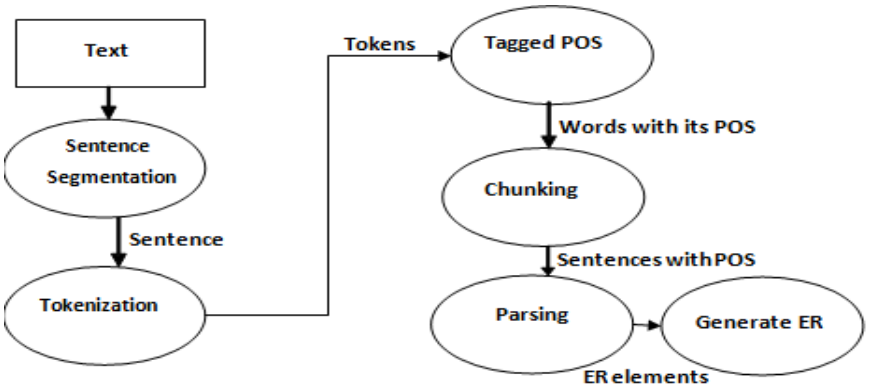

Figure 1. Generating ER from Natural Language Processing

Figure 1 describes the typical architecture for generating ER from natural language. The information extraction system begins by sentence segmentation processing, which is a morphological analysis applied to specifications followed by tokenization process. The result from tokenization process is words only. Part Of Speech (POS) process tagged each word with its abbreviations. Chunking and parsing apply multiple possible analyses on results. Parsing is the process of using a grammar to assign a syntactic analysis to a string of words forming parsing tree. Finally, information extracted from parsing tree used to generate ER diagram. Each process is described in detail in the following subsections.

3.1. Sentence Segmentation

In this step, morphological analysis is applied on the natural language text. User enters the requirement specifications in the provided workspace area. Then, analyses process is performed to determine sentence boundaries, and Split text into sentences. Usually, each sentence must end with period and this period terminates the sentence. Eliminate all non-word tokens like punctuations, removing plural suffixes in nouns, such as s, es, or ies, and converting plural entity names into singular. Figure 2 shows an example of sentence segmentation process using ER generator.

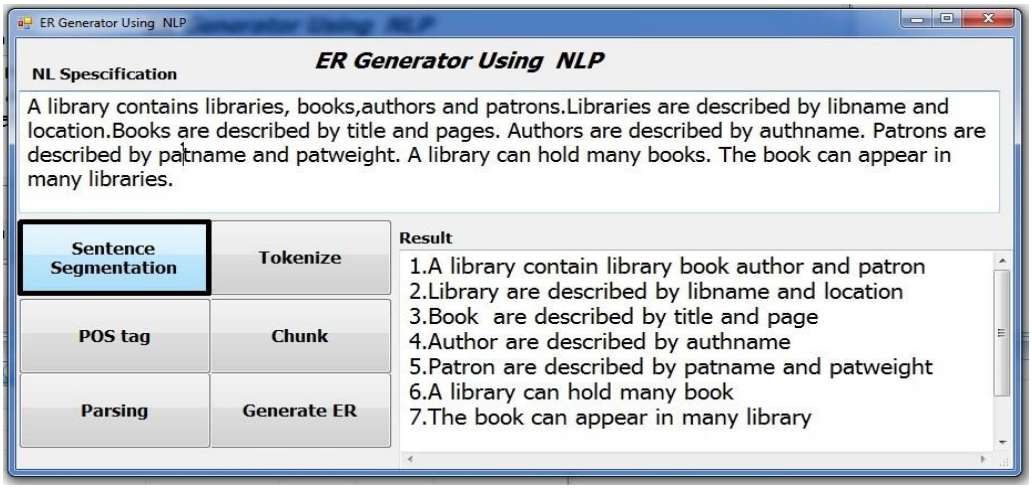

Figure 2. Sentence Segmentation Example

As shown in Figure 2, the text of requirements is written in natural language in *NL Specification* section. Requirements analysis is perform when the user press on *Sentence Segmentation* button. Basically, the process then determine the sentence boundaries, split text into separated sentences, eliminate all non-word tokens, such as punctuations, removing plural suffixes and converting plural entity names into singular. For example, in Sentence 1, contains is mapped to contain. Also, *libraries* and *authors* are mapped to *library* and *author* respectively.

3.2. Tokenization

In tokenization process, words and numbers in each sentence are identified. It is necessary to specify the sentence's components. Basically, the proposed tokenization is set to break up the given sentence into units called tokens separated by spaces. For example, the sentences "I like solving interesting problems ". The tagged sentences appear as <i> < like > < solving> <interesting> <problems>. Such implementation similar to string.split (' ') in programming language. Figure 3 shows the result of performing tokenization. Tokenization process can identify each word in user input data. However, compound words that use commas and periods add complexity. For example, a tokenizer may have to recognize that the period in "Mr. Ali" does not terminate the sentence.

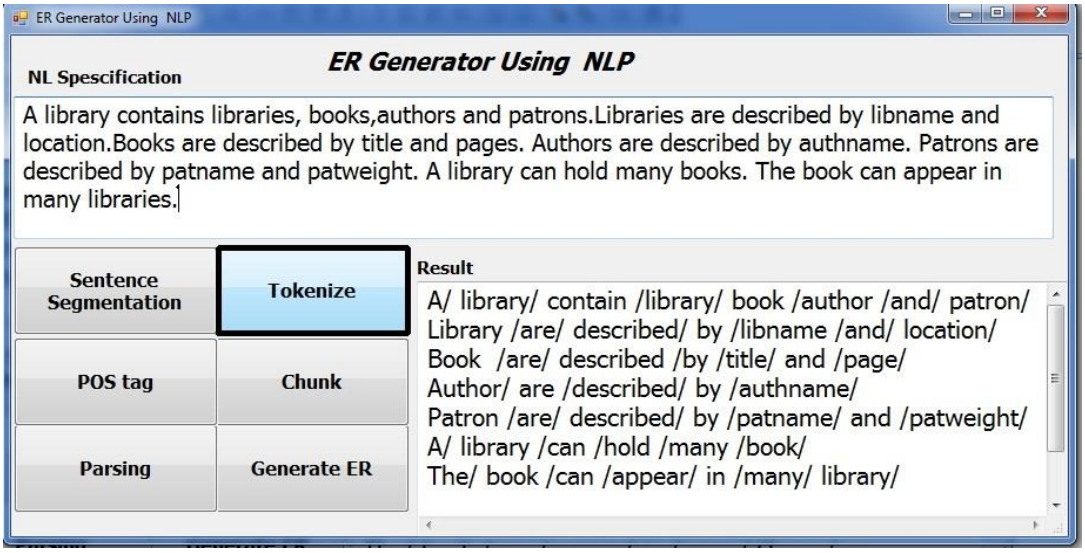

Figure 3. Tokenization Example

Tokenizing the specification as shown in Figure 3 include breaking up the given text into tokens. Tokenization process can identify each word in specification. For example, each sentence shall appear without period or comma and each word split from other words in the text.

3.3. Tagged Part Of Speech (POS)

Part Of Speech (POS) Tagging is the process of identifies a word in a text as corresponding to a particular part of speech, based on its definition and context. Table 1 summarize list of symbols and abbreviations. Word Net 2.1 [16] is useful to perform tokenization process. For example, tokenize the following sentences, "The little girl saw Ali with a crazy dog recently" is {the/ Article, little/Adjective, girl/Noun, saw/Verb,

Ali/Noun, with/Preposition, a/Article, crazy/Adjective, dog/ Noun, recently/ Adjective}. Figure 4 show the result of performing POS tagging on a given text.

Table 1. List of Symbols and Abbreviations

symbol	Abbreviation	symbol	abbreviation	symbol	abbreviation
S	Sentence	Art	Article	Adj	Adjective
N	Noun	NP	Noun phrase	Pro	Pronoun
V	Verb	VP	Verb phrase	PN	Proper noun
Adv	Adverb	Prep	Preposition	PP	Prepositional phrase

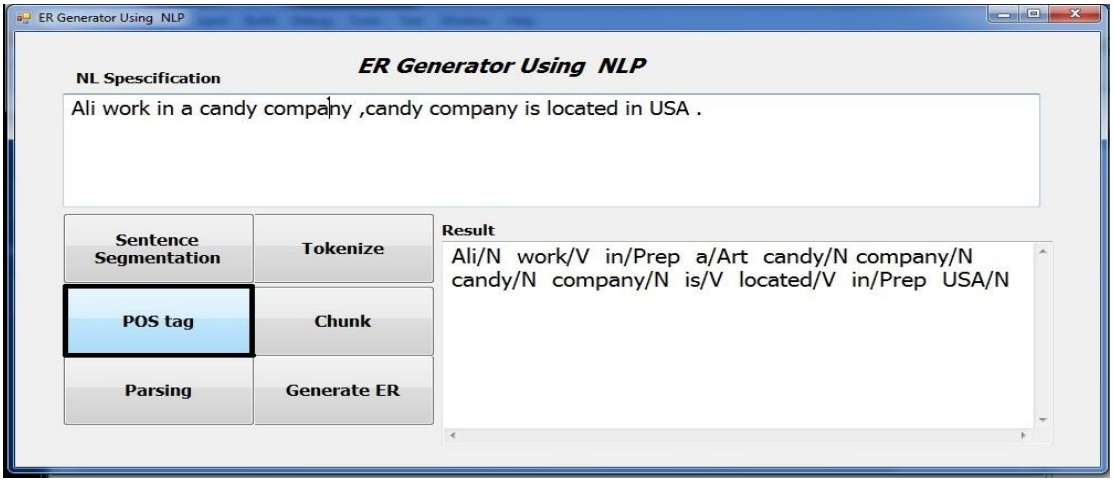

Figure 4. Perform POS Tagging

3.4. Chunking

Chunking is the process of taking individual units of information (chunks) and grouping them into larger units. Tokens of a sentence are group together into larger chunks, each chunk corresponding to a syntactic unit such as a noun phrase (NP) or a verb phrase (VP). To perform the chunking, a POS tagged set of tokens is required with tokens itself. Part of speech tagging tells whether words are nouns, verbs, adjectives, *etc*.., but it doesn't give any indication about the structure of the sentence or phrases in the sentence. Sometimes it's useful to have more information than just the parts of speech of words. Chunking usually selects a subset of the tokens together to indicate its type noun phrase or verb phrase. Figure 5 shows chucking process for the following sentences "we saw the yellow dog" is {we/NP, the yellow dog /NP}. Another example for the sentence "IBM bought Lotus" is {IBM/NP, Lotus /NP bought Lotus /VP}. Also, chunk the sentence "Ali hit the ball" is {Ali/NP, the ball /NP, hit the ball /VP}.

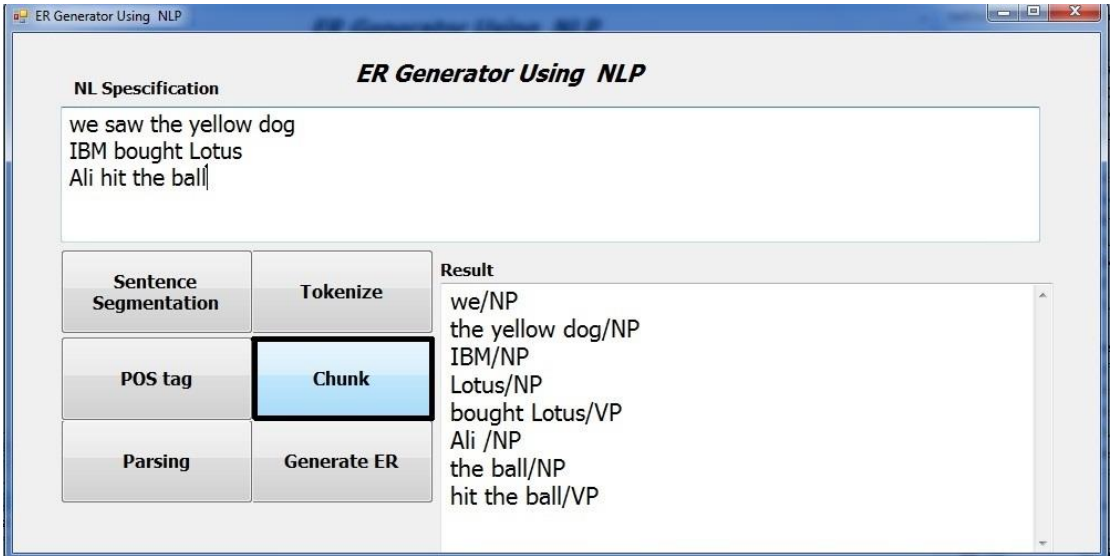

Figure 5. Perform Chunking

Chunking is a way of organizing information into familiar groups. Performing chunk process include tag tokens set with its POS. chunking usually selects a subset of the tokens together to indicate its type . Chunking is an intermediate step towards full parsing.

3.5. Parsing

Natural languages grammar is ambiguous and has multiple possible analyses. Each sentence may have many potential Parses tree. Most of them will seem easy to a human. However, it is difficult for decide which of them is in the specification. Therefore, Parsing process determines the parse tree of a given sentence. Sequences of words are transformed into structures that indicate how the sentence's units relate to each other. This step helps us in identifying the main parts in a given sentence such as object, subject…*etc*... Parsing examples are shown in Figure 6 and Figure 7. Some parsers assume the existence of a set of grammar rules in order to parse a given sentences. Following examples of such rules, however, recent parsers are smart enough to infer the parse trees directly using complex statistical models [17]. Parsing analysis will be able to extract nouns that are playing the role of entities or attributes, and extract verbs that act as a relationship between entities. Also, cardinalities and multiplicities information may be extracted from determiners, adjectives, model verbs and quantifies. This paper used Memory-Based Shallow Parser (MBSP) [8] as parser method. MBSP is a text analysis system provides tools for tokenization, sentence splitting, part of speech tagging, chunking and relation finding.

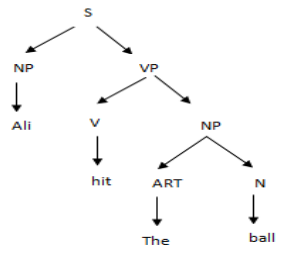

Figure 6. Parser Tree for the Sentence "Karak" is Located in Jordan

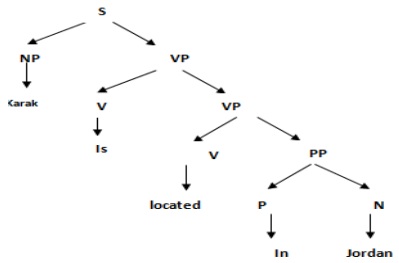

Figure 7. Parser Tree for the Sentence "Ali Hit the Ball"

The proposed methodology based on a set of identification rules that combine different concepts from other works as follows:

Rule 1: identify entities

1. A common noun may indicate an entity type [5, 9].
2. A proper noun may indicate an entity [5, 9].
3. in case of consecutive nouns existence, check the last noun, If it is not one of the words in set J where J= [number, no, code, date, type, volume, birth, id, address, name], it may be an entity type otherwise it may indicate an attribute type [22].

4: A gerund may indicate an entity type [5].

5: a specialization's relationship "A is a B" sentence's structure can relate two nouns [23].

6: A noun such as "database", "record", "system", "information", "organization" and "detail" may not be considered as a candidate for an entity type because it shows the business environment [22].

7: ignored every proper noun such as (Location name, Person name …*etc*..) [21].

Rule 2: Identify attributes

Attributes are nouns mentioned along with their entity, it may proceeded by the verbs has, have, or includes which indicate that an entity is attributed with a property. For example, in "employee has id, name, and address", employee is detected as an entity, and name, id and address are detected as attributes. Here some rules that identify attributes in specifications.

1. Noun phrase with genitive case may indicate an attributes [9].
2. If a noun is followed by another noun and the latter one belongs to set S where S= [number, no, code, date, type, volume, birth, id, address, name], this may indicate that both nouns are an attribute else it may be an entity [22].

3: A noun such as "vehicle no", "group no", "person id" and "room type" refer to an attribute [24].

4: The possessive case usually shows ownership it may indicate attribute type [9].

4: A noun phrase such as "has/have" may indicate attribute types [24].

Rule 3: Identify relationships

The main verb that occurs between two entities is more likely to be a relationship. Two entities can be separated by main verb only, by main verb and an auxiliary verb, or main verb and modal verb. For example, in {The bank is branched into many branches}, Branched is detected as relationship.

1: A transitive verb can indicate relationship type [5].

2: A verb followed by a preposition such as "by", "to", "on" and "in" can indicate a relationship type [9].

3: if a verb is in the following list {include, involve, consists of, contain, comprise, divided to, embrace}, this indicate a relationship of aggregation or composition [21].

4. An adverb can indicate an attribute for relationship [5].
5. A verb followed by a preposition such as {on, in, by, to} could be a relationship. For example, {Persons work on projects.} Other examples include {assigned to} and {managed by} [22].

Rule 4: Identify Primary Key.

1. Adverb (uniquely) indicates PK of an entity [18].
2. If the sentence is in the form of {"Subject" + "Possessive verb" + "Adjective" +"Object"}, then the object is a key attribute [25].

3.6. Generate ER

The ER generator is a rule-based system that identifies ER relationships, ER entities and ER attributes [18]. Once all words have been assigned to its ER element type, relevant information consisting of which words are entities, relationships, cardinalities and attributes are stored in text files. These text files are then used to generate ER diagram. Figure 5 show the prototype editor for the ER generating process. Currently, the prototype is in design stage. ER generator is easy to use and understand. However, the ER generator tool aims to provide minimal human intervention during the process. Figure 8 show the E-R diagram for library management system example.

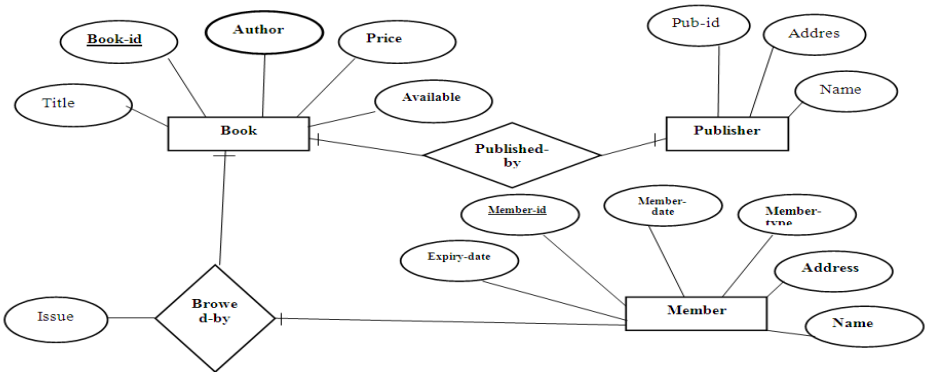

Figure 8. E-R Diagram for Library Management System

4. Limitation

Linguistic variation (Incomplete Knowledge) and ambiguity are the main problem in using NL. Part of- speech tagging also is harder than just having a list of words and their parts of speech, some parts of speech are complex or unspoken .difficulties of accessing information in given text is due to the complexity of natural language. Technologies of NLP are still a way from being able to understand information from unrestricted text. The heuristics approach suitable for small application domains not large one. Appling NLP in specific domain problems is more efficient and could make significant progress. Also there is no standard approach for automatically recognize objects and classes from English sentences. Moreover, the analysis process is the most critical and difficult tasks because most of input scenario is in natural language such as English or Arabic [20].

Generating Multi-document Text related to domain problem using NLP is more difficult from generating a single document [19]. Also the most challenging task is to be able to parse Arabic or Chinese language. Such language is different linguistic properties compared to English. Generally, Natural language processing is successful in meeting the syntax challenges. But it still has to go a long way in the areas of semantics and pragmatics.

5. Conclusion

Entity relationship modeling is a high level data modeling technique that helps designers create useful and accurate conceptual models. Much research has attempted to apply Natural Language Processing (NLP) to extract knowledge from requirement specifications. Heuristics based rules is used to parse the specifications. This approach pays particular attention to natural language processing techniques such as tokenization, tagging POS, chunking and parsing based on syntax heuristics rules.

Parsing results would be words with its Part Of Speech (POS); this result fed into ER generator to identify suitable data modeling elements according to heuristics based rules. This approach gives the Database designer an overview of the output of natural language processing. Moreover, provides designer with detailed modeling information that help them during database design. As future work, extend using of NLP to have semantics analysis rather than structural analysis to infer new things such as composite attributes, cardinalities, weak attributes *etc*... In addition, raise the using of artificial intelligent techniques (AI) such as support vector machine (SVM) and neural network for better understand of requirement specifications.

References

[1] S. Geetha, and G. A. Mala, "Automatic Relational Schema Extraction from Natural Language Requirements Specification Text", Middle-East Journal of Scientific Research, vol. 21, no. 3, **(2014)**, pp. 525-532.
[2] F. Hogenboom, F. Frasinca and U. Kaymak, "An Overview of Approaches to Extract Information from Natural Language Corpora," Information Foraging Lab, **(2010)**, p. 69.
[3] C. Andrews, "A Natural Language Interface Using First-Order Logic" A Major Qualifying Project Report: Submitted to the Faculty of (Doctoral dissertation, WORCESTER POLYTECHNIC INSTITUTE). **(2005)**.
[4] E. Castro, C. Dolores, M. Martinez and A. Iglesias, "Integrating Intelligent Methodological and Tutoring Assistance in a CASE Platform: The PANDORA Experience", Informing Science, **(2002)**, pp. 261-269.
[5] P. Chen, "English Sentence Structure and Entity Relationship Diagrams", International Journal of Information Science, vol. 29, **(1983)**, pp. 127-149.
[6] I. Y. Song, M. Evans, and E. K. Park, "A comparative analysis of entity-relationship diagrams", Journal of Computer and Software Engineering, vol. 3, no. 4, **(1995)**, 427-459.
[7] R. J. Abbot, "Program Design by Informal English Descriptions," Communication of the ACM, vol. 26 no. 11, **(1983)**, pp. 882 – 894.
[8] E. Buchholz, H. Cyriaks, A. Düsterhöft, H. Mehlan, B. Thalheim, "Applying a Natural Language Dialogue Tool for Designing Databases," International Workshop on Applications of Natural Language to Databases (NLDB'95), **(1995)**, pp. 119-133.
[9] A. M. Tjoa and L. Berger, "Transformation of Requirement Specification Expressed in Natural Language into an EER Model," Proceedings of the 12th International Conference on Entity Relationship Approach, Springer Verlag, New York, **(1993)**, pp.127-149.
[10] H. M. Harmain and R. Gaizauskas, "CM-Builder: An Automated NL-based Case Tool", 15th IEEE International Conference on Automated Software Engineering (ASE'00), **(2000)**, pp. 45-54.
[11] N. Omar, P. Hanna, and P. McKevitt, "Semantic analysis in the automation of ER modeling through natural language processing", Computing & Informatics, ICOCI '06. International Conference on, IEEE, **(2006)**, pp. 1-5.
[12] F. Gomez, C. Segami and C. Delaune, "A system for the semiautomatic generation of E-R models from natural language specifications", Data and Knowledge Engineering, vol. 29, no. 1, **(1999)**, pp. 57-81.
[13] L. Mich, NL-OOPs, "From Natural Language to Object Oriented Using the Natural Language Processing System LOLITA", Natural Language Engineering, **(1996)**, pp. 161-187.
[14] A. Montes, H. Pacheco, H. Estrada, O. Pastor, "Conceptual Model Generation from Requirements Model: A Natural Language Processing Approach", Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, London, UK ISBN: 978-3-540-69857-9, **(2008)**, pp. 325-326.
[15] K. Daghameen and N. Arman, "Requirements Based Static Class Diagram Constructor (Scdc) Case Tool", Journal of theoretical & Applied Information Technology, Islamabad Pakistan, **(2010)**, pp. 108-114.
[16] P. R. Kothari, "Processing Natural Language Requirement to Extract Basic Elements of a Class", ISSN.-2249-0868 Foundation of Computer Science PCS, New York, USA, vol. 3, no. 7, **(2012)**.

[17] D. M. Bikel, "On the parameter space of generative lexicalized statistical parsing models (Doctoral dissertation", University of Pennsylvania), **(2004)**.
[18] L. A. Al-Safadi, "Natural Language Processing for Conceptual Modeling", JDCTA, vol. 3, no. 3, **(2009)**, pp. 47-59.
[19] N. Madnani, "Getting started on natural language processing with Python", Crossroads, vol. 13, no. 4, **(2007)**, pp. 5-5.
[20] M. E. Elbendak, "Framework for using a Natural Language Approach to Object Identification". In RANLP, **(2009)**, pp. 23-28.
[21] H. Hatem and W. B. Abdessalem, "From user requirements to UML class diagram", arXiv preprint arXiv, vol. 1211, no. 0713. **(2012)**.
[22] N. Omar, P. Hanna, and P. Mc Kevitt, "Heuristics-based entity relationship modeling through natural language processing", in Fifteenth Irish Conference on Artificial Intelligence and Cognitive Science, , (AICS-04), **(2004)**, p. 302-313.
[23] S. H. Sebastian, "Link.: English Sentence Structures and EER Modeling", In Proceedings of APCCM'2007. p. 27-35.
[24] V. G. Storey, "View Creation: An Expert System for Database Design, ICIT Press, **(1988)**.
[25] S. Geetha, G. S. A. Mala, "Automatic database construction from natural Language requirements specification text", ARPN Journal of Engineering and Applied Sciences, ISSN 1819-6608, vol. 9, no. 8, **(2014)**, p. 1260-1266.
[26] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to word net: An on-line lexical database", International journal of lexicography, vol. 3, no. 4, **(1990)**, 235-244.

Authors

Eman Btoush, (emanbtoush26@gmail.com) received the B.S. in computer science From Mutah University, Mutah, Jordan, in 2002. She is currently a M.S. student of computer science Department. Mutah University, Jordan.

Mustafa Hammad, is an Assistant Professor at Information Technology department in Mu'tah University, Al Karak - Jordan. He received his PhD. in computer science from New Mexico State University, USA in 2010. He received his Masters degree in computer science from Al-Balqa Applied University, Jordan in 2005 and his B.Sc. in computer science from The Hashemite University, Jordan in 2002. His research interest is Software Engineering with focus on static and dynamic analysis and software evolution.

