A generic framework for requirements elicitation from informal descriptions

S. Murugesh[1], Dr. A. Jaya[2]

[1]Research Scholar, B.S. Abdur Rahman University, Chennai 600048.

[2] Professor & Research Supervisor, Department of Computer Applications, B.S. Abdur Rahman University, Chennai 600048.

Abstract— Unstructured documents refer to documents that contain information that either does not have a predefined data model or is not organized in predefined manner i.e. informal descriptions. Unstructured documents are heavy with lots of text information along with dates, numbers and facts as well. Common techniques for structuring text involves manually tagging with metadata, manual annotation is difficult because of number of issues, like the annotator must be familiar with the domain of the document of interest, preliminary training and guidelines are necessary for a particular annotation task as well as the process is time consuming and error-prone. Examples of unstructured data are books, journals, documents, body of e-mail, notes, data from technical surveys. This paper describes some of the difficulties in working with unstructured text collections and methods to overcome them. To obtain some competitive advantage in processing unstructured data, attributes have to be generated not only for single terms but for combined terms also.

***Index Terms*— Unstructured Documents, Information Extraction, Requirements elicitation, Natural Language Processing.**

I. Introduction

Databases have tremendously grown in every area of human activity, new and powerful tools are the need of the hour for discovering useful knowledge from the data. Nowadays lot of information is available in form of text, in e-mails, manuals, suggestions, complaints and so on. This unstructured information lacks an internal structure unlike the traditional databases. When information required to develop a system are specified in an unstructured format, to make productive use of all this information we apply Text Mining and shallower natural language processing techniques and generate a requirements model [5]. As far as system development is concerned most of the software requirements data available to software engineers are expressed in natural language and 90% of the data are unstructured. There is a lack of a formal approach with high expressiveness close to that of natural language that drives system analysts to use their own informal ways to gather requirements [3]. This paper presents the application of methods for transformation of requirements specifications expressed in natural language into semi-structured specifications [7]. Requirements elicitation is the process of discovering system requirements through consultation with stakeholders, from system documents, domain knowledge, or any other means of information (Ratchev et al., 2003).

Many existing off-the-shelf technologies, such as IBM Rational Requisite Pro (IBM, 2007), assist requirements engineers in authoring and organizing requirements specifications, but do not offer facilities for the analysis of the documentation provided by stakeholders and the extraction of requirements from it. There is a need to develop tools to support the entire elicitation process. Preliminary description of the system which is informal and expressed in natural language is later refined and revised based on the application domain. The modeling stage will be very difficult if this description is completely unstructured. This scenario description is to be refined later by domain experts and requirements engineers. Then the requirements are expressed graphically, which is the final outcome.

Instead of using linguistic information to infer annotations, we use instances of concepts from a semantic model of an application domain through the use of pattern-based rules specific to such a domain, to analyze different types of input text.

II. Literature Review

R.J Abbot proposed the idea that natural language text can be used for object oriented software modeling, classes or objects can be identified from nouns and methods of classes can be identified from verbs in a sentence. [5]. The major objective of object-oriented analysis is to identify Natural Language concepts that can be modeled in the form of object-oriented concepts. Mohammed Alawairdhi, suggested a scenario based approach for requirements elicitation for software systems complying with the utilization of pervasive computing technologies [8]. Marinos G. Georgiades recommened A novel software tool for supporting and automating the requirements engineering process with the use of natural language. Carlos Mario Zapata used templates for gathering requirements [10]. Syed Ahsan used questionnaires, questions were posted to customers through mobile devices to elicit requirements [11]. Huafeng Chen uses Structured Natural Language (SNL) (Subject, Predicate, Object Syntax) for text based requirements preprocessing using NLP techniques [13]. Marinos G. Georgiades uses a “template” and users should fill in the form [14]. Neil. W. Kassel follows question and answer approach to collect specifications from users [15].

III. PHASES IN REQUIREMENTS ELICITATION

In this paper we propose an approach to elicit requirements from informal specifications, the informal description has to be analyzed and information relating to the following has to be extracted

i. Actors : Relevant actors i.e. agents and roles have to be identified along with their objectives and capabilities.

ii. Goals : Goals have to be analyzed and refined into subgoals and tasks.

iii. Relations : Relations among actors have to be identified.

iv. Privacy : Privacy of actors if any, have to be identified if there is an explicit request or if it is implied.

In general the concepts to be identified from the informal descriptions of the system to be developed are classified as follows,

i. ***Basic Entity*****,** elements which form the basis of the scenario and can exist independently from any other entity.

ii. ***Complex Relationship*** includes a concept that binds basic entities together.

iii. ***Specific Entity Role*** implies the interpretation of the basic entities involved in a complex relationship on the basis of their role.

There is a need to develop architecture with the following phases as shown in Fig 1

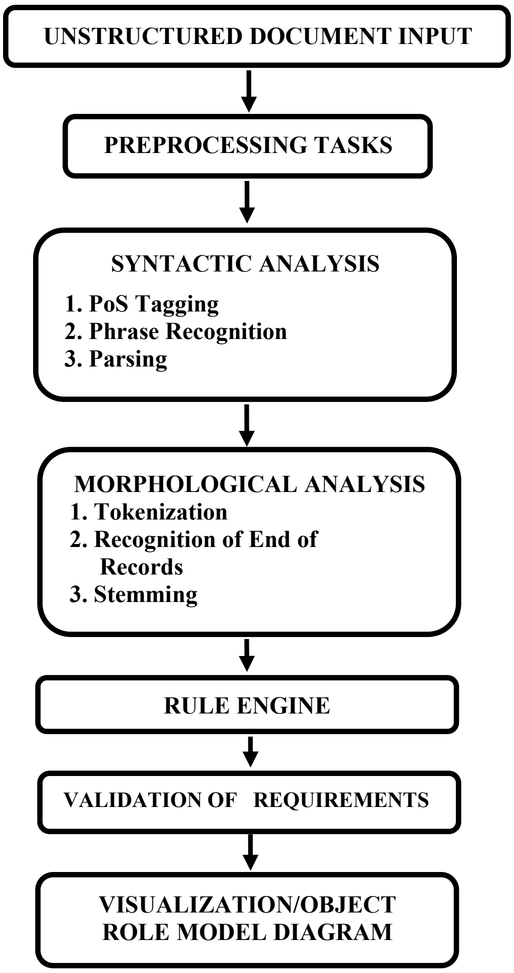


Fig 1. Requirements Elicitation from informal Descriptions

1. PREPROCESSING TASKS

The input document has to be preprocessed to remove noise, for stop word removal and for tokenization. In NLP studies, it is conventional to concentrate on pure analysis or generation while taking the basic units, namely words, for granted. Without these basic units clearly segregated it is impossible to carry out any analysis [6]. Use of Instance Filtering techniques, by providing a list of stop words [1]. First the input is tokenized and all individual words and symbols are separated. **Parse** the input document and break down into constituents, develop a parse tree that consists of features such as document, paragraph, phrase and word. Fig 1. presents the taxonomy of text preprocessing tasks for software requirements elicitation.[12]

A text document that contains the software requirements in unstructured form is given as input, after preprocessing, the document is represented in an intermediate form, suitable for applying the core mining operations.

A. Morphological Analysis

The first step in text preprocessing is the morphological analysis. Morphological or structural analysis is the process of breaking down morphologically complex words into their constituent morphemes (word meaning parts). Morphology deals with the smallest, useful unit of a document. Characters are the smallest unit.

i. **Tokenization**

The first step of Morphological Analysis is the tokenization. The aim of tokenization is the exploration of the words in a sentence. Textual data is only a block of characters at the beginning. For example, the sentence " ATM machine prints receipts" is tokenized as follows

<sentence>

<word>ATM</word>

<word>machine</word>

<word>prints</word>

<word>receipts</word>

</sentence>

ii. **Lemmatization and Stemming**

To reduce the dimensionality of the documents, the set of words describing the documents can be reduced by filtering and lemmatization or stemming methods. Filtering methods remove words from the dictionary and thus from the documents. A standard filtering method is stop word filtering. Lemmatization maps the verb forms to the infinite tense and nouns to the singular form. Stemming builds the basic form of words. Stemming reduces all the inflectional and derivational variants of words to a common form called the *stem.*

Certain suffixes and prefixes for removal during stemming are

Suffixes: ly, ness, ion, ize, ant, ent, ic, al, able, ary, ing.

Prefixes: de, en, in, pre, post, un, over.

After the stemming process each word is represented by its stem.

iii**. Recognition of End of Records**

An algorithm should search for end of records like ".", "!". The results of this step are sentences which can be treated as one unit.

B. Syntactic Analysis

The purpose of syntactic analysis is to determine the structure of the input text. This structure consists of a hierarchy of *phrases*, the *smallest* of which are the basic *symbols* and *largest* of which is the *sentence*. It can be described by a tree with one node for each phrase. Basic symbols are represented by leaf nodes and other phrases by interior nodes. The root of the tree represents the sentence. A sentence in English language contains nouns, verb, adverbs and other parts. Some parts are more valuable than others, for instance for requirements elicitation, nouns, pronouns and verbs are more valuable than others.

The syntactical analysis are divided into three subcategories

- Part of speech tagging
- Phrase recognition &
- Parsing

i. **Part of Speech Tagging**

The recognition of the elements of a sentence like nouns, verbs, adjectives, prepositions etc is realized through part of speech tagging (POS). In software requirements elicitation ***nouns*** represents ***tangible objects or entities*** and ***verbs*** represent the ***actions/events/relations*** that cause state changes on these entities. Entities and relations are the building blocks of any information system.

ii. **Phrase Recognition**

The phrase recognition is closely related to part of speech tagging. The phrase recognition (PR) caters for locating of groups of words, the phrases. PR is needed to keep relations between word groups, which would lose their meaning if disjoined. Phrases are of different types

- Prepositional phrase (eg. In the table)
- Noun phrase (eg. The king of India)
- Verb phrase (eg. Do business)
- Adjectival phrase (eg. Big machine)
- Adverbial phrase (eg. Very fast)

iii. **Parsing**

Parsing produces a parse tree for a sentence. A parse tree represents the relation of each word in the sentence to all the other words and also its function in the sentence i.e subject, object etc. Parsing is the process of structuring a sentence with the given grammar. The sentences are fractionalized into grammatical units. The structure of a sentence is represented in a tree structure. The tree structure is very useful for recognition of groups of words.

2. RULE ENGINE

Rule Engine is to be developed with two separate modules that handle Parse Tree and Dependency Graph, the rule engine is to implement algorithms to extract relationships.

The output of this phase is

a) Identification of candidate classes (nouns in Natural Language).
b) Establishing associations (capturing verbs to create association for each pair of classes).
c) Identification of class attributes. (Adjectives in Natural Language)
d) Identification of class operations (Verbs in Natural Language)

3. MAPPING

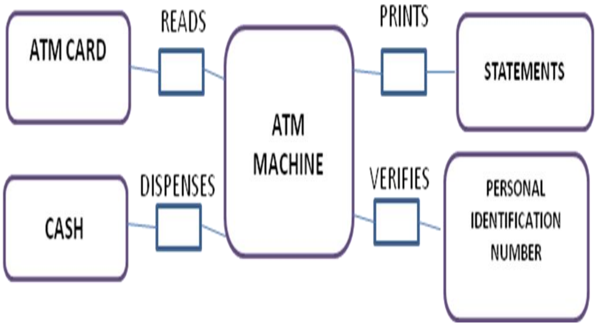
Mapping between the extracted information and the stored knowledge base specific to the domain of the system is to be developed. Mapping with a domain specific background knowledge source (Domain Ontology) is done to filter out invalid requirements. Corpus plays a significant role in modern Natural Language Processing (NLP). Corpus can be divided into two categories according to its purpose. One is general corpus including words which are not limited to a domain, the other is domain specific, which is similar with domain ontology, ontology is a formal specification of a shared conceptualization. Domain ontology usually defines concepts and their relationships in a certain domain. The main difference between corpus and ontology are that ontology includes more complex relationships of concepts and usually has the ability of reasoning, whereas corpus includes only simple relationships between words such as synonym and antonym [4].

4. VISUALIZATION

To visualize the document (containing requirements) as a graph using visualization service from SVG visualization, HP labs that utilizes Google Web Toolkit.

The identified Entities & Relations can be represented using an Object Role Model. (Visualization)

Example



To adapt this architecture to a new domain, it is essential to supply the components specific to that domain, i.e the annotation schema.

Annotation granularity should be at the following levels of granularity,

Sentence level in which descriptive statements of complex relationships are identified,

Word level in which, all noun phrases that describe basic and specific entities of complex relationships are identified.

Once the basic entities, their roles and relations are identified the framework is ready for requirements elicitation.

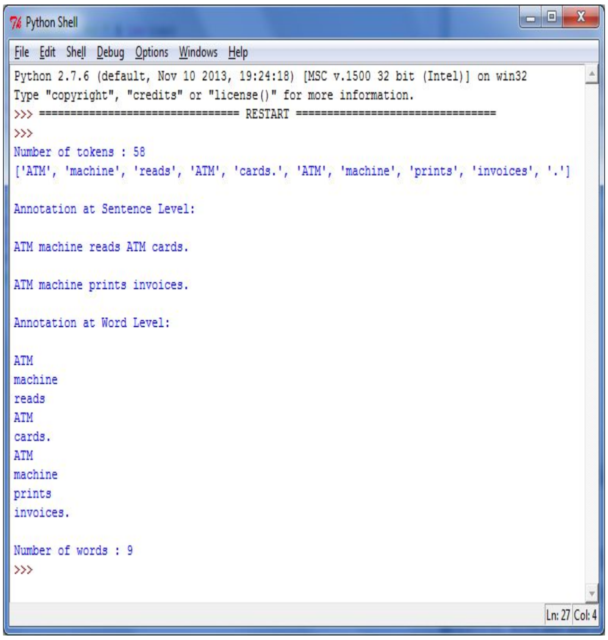
The extraction process should be organized as follows,

1. Identification of document structure, i.e. the recognition of the *scenario description* and structure it according to the document grammar.
2. *Markup of basic entities*, it is at this phase the document needs to be annotated with tags. The objective of Information Extraction is to identify a set of relevant domain-specific classes of entities and their relations. Use of stop word filters, a class of Instance Filtering (IF) techniques that remove from the data set all instances (i.e) tokens belonging to a list of stop words, as a method to identify tokens exclusively [1].
3. *Markup of complex relationships*, this phase identifies the complex relationships based on the domain and input driven patterns.
4. *Identification of entity roles*, based on the position in a statement that describes the relationship between the entities, the semantic roles are assigned.

So annotations for the concepts of interest at both word and sentence levels are generated.

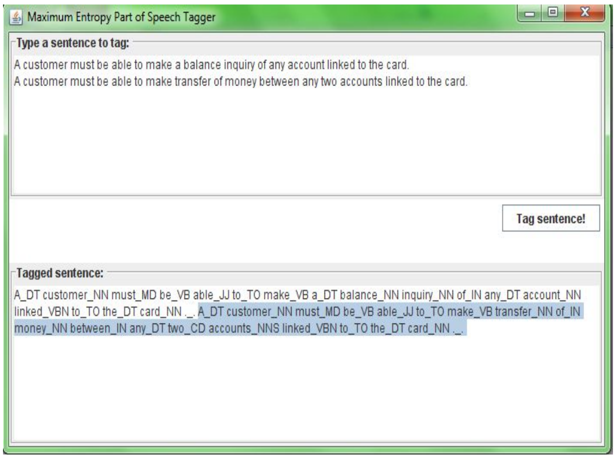
IV. ADVANTAGES OF THE APPROACH

The objective of the proposed approach is to make the requirements elicitation faster and complete. The system reduces the job of the analyst to manually go through the voluminous unstructured text data and arrive at requirements. First phase have been implemented, currently the work is developing rule engine and on developing ontology (domain-specific and a language ontology) for mapping the extracted information and the stored knowledge. Fig.2 shows the output of sample data at Sentence & Word level annotation implemented using Python. Fig 3 shows the output of sample data for the Part-of-Speech tagging.



```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Number of tokens : 58
['ATM', 'machine', 'reads', 'ATM', 'cards.', 'ATM', 'machine', 'prints', 'invoices', '.']

Annotation at Sentence Level:

ATM machine reads ATM cards.

ATM machine prints invoices.

Annotation at Word Level:

ATM
machine
reads
ATM
cards.
ATM
machine
prints
invoices.

Number of words : 9
>>>
```

Fig. 2- Sentence & Word Level Annotation using Python & NLTK



```
Type a sentence to tag:
A customer must be able to make a balance inquiry of any account linked to the card.
A customer must be able to make transfer of money between any two accounts linked to the card.

Tagged sentence:
A_DT customer_NN must_MD be_VB able_JJ to_TO make_VB a_DT balance_NN inquiry_NN of_IN any_DT account_NN
linked_VBN to_TO the_DT card_NN ._. A_DT customer_NN must_MD be_VB able_JJ to_TO make_VB transfer_NN of_IN
money_NN between_IN any_DT two_CD accounts_NNS linked_VBN to_TO the_DT card_NN ._.
```

Fig. 3 - Output of Part of Speech Tagging

V. CONCLUSION

The proposed methodology provides a framework for automated gathering of requirements from voluminous unstructured text documents with minimum amount of time when compared to manual elicitation which is difficult, time-consuming and error prone.

REFERENCES

[1] Alfio Massimiliano, Claudio et al, "Instance filtering for Entity Recognition", SIGKDD Explorations, volume 7, Issue 1- Page 11, 2002.

[2] H.M. Harmain, R. Gaizauskas, "CM Builder: An automated NL-based CASE tool", IEEE, 2000.

[3] Marinos G Georgiades & Andreas S. Andreou, "A novel methodology to formalize the requirements engineering process with the use of natural language", IADIS International Conference Applied Computing 2010.

[4] Huafeng Chen, Keqing He, Peng Liang, Rong Li, "Text-based requirements preprocessing using nature language processing techniques", IEEE 2010.

[5] R.J. Abbott, "Program Design by Informal English Descriptions", Communications of the ACM, Nov. 26(11), 1983, pp. 882-894.

[6] Jonathan J Webster & Chunyu Kit, "Tokenization as the initial phase in NLP", ACTES DE COLING-92,NANTES, 1992.

[7] Andrew McCallum," Distilling Structured data from unstructured text", ACM QUEUE, November 2005.

[8] Mohammed Alawairdhi, "A scenario based approach for requirements elicitation for software systems complying with the utilization of pervasive computing technologies", IEEE Annual Computer Software and Applications Conference Workshops.

[9] Marinos G. Georgiades, "A novel software tool for supporting and automating the requirements engineering process with the use of natural language", International Journal of Computer Science & Technology.

[10] Carlos Mario Zapata, "Computational Linguistics for helping requirements elicitation, a dream about automated software development", IEEE Association for Computational Linguistics.

[11] Syed Ahsan, "An approach to support and partially automate requirements engineering activities", IEEE 8th Conference on Computer & IT.

[12] S. Murugesh, "Role of preprocessing tasks in Text Mining for software requirements elicitation", International Journal Of Research In Advance Technology In Engineering (IJRATE) Volume 1, Special Issue, October 2013.

[13] Huafeng Chen, "Text based requirements preprocessing using NLP techniques", IEEE International Conference on Computer Design & Applications.

[14] Marinos G. Georgiades, "A requirements engineering methodology based on natural language syntax & semantics", IEEE International Conference on Requirements Engineering.

[15] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 4, pp. 570-578, July 1993.

[16] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34-39, Jan. 1959.

[17] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, and M. Miller, "Rotation, scale, and translation resilient public watermarking for images," *IEEE Trans. Image Process.*, vol. 10, no. 5, pp. 767-782, May 2001.