Sagar Nath

VU1F1415014

**Experiment No:1**

**Aim:** Case Study on Natural Language Processing

**Title:** Application of natural language processing in object oriented software development

**Introduction:**

Software Requirement Specification (SRS) document forms the basis of problem analysis between client and developer. SRS needs to be very specific, while serving as a basis, to proceed towards implementation of desired software. It is observed that sometimes SRS is expressed in any natural language as comprehensible by the client; but it may contain ambiguity, inconsistency, and large data from the developer's planning point of view.

Identifying major functionalities from the OO analysis point of view plays an important role in project success because they are extracted from SRS, which is written in an informal style. The use of formal languages like Unified Modeling Language (UML) have been applied to avoid the incompleteness and ambiguous problem of natural language. The problem with SRS document is that the text in natural language may be ambiguous, with many possible interpretations. The delivery of the system may affected by vagueness. Hence text in Natural language needs to be processed to get a better understanding from SRS, and with less likelihood of having errors.

In recent years, the object-oriented software development style is the accepted style by developers as the present day software development languages are object oriented in nature. Class is the one of the core parameter of Object-oriented concept. Hence, the first step in object-oriented analysis of software is to find out the classes, functions, and the attributes associated with those classes.

Natural Language Processing (NLP) combines the effect of computer science and linguistics branch that concern with the interaction between the computer and human languages. Natural Language generation systems mostly convert information from human readable form to a right kind of database.

Here, an attempt has been made to analyze the document of SRS, written in any natural language and find out the class diagram and the relationship between them. The method of finding out the candidate for class diagram is done automatically.

**Background:**

R. J. Abbott proposed an approach to analyse a particular language in Ada, based linguistic analysis of informal English. The paper develop an informal strategy and formalizing it by identifying data types, objects and operators. To derive the output as per OO concept, stress on noun and noun phrases is given. Data types are suggested by Common nouns, objects are referenced by proper noun and references. Operators are identified by verbs. using If, then, else, for, do, until, when, etc. are suggested by control structures.

A. S. Cockburn suggest an approach for designing object oriented system using the linguistic metaphors. His investigation moves in the direction of reducing adverbs and adjectives to verbs i.e., being applied to nouns. But he does not provide a model to handle such cases. The paper inform that adverbs and adjectives not only reduced to their corresponding verbs but also considered as a part of a larger framework. He used the language as a metaphorical basis. The analysis used for the structure and message of the object and naming components like class, object and methods.

Cordes and Carve made an attempts generate the object model from requirement document. For the generation of knowledge base from requirements human interaction is needed. The subsequent translation of the domain knowledge into object models is automated; thus the ambiguities are resolved. Formalized knowledge translation into object model is critical to initial requirements specification quality. Still, the requirement analysis, its process and tools try to narrow down the difference between formal software models and informal requirements. The prospect of automating more of the analysis process is certainly intriguing.

**Alternatives:**

According to the paper "Requirements validation via automated natural language parsing”, only the concern verbs and nouns are found out. The object diagram is done by analysis of these two types of words only. In that paper the attribute for a class is only guessed by a person having the knowledge about the domain and that is being implemented. But in the present case study, the total input text is analyzed and different POS are tagged to each word. Like paper "Requirements validation via automated natural language parsing”, in this case study nouns and verbs are found out and they are the candidates for class name and its member function. But for class attributes, the words of type adjective and adverb are the best candidates as per paper Program design by informal English descriptions" , which is also being find out by the program used in this paper. So, if the performance of the two papers are compared, the paper "Requirements validation via automated natural language parsing” only provides the required information and rest information is retrieved manually; but in the present case study,, after finding the required information few additional information are also provided which may be helpful in many other instances.

**Methodology:**

The main job of NLP system is to find out grammar, lexicon, semantic and pragmatic which describes natural language syntax, word’s lexical information, sentence's’ literal meaning and sentences’ non-literal meaning respectively. Parser also exist which break sentences into parts and generate tree structure.

There are many problems associated to the processing of unrestricted NL in SRS. These are

* The entry for all words are not present which the system may encounter and also the meaning is not convincing. First of all, the unknown words of the sentences are difficult to process. Secondly, the final analysis of the sentence may not give correct result.
* During the process of parsing syntax tree is generated. There might be case that for a single sentence there might be multiple Syntax tree. It is difficult to find out the correct one which force to compromise with the proper analysis.
* It is always difficult to take care of tense used in sentence and may times give an wrong semantic analysis.

These problem ask to give more stress on NLP’s use in requirement analysis. By restricting the scope of the language that is used, the effect of the issues can be minimized.

This process also known to be synchronous language model and is separated from the earlier models, which concludes that human language processing follows a strict sequential manner. Psycholinguistic research suggests that dynamism of language processing and variable order of levels.

Tagging of parts of speech (POS) is a semantic analysis approach and includes assignment of one or more tags to each words. A tagger assign tags to the words. The different types of part of speech tags are as follows:

* The name of almost everything starting from person, place, thing, quality, or action are treated as noun. It acts as subject, object, complement, appositive in a sentence.
* Verbs being the most important part of sentence inform about the action or the. Verbs inform about the state, situation or the doing of the subject.
* Adjectives used describe or modify other words. That identify or quantify another person or thing.
* Adverbs acts as a modifier to adjective. It appear just before that the word it modify.
* Article in English are invariable. The change of gender or the number of nouns do not affect the three articles i.e. ``a, an, the". The learner has to decide noun-by-noun which one of the articles to be used. In fact, there are several choices to make, it is even sometimes needed that no article is necessary in any sentence.
* Conjunction is a part of speech that acts as a connector between two words, sentences, phrases or clauses.
* Interjections are short exclamations like Oh! Um or Ah! Usually these words are used in speaking rather than writing without having any real meaning. Having no grammatical meaning, it sometimes followed by exclamation mark (!).
* Preposition is a word which shows the noun's (or the pronoun's) relationship to another word in the sentence by preceding a noun (or a pronoun) to. A preposition is never followed by a `verb' but it is followed by a 'Noun'.
* Pronoun are small words that take the place of a noun. Pronoun can be used instead of a noun. If pronouns are not there, the repetition of nouns have to be done. Pronoun can perform multiple functions like subject, objects or complements.

In this case study,, the method of semantic and morphology rules are used to analyze the input text used and the different words are identified that help in object oriented analysis.

**Proposed Solution:**

In OO analysis, individual objects are combined to form a software system. Message passing is a mode of communication between them. The values of the attributes and their attributes represents the internal states. Same featured objects are grouped into a class. Hence, Objects are the constituent part of the class.

The main purpose of the design activity is to plan for a design document from informal specification as laid down in SRS document. The modular design document presents the modular structure of a formal specification i.e., external design of the class module. Thus the information about the class module name, function name which each class have contained in it.

Several sub activities are combined to form the design activity. Each producing an informal result. A text file of SRS document is being considered as input. It contains information about the desired software. Then, the file is tokenized into words. The word is searched in different tables.

Eight different tables are taken into consideration for the analysis as there are nine different types of Part of Speech. The ninth part of speech, “Noun" is an open set i.e. the listing of nouns in a table cannot be possible. After analysis of each word if the word is not found in any of the tables, the word is identified to be a `noun'.

However the object diagrams generated by this method is not full proof. The reasons are as follows:

* Inadequacy of Parser: While the parser used in the paper accepts a good deal of English text, but still it cannot handle many sentences. For example, few sentences like the hyphenated words, idiomatic expressions, and quotation marks are not accepted by the parser. But, a complete parser can parse all sentences.
* Insufficient knowledge about the domain: Sentences of the form “Library system accepts a cards'' can be difficult to process for a computer i.e., from whom do the Library System accept the card from whether from faculty or student? A person having domain knowledge can guess the things and carry out the process but a computer program cannot.
* Improper guidelines: Though there are few rules derived for a particular work but it cannot handle all the instances. Consider a query to check if a person is a customer. A person having domain knowledge needs to knows where he has to start the search. So that the chances of getting the result of the query is faster and very recent information about the candidate is retrieved

**Algorithm Used:**

Considering the SRS document as input, the document is tokenized into constituent words using a Java function “StringTokenizer". This function converts the source file into words. Eight different tables such as adjective, adverb, article, preposition, pronoun, conjunction, interjections and verb are used for analysis. The tokenized word is being searched in the respective tables and checked whether the word is present in any table. If the word is present in the table, it is identified at the output as ``WORD----TABLE NAME'', i.e., ``Each--PRONOUN''. If the word is not present in any of the table then it is considered to be of the type `noun' and the word is stored in NON array.

The words that are present in `verb' table are also stored in NON array along with the words of type `noun'. So, the NON array contains both the nouns and verbs type words. After getting all the verbs and nouns stored in NON array, then there is a task to find out the eligible candidate for the class name and the relationship between them.

From the NON array the program generates the output shown in Figure 3 where associations are indicated by a triplet i.e., class -- association -- class. Operations and attributes are indicated by starting their name and the class to which they belong. The parser may produce duplicate suggestions which are done manually.

**Conclusion:**

An effort has been made in this case study to present the software design process based on natural language, and by obtaining the object oriented elements. The process tool consists of ``eclipse indigo" for Java programming and to find out the class name, its attribute and the function inside them.

In this case study, eight different tables are created. SRS document is considered as input to the tokenizer. The output of the tokenizer is searched in different tables to find out which POS, it belongs to. Again from the different POS, both noun and verb tagged words are stored in NON array. Then the array is analyzed to finally get the output.

In future, different expert system logic may be applied to solve different varieties of NLP problems. Grammatical parser is used in this paper to find out required output. Comparing the output of other types of parser to grammatical parser not only helps to find out right solution to the problems but also a mechanism to improve our result. The results give the information about the thoroughness and cost effectiveness of the algorithm used.

**References:**

* Russell J. Abbott, "Program design by informal English descriptions," Commun. ACM, vol. 26, no. 11, pp. 882--894, 1983.
* Sastry Nanduri and Spencer Rugaber, "Requirements validation via automated natural language parsing," in System Sciences, 1995. Proceedings of the Twenty-Eighth Hawaii International Conference on, vol. 3, 1995, pp. 362--368.
* Roger S Pressman, Software engineering: a practitioner's approach.: McGraw-hill New York, 2010, vol. 7.
* Ali Stair Cockburn, "Using Natural Language as Metaphoric Basis for Object Oriented Modeling," 1992