A Step towards Software Requirements Elicitation from Unstructured Documents

S. Murugesh
Research Scholar
B.S. Abdur Rahman University, Chennai

A. Jaya, Ph.D
Professor & Research Supervisor,
Department of Computer Applications,
B.S Abdur Rahman University, Chennai,

ABSTRACT
Good requirements have to be gathered for software development. Most of the requirements from the client of a proposed software system are available in informal or unstructured documents. Most often requirements are ill defined. Deficiencies in software requirements are mostly identified only after deployment.

Most software requirements data available to software engineers are expressed in natural language & 90% of data are unstructured. Customers might not be able to provide all the requirements since they are not sure of what they want. Also they are unable to state the requirements due to incomplete knowledge of the applications functionality.

This paper proposes methods to derive formal specifications from informal or unstructured documents using techniques from natural language processing & text mining. The purpose of using natural language processing techniques is that sentences should be understood without human intervention and to understand the document in a way how human beings interpret & understand. The objective is to elicit the requirements from informal documents without using any strict template & to keep human intervention to the minimum. A parser or POS tagger is to be used to find the nouns, verbs. Transform free form text into an Intermediate Form (IF), Extract Entities & Relations then cluster Entities & Relations. Natural language transcripts contain lots of user requirements specified in their own language without any technicalities. For program development these informal requirements have to be derived, studied for their feasibility and then built into the software which would satisfy the requirements of the customer.

Capture the requirements for the system and generate specifications from the captured requirements so that the software engineers and customers can understand them.

General Terms
Natural Language Processing, Text Mining.

Keywords
Unstructured Documents, Information Extraction, Natural Language Processing, Text Mining, Requirements Engineering.

1. 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. Almost all documents on paper are now available in electronic formats. Since electronic format facilitates for safe storage and consumes less space at the same time provides quick access to the documents stored. Text documents are very large and there is lot of hidden patterns or relations in the data. Moreover since text data is not available in numerical format, statistical methods cannot be applied directly to analyze them. This unstructured information lacks an internal structure unlike the traditional databases. To make productive use of all this information we apply Text Mining techniques. 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.

Requirements Engineering involves Requirements Elicitation, Requirements Analysis and Requirements Specification. Software requirements elicitation may be the most important area of requirements engineering, since any errors produced at the requirements gathering stage and if they are undetected they get propagated to the later stages of software development and sometimes identified only after implementation. Software engineers also spend less time on this task. Errors are produced in requirements since customers do not know precisely what they need or they do not know how to specify or do not know that functions the application to be developed should possess. So the presence of domain experts, customers and software engineers are vital. In this paper we present an approach to partially automate the requirements elicitation process.

We have proposed an integrated framework using natural language processing techniques for preprocessing and text mining for the core mining operations. Structured data are stored in a pre-defined data model. Whereas unstructured data are available as freeform text and it is difficult to extract, query or search for information. In requirements elicitation there is a need for the participation of domain experts, since it is not possible to have them always, as an alternative a

domain-specific background knowledge source is used. Domain knowledge is used for concept extraction and validation activities.

2. PROPOSED INTEGRATED APPROACH

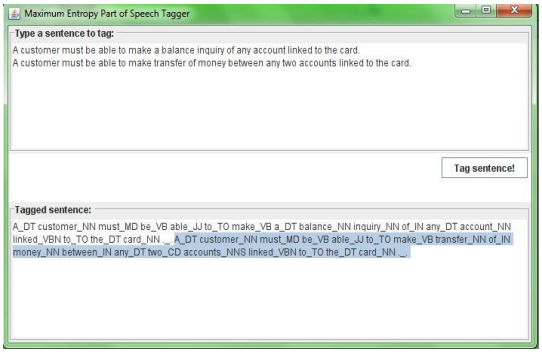
The approach encompasses the following

1. The initial inputs i.e. the unstructured document collection are processed by using Natural Language Processing techniques. This is the phase of Information Extraction wherein we extract structured information from unstructured machine-readable document. We extract Entities i.e. names of people, companies & location etc and Events [1] i.e. an activity or occurrence of interest in which entities participates.
2. The tasks include tokenization and Part of Speech (POS Tagging). Tokenization or Zoning, splits the input document into words, sentences and paragraphs. Using the Part of Speech Tagger, a token is marked with the part of speech in the sentence, such as noun or verb and it is unambiguously tagged.

We mainly perform Term & Event extraction on each document with an intention that they are likely to have meaning in the domain and then apply mining on the features extracted [2]. The system is supposed to contain two components namely Text Analysis component and Data Mining component. The Text Analysis component converts unstructured data into structured data, the Data Mining component applies data mining techniques namely clustering and association on the information extracted from the first component. These are the several steps of preprocessing applied to the data, which removes stop words, punctuations and HTML tags.

We take for example a sample text document file that contains specifications for development of an ATM system. Here output of the POS tagger which takes as input two sentences available in a document. The two sentences are

i. A customer must be able to make a balance inquiry of any account linked to the card.
ii. A customer must be able to make transfer of money between any two accounts linked to the card. The output is given below



3. Using the output from the POS tagger, Derive relevant Entities and Relations by mapping with domain specific background knowledge source, thereby filtering out the unfeasible requirements. The domain-related knowledge can enhance the performance of NLP tasks [3] and can also resolve structural ambiguities.
4. Using k-means clustering algorithm [4] the collection of tokens are partitioned into set of clusters. Basically we begin with two clusters i.e. one for noun and another for verb. The clustering algorithm is made to work on the document-term matrix.

The most prominent feature of the text documents to be clustered is their complex internal structure. To be clustered, the documents must be converted into vectors in the feature space. Each vector representing the document in this space will have a component for each word. If the word is not present in the document/sentence, the word's component of the document vector will be zero. Otherwise, it will be some positive value, which may depend on the frequency of the word in the document and in the whole document collection.

Use of Document-term matrix model to represent the given document would facilitate quicker execution of the clustering algorithm. A document term matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. Rows of a document term matrix correspond to document and columns correspond to the terms.

The same model can be extended to represent sentences in the document.

For instance if one has the following short document
D1-S1= "ATM machine reads ATM card" D1-S2= "ATM machine dispenses cash"
Where S1 and S2 refers to sentences in document1 (D1)

Then the document term matrix is shown in Table 1

Table 1 : Document Term Matrix

D1	ATM	machine	reads	ATM	card	dispenses	cash
S1	1	1	1	1	1	0	0
S2	1	1	0	0	0	1	1

The k-means Clustering algorithm partitions a collection of vectors $\{x_1,x_2,x_3 \ldots x_n\}$ into a set of clusters $\{c_1,c_2,c_3,\ldots c_K\}$. The algorithm needs k cluster seeds for initialization. The can be externally supplied as verbs and nouns among the vectors.

The algorithm proceeds as follows

Initialization: k seeds, here 2 i.e. verbs and nouns to form the core of k clusters, every other vector is assigned to the cluster of the closest seed.

Iteration : The centroids Mi of the current clusters are computed

$$M_i = |C_i|^{-1} \sum_{x \in ci} X$$

Stopping Condition: At convergence – when no more changes occur. Each vector is reassigned to the cluster with the closest centroids. (Centroid of the cluster is the most frequently occurring term).

The output of the clustering algorithm is the set of entities and relationship among entities that denotes feasible requirements.

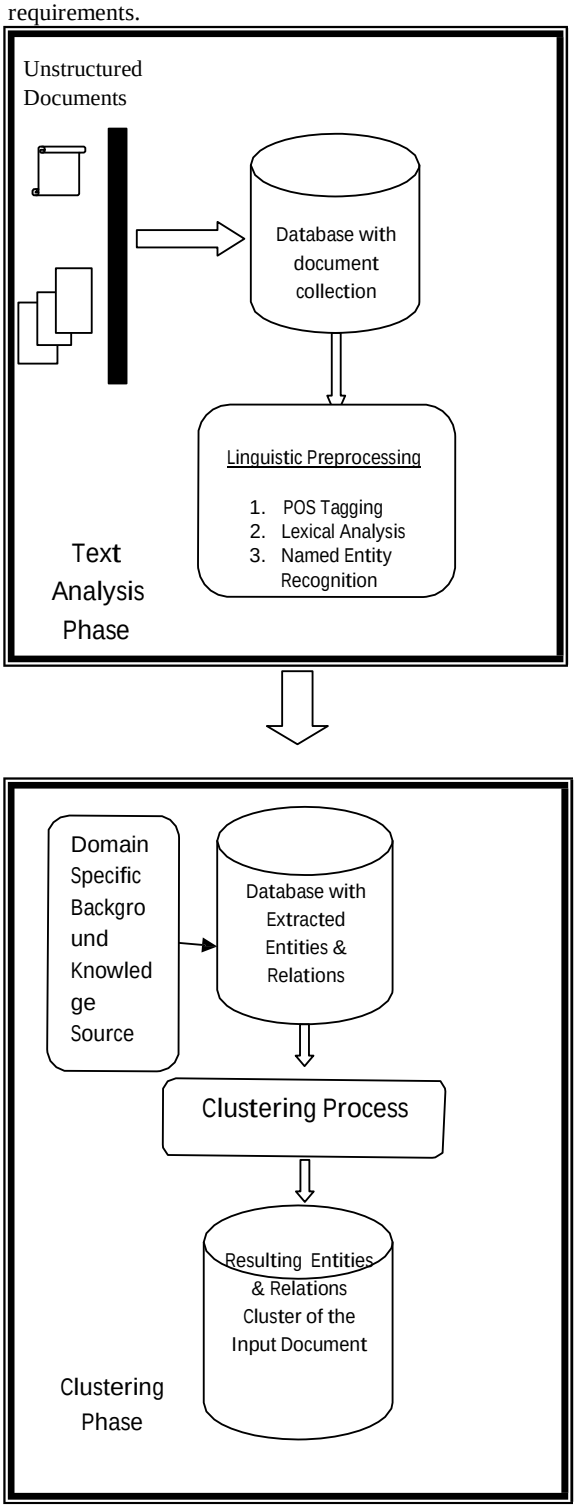


Fig 1: Functional Architecture of Text Analysis & Clustering Phase

Figure 1 shows the preprocessing phase that involves the application of natural language processing techniques and subsequently applying the core text mining operation i.e clustering on the extracted entities and relations, resulting in feasible requirements.

3. CONCLUSION

In this paper, we proposed an integrated approach for solving some problems of requirements engineering. Our approaches are hypothetical and are still researching on how to improve our present approach. Next we plan to design a detail functional architecture and implement it in a real world situation and thereafter validate.

4. REFERENCES

[1] Yu-shan Chang et al , Stanford University, Applying Name Entity Recognition to Informal Text.

[2] Marinos G Georgiades et al, IADIS International Conference Applied Computing, 2010, A Novel Methodology to formalize the requirements engineering process with the use of natural language.

[3] H. M Harmain et al, IEEE, 2000, CM- Builder: An automated NL-based CASE Tool.

[4] Liping Jing et al, International Journal of Electrical & Computer Engineering, 1:5, 2006, A text clustering system based on k-means type subspace clustering and ontology.

[5] Marinos G Georgiades et al, IEEE International Conference on Requirements Engineering, 2005, A Requirements Engineering methodology based on Natural Language Syntax and Semantics.

[6] Carlos Mario Zapata, NAACL HLT 2010 Young Investigators workshop on Computational Approaches to Languages of the Americas, California, June 2010, Computational Linguistics for helping Requirements Elicitation: a dream about automated software development.

[7] Dong Lili et al, International Symposium on Intelligence Information Processing and Trusted Computing, 2010, Research on User Requirements Elicitation Using Text Association Rule.

[8] Huafeng Chen et al, International Conference on Computer Design and Applications, 2010, Text-based Requirements Preprocessing using Natural Language Processing Techniques.