

Automatic Rule Based Approach for Solving Verbal Reasoning Blood Relationship Problems

A Dissertation submitted

By

Neha Narang

(11000998)

to

Department of Computer Science and Engineering

In partial fulfilment of the Requirement for the

Award of the Degree of

Master of Technology in Computer Science and Engineering

Under the guidance of

Mrs. Darvinder Kaur

(Assistant Professor)

(MAY 2015)



LOVELY
PROFESSIONAL
UNIVERSITY

Improving Education Transforming Minds

School of: Lovely Faculty of Sciences & Technology

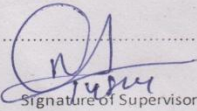
DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Neha Narang Registration No: 11000998
 Batch: 2010-2015 Roll No:
 Division: K2006 Parent Section:
 Name of Supervisor: Designation: AP
 Name: Devinder Kaur Qualification: M.F.
 U.I.D.: 14814 Research Experience: NLP

SPECIALIZATION AREA: Artificial Intelligence (Pick from list of provided specialization areas by DAA)

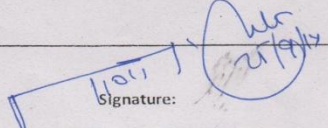
PROPOSED TOPICS

1. Automated knowledge based approach for solving analytical relationship problems.
2. Natural Language Processing
3. On OMC sheets

 2/9/14
Signature of Supervisor

PAC Remarks:

Topic 1 is approved.
Publication is expected from the student.


Signature:

Date:

APPROVAL OF PAC CHAIRPERSON:

- *Supervisor should finally encircle one topic out of three proposed topics and put up for a approval before Project Approval Committee (PAC)
- *Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.
- *One copy to be submitted to Supervisor.

ABSTRACT

In computer science, rule based approach plays a weighty role to store and manipulate knowledge to interpret information in a noteful manner. In this paper, a simple rule-based approach which automatically acquires its rules which is used to determine the solution for verbal reasoning blood relationship problem in step by step manner. They are often used in artificial intelligence applications and research.

Rule based approach requires Natural Language Processing technique for parsing and morphological analysis of sentences. It also requires a large database for matching and mapping of given words of a relationships stored in database. This research will facilitate in machine learning and processing to relationship problems by computers just like humans.

CERTIFICATE

This is to certify that Neha Narang has completed M.Tech dissertation proposal titled **Automatic rule based approach for solving verbal reasoning blood relationship problem** under my guidance and supervision. To the best of my knowledge, the present work is in the result of her original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma. The dissertation proposal is fit for the submission and the partial fulfillment of the conditions for the award of M.Tech Computer Sciences and Engineering.

Date:

Signature of Advisor

Name:

UID:

ACKNOWLEDGEMENT

If words are considered to be the signs of gratitude then let these words convey the same. My sincere gratitude to my mentor “Mrs. Darvinder Kaur”, giving necessary directions on doing this research to the best of my abilities as she is the mentor of the thesis for correcting various problems of mine with attention and care.

Neha Narang

DECLARATION

I hereby declare that the dissertation proposal entitled, **Automatic rule based approach for solving verbal reasoning blood relationship problems** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: April 29, 2015

Neha Narang

11000998

TABLE OF CONTENTS

	Page no
Abstract	i
Certificate	ii
Acknowledgement	iii
Declaration	iv
Table of Contents	v
List of Figures	vi
List of Tables	vii
CHAPTER 1 INTRODUCTION	1-18
1.1 Introduction on NLP	1-2
1.2 General architecture for solving problem on NLP	2-16
1.3 Tool used for solving the problem in NLP	16-18
CHAPTER 2 REVIEW OF LITERATURE	19-22
CHAPTER 3 PRESENT WORK	23-28
SCOPE OF THE STUDY	23
OBJECTIVES	24
METHODOLOGY	25-28
CHAPTER 4 RESULTS AND DISCUSSIONS	29-40
CHAPTER 5 CONCLUSION AND FUTURE SCOPE	41
CHAPTER 6 REFERENCES	42-43

LIST OF FIGURES

FIGURE No.	Figure Content	Page No
Figure 1	Components for rule based system	2
Figure 2	General architecture for solving problem in NLP	3
Figure 3	Syntactic Analysis	4
Figure 4	Relations from one generation to another	5
Figure 5	Rules those are stored in database	6-8
Figure 6	Block diagram for solving blood relationship problem	27
Figure 7	Flowchart for solving the problem	28
Figure 8	Interface of Module 1	29
Figure 9	Interface of Module 2	30
Figure 10	Input for Module: 1, Type: 1	31
Figure 11	Output for Module: 1, Type: 1	32
Figure 12	Input for Module: 1, Type: 2	33
Figure 13	Output for Module: 1, Type: 2	35
Figure 14	Input for Module: 1, Type: 3	36
Figure 15	Output for Module: 1, Type: 3	37
Figure 16	Input for Module: 2	38
Figure 17	Output for Module: 2	39

LIST OF TABLES

Table No.	Table Content	Page No.
Table 1	List of blood relationships.	9
Table 2	Data sets for finding the accuracy and efficiency of the system.	40

CHAPTER -1

INTRODUCTION

1.1 Introduction on Natural Language Processing

Language is all about communicating about the world. By studying language, a person can understand more about the world. But there are many difficulties to do so as how knowledge can be exploited about the world, in combination with linguistic facts,[3] to build computational natural language system? What is the platform or the framework through which humans can interact with computer?

Natural Language Processing is the branch of computer science and scientific study of a language that focused on developing a system which allows computers to communicate with people using everyday language.[14] It is one of the technique through which human can interact with the system. The major concern in the field of NLP is to perform the task in useful and an efficient manner by the machine in comparison to human being. It is the process in which human language is provided as an input and the conversion of this input is used to form a representation.[3]

NLP can be used to solve the problems of verbal reasoning blood relation problem. Rule based approach is used to develop a system that will provide answers to a problem in lieu of human beings.

It consists of 4 main components[2,13]:-

- **Rules:** A list of rules that shows definite type of knowledge base. Knowledge is stored as rules in the rule-base also known as the knowledge base.
- **An inference engine:** An inference engine is a tool that is used to draw the conclusion by applying logical rules to the knowledge base to generate new knowledge.[17] A system production program is executed by the interpreter so as to perform match-resolve-act cycle.
- **Match:** In the first stage, the productions in the left-hand side are matched against the working memory's contents. As a result a conflict set is obtained, which consists of

instantiations of all satisfied productions. An instantiation of a production is an ordered list of working memory elements that satisfies the left-hand side of the production.

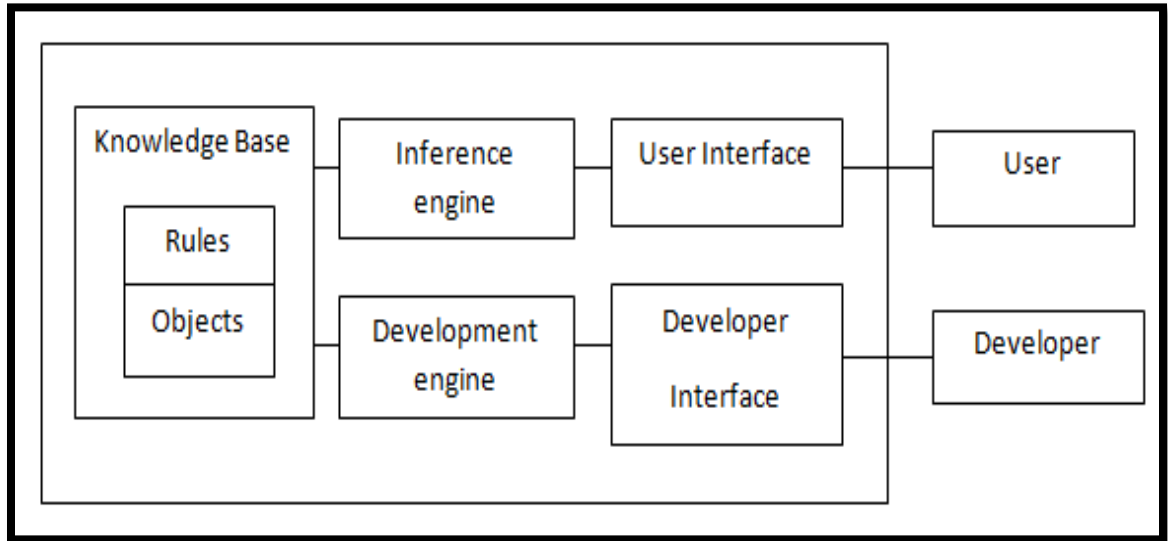


Figure 1: Components for rule based system[10]

- **Conflict-Resolution:** In the second phase, selection of one of the production instantiations in the conflict set is performed for execution. [2] The interpreter will halt when no productions are satisfied.
- **Act:** The executions of the production that are chosen in the conflict-resolution phase are accomplished. By using these actions the contents of operation memory may vary. At the end of this phase, execution reverts to the first phase.
- **A user interface:** A user interface provides a framework through which a human can interact with the system.

1.2 General architecture for solving a problem of NLP

1.2.1 Filters

In this module the transmutation from source format (e.g. HTML or XML) to plain text is executed i.e. it removes multiple blanks or void at the origin of sentences.

1.2.2 Tokenization

Tokenization or Morphological Analysis is the procedure of graduating the sentences into words, phrases, symbols or some other meaningful elements called tokens which are generated by list of actual words by which sentence are being made. These tokens are necessity as an input in the next phase. The main intend of this module is to recognize the different tokens which are personate in a text and disconnect them in such a way that every word and punctuation mark will be examine as a distinct token. It is used for Natural Language Processing of human languages such as English to make it understandable to computers.

For example: text:- To be a good learner, first be a good listener.

The output of the text will be:-

['To', 'be', 'a', 'good', 'learner', ',', 'first', 'be', 'a', 'good', 'listener', '.']

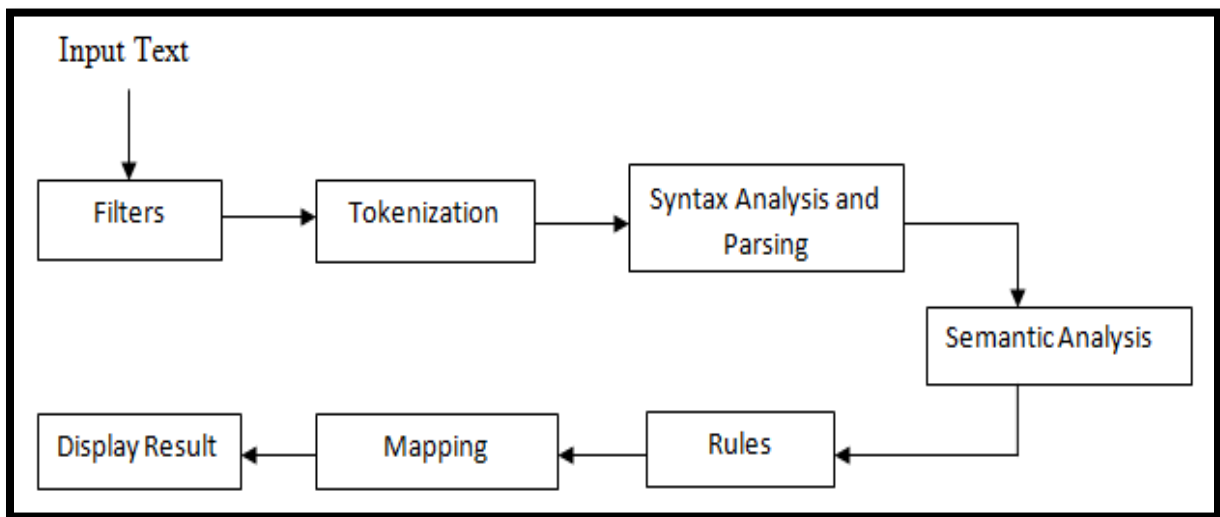


Figure 2: General architecture for solving problem in NLP[10]

1.2.3 Syntax Analysis (Parsing)

The main scope of this process is parsing. It plays an essential role in understanding many language systems. In this step, a simple sentence is granted as an input which is converted into a hierarchical form that address to the units of meaning in the sentence. It uses the first components of the token to produce tree probably intermediate structure that depicts the grammatical structure of the token stream.

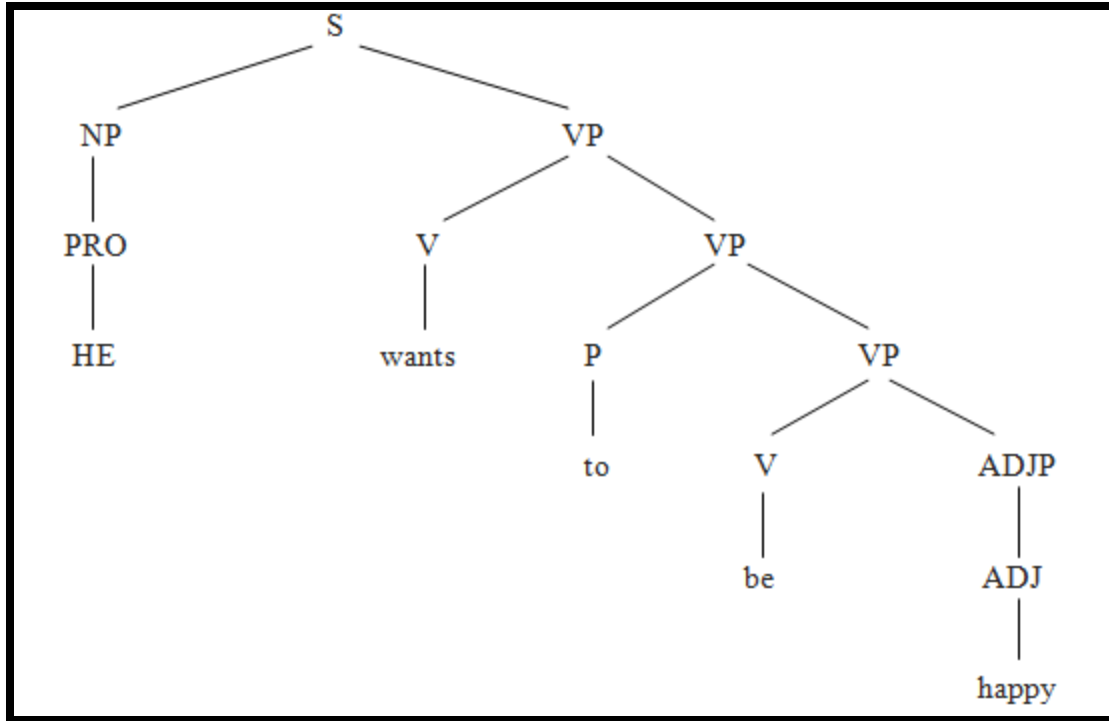


Figure 3: The result of syntactic analysis of “He wants to be happy” [3]

There are distinct parsing formalisms and algorithms in which formalism has two leading components:

- Grammar: A declaratory representation that portray the syntactic structure of sentences in the language.
- Parser Based on the morpho-syntax rules, it captivate the input and gives output as a syntax tree.

1.2.4 Semantic Analysis

Semantic analysis is used to check whether inserted sentence is accurate or not. Although the main intend of semantic analysis is the formation of the target language representation of the sentence’s meaning which indicate assigning meanings to the structures created by syntactic analysis. Semantic can play an import role in opt among contend in syntactic analyses. For example: I need to complete the project – here project is a plan/ a proposal or to cause a shadow.[3] So here we have to make up one's mind the formalisms which will be utility in the meaning representation.

1.2.5 Rules

Knowledge can be represented either in the form of rules or facts. Conclusions can be drawn on the basis of experience, beliefs based on existing or new information, attitudes, self-determination and institutions.

To generate the rules for verbal reasoning you should have a sound knowledge of the blood relation in order to solve the questions. Blood relation logical problems mainly deal with the hierarchical structure of a family i.e., grand-parents, parents and children etc.[16] Different relationships between the family members of different generations will be given. To solve the questions related to blood relations, the entire family tree has to be drawn by putting the various relationships.

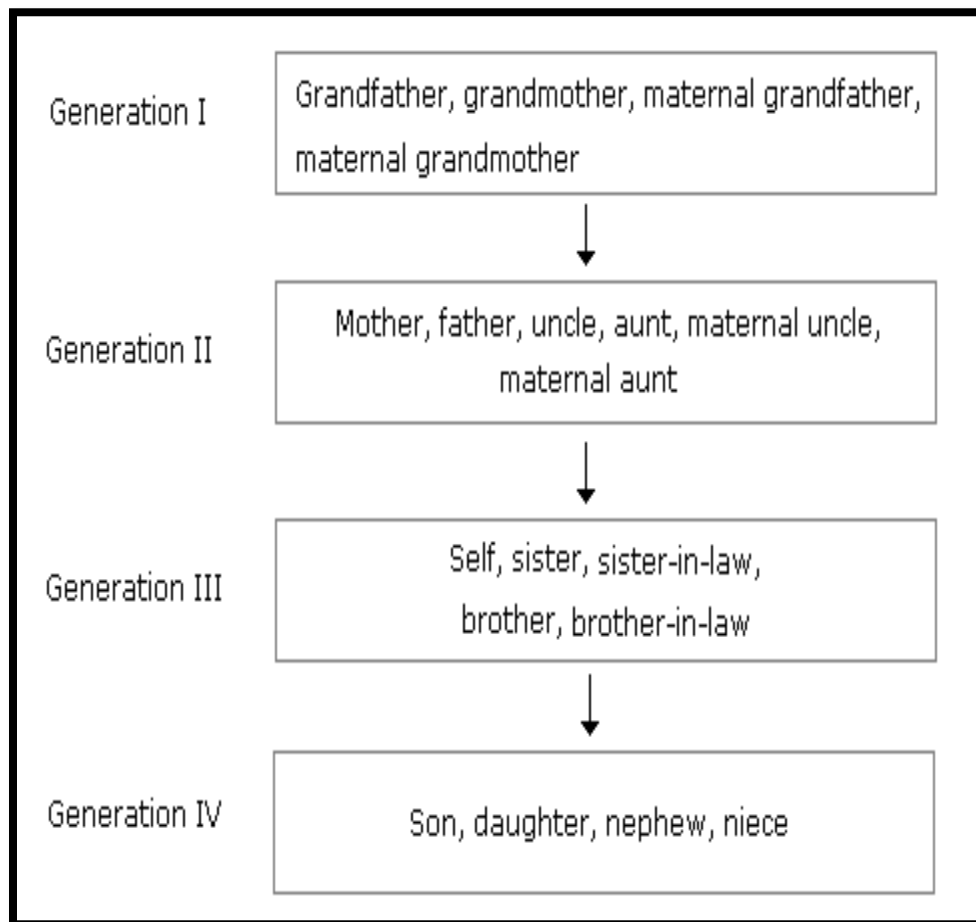


Figure 4: Relations from one generation to another.[16]

All the rules that are stored in the database and used by the system are:-

```

parent(X, Y)      :- (paren(X, Z), sibling(Z, Y), X \= Y, X \= Z, Z \= Y);
                  (paren(Z, Y), married(X, Z));
                  (paren(X, Y), X \= Y).

father(X, Y)      :- male(X),    parent(X, Y).

mother(X, Y)      :- female(X),  parent(X, Y).

sibling(X, Y)     :- (siblin(X, Z), siblin(Z, Y), X \= Y);
                  (siblin(X, Z), siblin(Z, M), siblin(M, Y), X \= Y, X \= Z, X \= M,
                  Z \= M, M \= Y); (siblin(X, Y), X \= Y).

brother(X, Y)     :- male(X),    sibling(X, Y).

sister(X, Y)      :- female(X),  sibling(X, Y).

uncleoraunt(X, W) :- (sibling(X, Y), parent(Y, W), X \= W, X \= Y, W \= Y);
                  (uncleoraun(X, W), X \= W).

uncleoraunt(X, Z) :- (married(X, Y), sibling(Y, W),
                  parent(W, Z), X \= Z, X \= Y, X \= W, Y \= W, Y \= Z, Z \= W).

uncle(X, W)       :- male(X),    uncleoraunt(X, W).

aunt(X, W)        :- female(X),  uncleoraunt(X, W).

nieceornephew(X, Y) :- (parent(Z, X), sibling(Z, Y), X \= Y);
                  (grandparent(Z, X), parent(Z, Y), Z \= X, Z \= Y, not(parent(Y, X)));
                  (uncleoraunt(Y, X));

```

nephew(X, Y) :- male(X), nieceornephew(X, Y).

niece(X, Y) :- female(X), nieceornephew(X, Y).

grandparent(X, Z) :- (parent(X, Y), parent(Y, Z), X\=Z, X\=X, Y\=Z);
 (parent(X, Y), sibling(Y, K), (parent(K, Z), X\=Z, X\=Y,
 X\=K, Y\=K, Y\=Z, K\=Z)); (grandparen(X, Y), X\=Y).

grandfather(X, Z) :- male(X), grandparent(X, Z).

grandmother(X, Z) :- female(X), grandparent(X, Z).

child(Y, X) :- (sibling(Y, Z), parent(X, Z));
 (chil(Y, X), X\=Y).

son(Y, X) :- male(Y), child(Y, X).

daughter(Y, X) :- female(Y), child(Y, X).

grandchild(Y, X) :- (grandparent(X, Y)); (parent(X, Z), parent(Z, W), sibling(W, Y));
 grandchil(Y, X).

grandson(Y, X) :- male(Y), grandchild(Y, X).

granddaughter(Y, X) :- female(Y), grandchild(Y, X).

married(X, Y) :- (paren(X, Z), paren(Y, W), sibling(Z, W), X\=Y, Y\=Z, X\=Z).

husband(X, Y) :- (male(X), married(X, Y)); (not(female(X)), married(X, Y)).

wife(X, Y) :- (female(X), married(X, Y)); (not(male(X)), married(X, Y)).

sisterinlaw(X, Y)	:- (married(X, Z), sibling(Z, Y), X \= Y, X \= Z, Y \= Z); (sisterinla(X, Y)).
parentinlaw(X, Y)	:- married(Y, Z), parent(X, Z).
fatherinlaw(X, Y)	:- male(X), parentinlaw(X, Y).
motherinlaw(X, Y)	:- female(X), parentinlaw(X, Y).
siblinginlaw(X, Y)	:- married(Y, Z), sibling(X, Z).
brotherinlaw(X, Y)	:- male(X), siblinginlaw(X, Y).
sisterinlaw(X, Y)	:- female(X), siblinginlaw(X, Y).
childinlaw(X, Y)	:- married(X, Z), child(Z, Y).
soninlaw(X, Y)	:- male(X), childinlaw(X, Y).
daughterinlaw(X, Y)	:- female(X), childinlaw(X, Y).

Figure 5: Rules those are stored in database

1.2.6 Mapping

Mapping is the drawing conclusions from the given premises to solve problems and make decisions. It basically manipulates the given knowledge and generates new knowledge from the defined rules and facts that are stored in the database. It derives the new knowledge with the help of logics or by using inference. So, the relationships that are stored in database and will be used in mapping are:-

Table 1: List of blood relationships [16]

S.No	Person	Relationship
1	Mother's or Father's son	Brother
2	Mother's or Father's daughter	Sister
3	Mother's or Father's brother	Uncle
4	Mother's or Father's sister	Aunt
5	Mother's or Father's father	Grandfather
6	Mother's or Father's mother	Grandmother
7	Son's wife	Daughter-in-Law
8	Daughter's husband	Son-in-Law
9	Husband's or wife's sister	Sister-in-Law
10	Husband's or wife's brother	Brother-in-Law
11	Brother's son	Nephew
12	Brother's daughter	Niece
13	Uncle's or aunt's son or daughter	Cousin
14	Sister's husband	Bother-in-Law
15	Brother's wife	Sister-in-Law
16	Grandson's or Granddaughter's daughter	Great grand daughter

TEXT SEGMENTATION

Topic analysis comprises of two main procedures: topic identification and text segmentation. On one hand where topical identification deals with simply classifying the text, whereas on the other hand text segmentation is the methodology of partitioning composed content into significant units, for example, words, sentences, or points. The term applies both to intellectual methods utilized by people when reading content, and to manufacture methodologies implemented in PCs, which are the subject of natural language processing.[1]

Text segmentation methodology is based on a technique which step by step builds a model to extract features that are associated with the presence of units in labeled training text. This model uses two types of features:

First, the topicality feature that uses adaptive language models in a new way to detect broad changes of topic, and second, the cue-word feature that identify occurrences of specific words, which can be domain-specific words. Evaluation of approach on quantitative and qualitative grounds demonstrates its effectiveness in two very different domains. Quantitative results are shown by error metric, which combine precision. This metric is used to make a quantitative assessment of the relative contributions of the different feature types, as well as a comparison with decision trees.

Segmentation of text into topics is useful in many natural processing tasks for instance, it improves information extraction or speech identification. It is also found helpful in topic recognition, tracking method and text review issues.

However, it is quite an uncertain task because it is found that people assessing the text segmentation methods often differ in topic boundaries. Hence, evaluating it is a bit tedious.[23]

SENTENCE PARSING

Parsing is the method by which string of symbols is analyzed. This string can be in natural language or in computer language, with a condition that it should conform the rules of a formal grammar.

Sentence parsing has distinct meanings in distinct branches of computer science. In traditional parsing, sentence parsing is considered as a method which understands the exact meaning of a sentence. It usually pays stress on grammatical divisions like subject and predicate.

In computational linguistics, the sentence parsing refers to the analysis of a sentence or string by a computer into different component, resulting in a parse tree which shows relation, semantic or other information belonging to them.[25]

In psycholinguistics, sentence parsing analyze a sentence the way humans analyze them dividing it into grammatical components, different parts of speech, relations and so on.

However, in computer science, it is used in the assessment of computer languages, in order to assist in the writing of compilers by analyzing the input data into components.

A parser is a component of software which takes data as input and develops a data structure – which can be a parse tree, a syntax tree or any other tree structure. The parsing can be carried out in different steps or in a single step. If it is carried out in steps, parser is lead by lexical analyzer, which helps in creating tokens from input characters; on the other hand, these can also be combined in scanner less parsing. for programming the parsers, it can be done automatically or semi-automatically by parser generator or can be programmed by hand. Parsing as an output gives the formatted output. These can be applied to various domains, but usually appear together, like scanf/printf pair, or input/output pair [25].

Thus, parsing is an important aspect in dividing a sentence into words and matching those words with the pattern, already defined in the database.

PATTERN MATCHING

A regular expression determines a search pattern for strings. The abbreviation for regular expression is regex. The search pattern can be anything from a simple character, a fixed string or complex expression confines special characters describing the pattern.[15] The pattern decide by the regex may match one or several times or not at all for a given string. Regex pattern matcher is used to match the pattern of a sentence and apply the rules accordingly. Some of the patterns defined in this system are:-

Relations Pattern

- `((IN |CC)(NNP|NN|DT) (VBZ)(.| DT)(NN) (IN) (NNP|NN|DT))`
If/ Mohan is /a/an/the brother of Sohan
- `((IN)(NN|NNP|DT) (VBZ)(.| DT)(NN) (IN) (NN|NNP|DT) (CC)(| IN)(NN|NNP|DT) (VBZ)(.| DT)(NN) (IN) (NN|NNP|DT))`
If Mohan is the brother of Sohan and Radha is the sister of Mohan.
- `((IN |CC)(NNP|NN|DT) (VBZ)(.| DT)(NN) (NNP|NN|DT))`

If Sunita has a sister Anita.

- (|IN |CC)(NNP|NN|DT) (VBZ)(.| DT)(NN|NNP|DT) (POS) (NNP|NN|DT)

If Avi is Sonal's brother.

- (|IN)(NNP|NN|DT) (CC) (NNP|NN|DT) (VBZ) (NNP|NN|DT) (VBZ)(.| DT)(NN) (IN) (NNP|NN|DT)

If A + B means A is the brother of B;

- (|IN)(NNP|NN|DT) (CC|:|NN|CD|SYM|@|#\$|#|\$) (NNP|NN|DT) (VBZ) (NNP|NN|DT) (VBZ)(.| DT)(NN) (IN) (NNP|NN|DT)

Question Pattern

- (|RB)(WRB) (VBZ) (NNP|NN|DT) (JJ|VBN) (TO) (NNP|NN|DT)

How is Ram related to Sohan

- (|RB)(WP) (VBZ) (NNP|NN|DT) (IN) (NNP|NN|DT)

What is Ram of Mohan

- (|RB)(WRB) (NNP|NN|DT) (VBZ) (VBN|JJ) (TO) (NNP|NN|DT)

How Ram is related to A.

- (WP) (VBZ)(.| DT)(NN) (IN) (NNP|NN|DT)

Who is the uncle of Jagat

- (|DT)(NNP|NN|DT) (IN) (NNP|NN|DT) (VBZ)

The uncle of A is.

- (WDT) (IN)(.| DT)(VBG)(.| NN)(VBZ)(.| DT)(NN)(.| IN)(NNP|NN|DT)

Which of the/ following option/ is the uncle of A.

- (WDT) (IN)(.| DT)(NN) (VBZ) (NN) (IN) (NNP|NN|DT)

Which of the option mean C is the son of M.

- (WDT) (IN)(.| DT)(VBG) (VBZ)(.| IN)(NNP|NN|DT|PRP) (VBZ)(.| DT)(NN) (IN) (NNP|NN|DT)

Which of the following means that M is the wife of Q?

- (WDT) (IN)(.| DT)(JJ) (NNS|NN) (VBZ)(.| IN)(PRP) (VBZ)(.| DT)(NN) (IN) (NNP|NN|DT)

JAVA

Java is a computer programming language which is based on classes and objects, particularly used for client-server web applications. Java is platform independent which means that the developer has to write the code once and can run that program anywhere i.e. "write once, run anywhere"(WORA). The developer does not need to recompile the code again and again so; the program can run on any platform that supports Java. It was developed by James Gosling at Sun Microsystems which has been acquired by Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. As java applications are compiled to bytecode that can run on JVM instead of computer's architecture. It has many advantages that it is robust and secure. It executes the program with high performance moreover it is interpreted, threaded and dynamic.

JAVA PLATFORM

A Java virtual machine (JVM) is a conceptual processing machine which is known for its detail, execution, and cases. The purpose depicts the need of a JVM execution. It guarantees all usage is interoperable on the off chance that it has single particular. A JVM usage is a PC program that meets the prerequisites of the JVM detail in an agreeable and ideally every formant way. An occasion of the JVM is a methodology where PC system assembled into Java bytecode gets executed.

The Java virtual machine is a conceptual (virtual) PC characterized by a particular. This particular overlooks execution subtle elements that are not key to guarantee interoperability. For instance, the memory design of run-time information regions, the waste gathering calculation utilized, and any inward advancement of the Java virtual machine guidelines (their interpretation into machine code). The fundamental explanation behind this oversight is to not superfluously compel practitioners. Any Java application can be run just inside some solid execution of the conceptual particular of the Java virtual machine.

REGEX

The Java Regex or Regular Expression is an API which defines pattern for finding or changing the strings. One of the most important applications of REGEX is to define restriction on strings such as password and email validation.

In REGEX user is made available with `java.util.regex` package which includes 1 interface and 3 classes.

`java.util.regex` package

It provides following classes and interface for regular expressions. The `Matcher` and `Pattern` classes are widely used in java regular expression.

1. `MatchResult` interface
2. `Matcher` class
3. `Pattern` class
4. `PatternSyntaxException` class

A search pattern for strings is defined by general expression. The shorten form for regular expression is `regex`. The search pattern that is described by the `regex` can be in any form either a simple character or a fixed string or any complex expressions which contains some special character that describes the patterns, which might match one or a few times or not in

the least for a given string. General expression can be utilized to alter, control and search the text.[24]

The methodology of analyzing or changing a content with a regex is called: The regular expression is applied to the text (string) . The pattern defined by the regex is applied on the text from left to right. When a source character has been utilized as a part of a match, it cannot be reused. For example, the regex aba will match ababababa only two times (aba_aba__).

REGEX is basically used to define restriction on strings for a password and email validation. Literal string is an example for regular expression. For example, the Hello World regex will equivalent the "Hello World" string. (dot) is another case for standard declaration. A dot matches any single character; it would go with, for example, "a" or "z" or "1".

Representation of regular expression in the package java.util.regex

The java.util.regex bundle principally comprises of three classes: Pattern, Matcher, and PatternSyntaxException.[17]

A pattern object is an accumulation of regular expressions. In a pattern class there is no public constructor available, hence, whenever a pattern need to be created, first of all a call is made to one of public static method, which then returns a pattern object. These static methods accept a regular expression as the first argument.

A Matcher object is a tool that understands the pattern and carries out match operations against a string which is given as an input. In Matcher also, there is no definition of public constructors. So, to obtain a matcher object, a matcher method is invoked on a pattern object.

PROLOG

Prolog is a declarative programming language which means that the program logic is uttered in terms of relations, signify as facts and rules. So the programmer do not write the code about what a computer should do line by line, as in procedural language alike in C and Java. The general idea is to describe a situation and on the basis of that situation the compiler will

convey a solution in terms of true and false and if it contains the variables then it inform about the values of the variables.

"It is often said that it is easier to learn prolog for someone who does not have any experience in procedural programming than for someone who does."[18]

Prolog is most beneficial in the areas related to Artificial Intelligence research, such as problem solving, (path) planning or natural language interpretation.

Prolog is considered to be a difficult language, because every individual has its own way of thinking to adopt and amount of recursion in Prolog programs. So, when used correctly, Prolog can be a very powerful language.

JPL

JPL is a library that provides a bidirectional¹ interface between Java and Prolog by using the Java Native Interface (JNI) to connect to a Prolog engine through and the SWI-Prolog foreign interface which basically used to embed Prolog in Java and vice-versa. JPL is not a pure Java implementation of Prolog; it makes extensive use of native implementations of Prolog on supported platforms.[19]

JPL is considered to be the important library that is designed in two layers, a low-level¹ interface to the Prolog FLI and a high-level Java interface for the Java programmer who is not concerned with the details of the Prolog FLI. The low-level interface is provided for C programmers who may wish to port their C implementations which use the FLI to Java with minimal complaints.

1.3 Tool used for solving the problem in NLP

Stanford CoreNLP: A Suite of Core NLP Tools

Stanford CoreNLP is an integrated framework that provides a set of natural language analysis tools which can take plain text as a input and give the base forms of words, their parts of speech, whether they are name of the person, organization or any institution, normalize dates,

times, and numeric quantities, and mark up the structure of sentences in terms of phrases and word dependencies, that indicate which noun phrases refer to the same entities, indicate sentiment, etc. It includes many natural language tools that contain the tokenizer, the part-of-speech (POS) tagger, the named entity recognizer (NER), the parser, the coreference resolution system, the sentiment analysis, and the bootstrapped pattern learning tools. The fundamental distribution provides model files for the analysis of English, but the engine is compatible with models for other languages as well like models for Chinese and Spanish, and Stanford NLP models for German and Arabic are usable inside CoreNLP. Its goal is to make it very easy to apply a bunch of linguistic analysis tools to a piece of text. Starting from plain text, just write two lines of code and run all the tools on it. It is designed to be highly flexible and extensible. With a single option you can change which tools should be enabled and which should be disabled. Its analyses provide the foundational building blocks for higher-level and domain-specific text understanding applications.[20]

Stanford Tokenizer

A tokenizer divides text into a sequence of tokens. PTBTokenizer is an efficient, fast, deterministic tokenizer class suitable for tokenization of English words. It splits the text into sentences. A sentence is split when it found the sentence-ending character as (. , ; , or ?). It has the ability to tokenize the text at a rate of about 200,000 tokens per second. It was initially designed to largely mimic Penn Treebank 3 (PTB) tokenization, hence its name, though over time the tokenizer has added quite a few options and a fair amount of Unicode compatibility, so in general it will work well over text encoded in the Unicode Basic Multilingual Plane that does not require word segmentation (such as writing systems that do not put spaces between words) or more exotic language-particular rules (such as writing systems that use : or ? as a character inside words, etc.)[21].

Stanford Log-linear Part-Of-Speech Tagger

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that has the ability to read text in some language as an input and assigns parts of speech to each token such as noun, verb, adjective, etc., as an output. Although generally computational applications use more fine-grained POS tags like 'noun-plural'. pos.model: POS model to use. There is no need to

explicitly set this option, unless you want to use a different POS model (for advanced developers only). By default, this is set to the english left3words POS model included in the stanford-corenlp-models JAR file.[22]

CHAPTER -2

REVIEW OF LITERATURE

Manning, Christopher D., Surdeanu, Mihael, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014, presented the design and active development of the Stanford CoreNLP tool-kit. Since there are several good natural language analysis toolkits, but Stanford CoreNLP is one of the most efficient tool-kit which has a Java annotation pipeline framework which uses the general architecture steps of natural language processing(NLP) i.e. tokenization, Stanford POS Tagger, Stanford Parser, Stanford Named Entity Recognizer, Stanford Word Segmenter and classifiers. In this paper the author described the design of the system, its strength, different patterns, the set of provided annotations and how their properties control them. This toolkit is very extensively used, both in the research NLP community and also among commercial and government users of open source NLP technology.[6]

Ning Fang and Yongqing Guo (2013), proposed a web-based interactive intelligent tutoring system that was developed and assessed in an engineering dynamics course. To improve student learning in an engineering dynamics course, a web-supported interactive ITS consisting of two learning modules was developed. An intelligent tutoring system (ITS) is an interactive learning tool that allows students to solve technical problems with the direction of a virtual tutor. During problem solving, students can ask the virtual tutor questions or request hints on what to do next, just as may occur in an actual classroom environment where students can ask a human tutor questions or request hints. These ITS modules help students to study the principle of Work and Energy to solve particle and rigid-body dynamics problems. Student literature gains were compared using a quasi-experimental research design that complex pretests and posttests in both a control semester (n = 62) and a management semester (n = 44). Therefore, the ITS modules increased student learning profitable by 37-43%.[11]

Anatolii Vovk , Denys GirnykIn (2013), distribute with the Web supported notation, which is used to accomplish a natural language representation of mathematical texts and preserves their semantics. A visualization of notation for browsers is developed and its export to standard data format TeX, Content MathML and PDF. The intimate WEB-notation provides an interactive connection over the Internet, as well as compatibility and interoperability of adapted texts in other applications. It does not fully shield the semantics of mathematical formulas, which is specified as a formal interpretation of the mathematical formation on a computer language and the structure of the online the transactions are also defined. An interactive communication over the Internet, with serve of the notation MTV, is carried out between internet browsers with the ActiveX-plugins, which yield the notation and add editing functions. The future work enclosed nature of plugins for open source software under the Linux OS.[7]

Cinková, Silvie, Martin Holub, and Vincent Kríž (2012), debate about the experiments with semantic annotations which are supported on the corpus pattern analysis and the lexical resource that reveal the necessity of an evaluation measure that would recognize the efficient relation between the semantic granularity of semantic categories. In this automatic approach, personal names are noticeable manually. A corpus based learning method is presented by Kwak et al. that can index distinct types of nouns by using rules which are automatically extracted from a large tagged corpus. It is to be emphasized that reliable gain does neither directly assess the quality of the entry nor the quality of annotations. The output has to be analyzed by a man. This tool is capable enough that it can be manner with any kind of annotation where the tag set is to be enhance during the annotation. [5]

Rutu, Mulkar-Mehta, Jerry Hobbs and Eduard Hovy (2011), debates on the phenomenon of granularity in native language. The level of elaborated description of an event is dissolve by using granularity. With the help of an annotation study validation of standard is being done and then develops a system for automatic granularity extraction and ultimately compares its performance with the condition of an art technique for answering causality-style questions to

empirically evaluate the significance of granularity structures for automated question answering.[4]

Matsuda, Noboru, (2008) debates on SimStudent which is a machine-learning agent which is initially developed to aid newbie authors to create cognitive tutors without weighty programming. A Cognitive Tutor Authoring Tools (CTAT) is used to renew into an existent suite of software tools. SimStudent helps authors to compose an expert model for a cognitive tutor to solve the problems. In Authoring by Tutoring, the author interactively tutors SimStudent by posing problem, gives feedback on the basis of response conceive by the SimStudent and provide hints when SimStudent can't accomplish steps correctly. So Authoring by Tutoring is beneficial in creating a high quality expert model which is bulky due to the feedback on the accuracy of learned predictions applied while explain the problems. It was analyzed that programming by tutoring is a practical and effectual technology automated cognitive modeling.[9]

Punyakanok, Vasin, Dan Roth, and Wen-tau Yih (2008), observes that the semantic parsing of sentences is to be a weighty task on natural language understanding and has an instant application in tasks resembling, information extraction and question answering. The main scheme of Semantic Role Labeling (SRL) is to recognize all constituents that fill a semantic role, and to figure out their roles in a superficial semantic parsing task, which is necessity for each predict ate in a sentence. The goal of the semantic role labeling task is to discover the predicate–argument structure of each predicate in a stated input sentence. In this process, the full parse tree is disposed as an input to SRL system which is considered to be the most ordinary system architecture. [12]

Barcala, Francisco-Mario(2002), concentrate on a professional tokenizer that is necessity to analyze an ample amount of texts by applying a set of natural language processing techniques which induce to a number of linguistic phenomena and also for pre-tagging tasks alike proper noun recognition. It gives the itemized description of pre processor module for the precise segmentation of texts which demonstrate number of complicated linguistic phenomena, which includes the recognition of proper nouns. Several other originators have also examined

the significance of name recognition of retrieval system. Pfeifer et al. study the search methods for the efficiency of several methods of single surname. In comparison with automatic approach, Kwak et al. present a corpus supported learning method that can index several types of (common) compound nouns by using rules that are automatically extracted from a colossal tagged corpus. [8]

M. El - Arabaty (1998), describes the approach for ESSAP architecture which is an expert system for solving the problem of aerospace by using artificial notice techniques and the mathematical models. The ESSAP can be developed to aid the expert in the analysis of experiments and their result definition. So while designing the ESSAP architecture the researcher determines that the system has the capability to take a advice action automatically if the user agrees in outcome. ESSAP has the aptitude that it can accept the advice an act on it so as to communicate their result with one another and suggestion changes dynamically when the system trace the user interface.[10]

Jonathan J. Webster and Chunyu Kit (1992), addresses the weight and the complicity of tokenization at the beginning of Natural Language Processing. From the sight points of lexicography and pragmatic implementation, respectively notions of word and token are discuss and defined. Automatic division of Chinese words is delivering as an apprehensible of tokenization. Practical approaches are developed to recognize the compound tokens in English, like idioms, phrasal verbs and fixed expressions are developed. In NLP, while taking the basic units for granted, it is canonical to concentrate on simple analysis. In this process, the notion of token must be determine first and after that computational processing can progress. The researcher has decided constitutes of token from two perspectives i.e. one from the lexicographer's experience with distinguishing words, the other from the experience of researchers in the range of Chinese NLP. In the case of Chinese NLP, the researcher concerned for some worthy lessons which can be referred for the recognition of idioms, phrasal verbs and other unalterable-expressions in English.[13]

CHAPTER- 3

PRESENT WORK

In this thesis, an automatic rule based approach is used to solve the problem of blood relationship problem by using Natural Language Processing. NLP provides a framework through which a user can interact with the system. In this system the user can enter the questions in tow modules. In first module the user is able to find the relationship and in the second module the user is able to find the equation of the relationship.

SCOPE OF THE STUDY

The main aim of the research is to reconnoitre the answers to questions through the application of scientific procedures. The capital aim of the research is to find out the fact which is secret and which is not been disclose yet.

1. It can be helpful for the students that are preparing for the competitive exams and can understand the problem in a proper and an efficient manner [13].
2. It is used to make a decision support system that will work in a similar manner like humans and can respond just like humans by understanding the different forms of sentences given to it as an input[7].
3. It can also help in developing an intelligent tutoring system (ITS) which is an interactive learning tool that allows students to solve problems with the guidance of a virtual tutor. [9]
4. It can sometimes act as a 10 year old child that can finds relationship among family members.[1]

OBJECTIVES OF THE STUDY

1. Problem is formulated by doing extensive literature survey.
2. To generate a set of rules for creating the relationships by using different set of facts that are defined in database using Prolog.
3. To study and analyze various steps for generating sentence tokenization, parts of speech tagging, word stemming through Stanford tool and mapping by using Prolog's rule.
4. To propose and design an algorithm which takes input as a paragraph or sentences and after processing, the solution of the problem is generated as an output.
5. To verify and test the efficiency and accuracy of the result.

RESEARCH METHODOLOGY

The project is fully based on Natural Language Processing (NLP) so the main goal is to simplify the task by choosing a language and tool that will support work. So Stanford Core NLP is the appropriate tool for implementing the idea and Prolog is a programming language in which program logic is expressed in terms of relations, represented as facts and rules.

JPL is a library that provides a bidirectional interface between Java and Prolog by using the Java Native Interface (JNI) to connect to a Prolog engine through and the SWI-Prolog foreign interface which basically used to embed Prolog in Java and to embed Java in Prolog.

ALGORITHM

- STEP 1: Read question from text box.
- STEP 2: Segment question in sentences.
- STEP 3: Set count=0.
- STEP 4: Set Rel=null.
- STEP 5: while(count<no_of_sentences-1) repeat step 6 to 11.
- STEP 6: Tag sentence=count using Stanford POS tagger.
- STEP 7: Separate tokens and tag into two arrays.
- STEP 8: Match pattern of Tag from already defined rules available in knowledge base.
- STEP 9: Use tokens as per matched pattern to form temporary rules.
- STEP 10: Set Rel=Rel+rules.
- STEP 11: count=count+1

- STEP 12: Read knowledge base file from memory for relationship data in a string variable k.
- STEP 13: Create a temporary file.
- STEP 14: Add Rel and k to temporary file.
- STEP 15: Tag sentence=count using Stanford POS tagger for getting question.
- STEP 16: Match pattern of Tag from already defined rules available in knowledge base for testing all relation.
- STEP 17: Set m=0
- STEP 18: Set match=0
- STEP 19: while(m<no_of_relation) repeat step 18 to 20
- STEP 20: if(Relation(Person1,Person2)==true)
- STEP 21: Set match=1 and Display Relation.
- STEP 22: m=m+1
- STEP 23: If (match==0)
- STEP 24: Display “Can’t Find Relation”.
- STEP 25: END

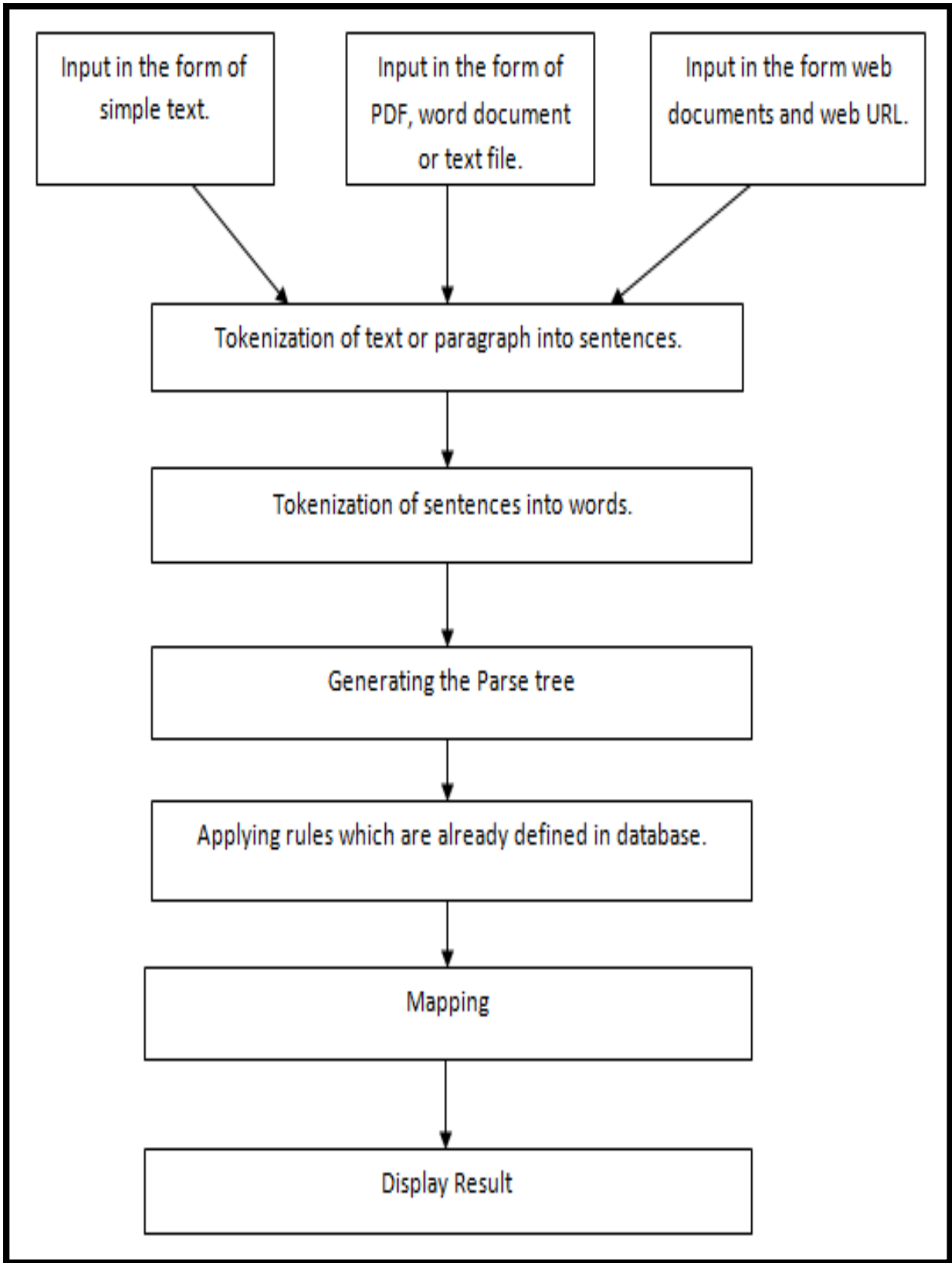


Figure 6: Block diagram for solving blood relationship problem

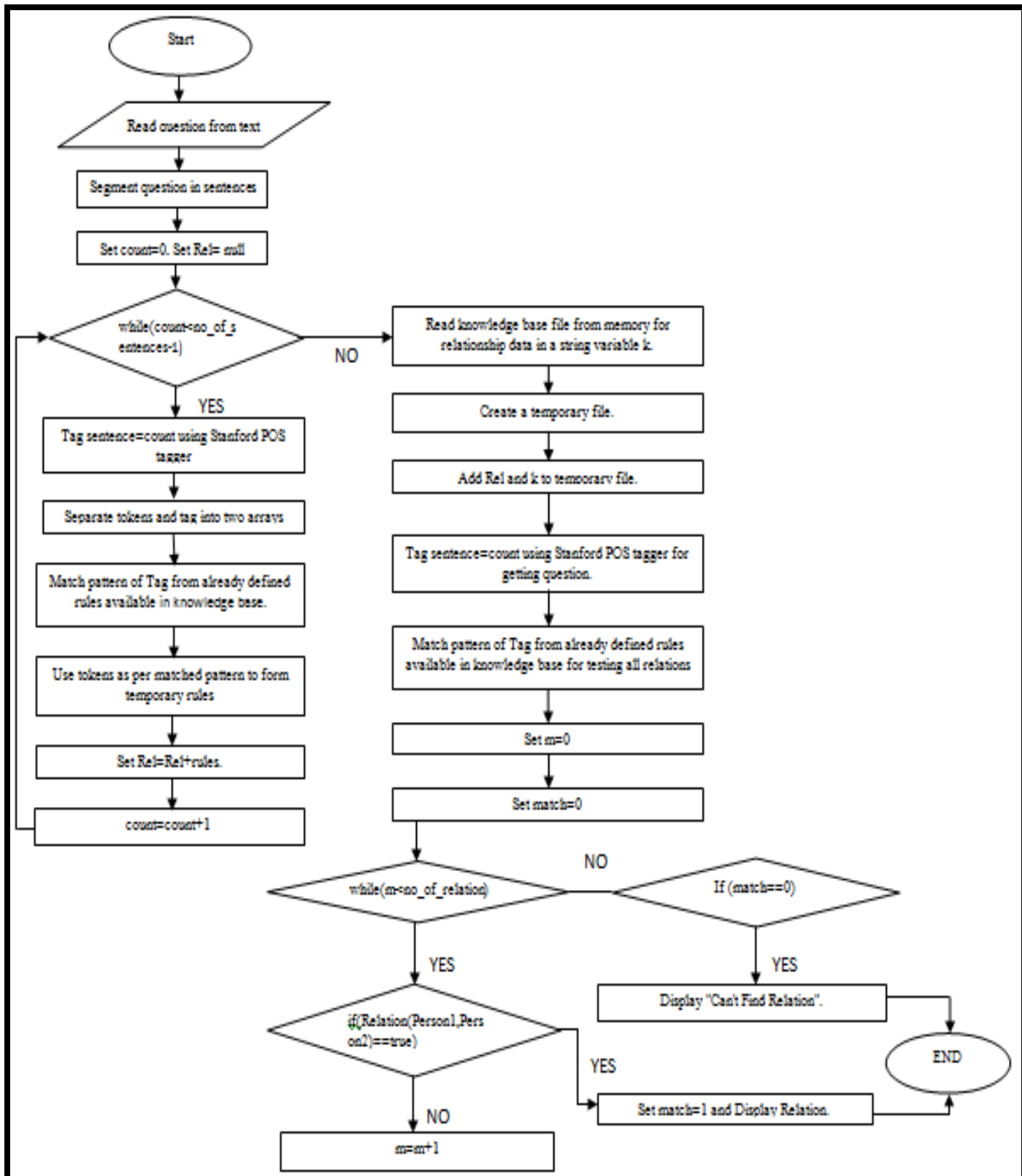


Figure 7: Flowchart for solving the problem

CHAPTER- 4

Results and Discussions

In this system, a blood relationship problem can be solved by using Stanford tool and PROLOG JPL libraries. This system consists of two modules. The user can choose any of the module depends upon the type of question. The interfaces of both the modules are shown in figure 7 and figure 8 respectively. By default, module 1 is display to the user. In module 1, the user enters the question in the Enter Question text area and then click on the Find Relation option. After click on the button, the relation will be displayed on the Relation text area and the Reset option gets enable. After click on the Reset option, the user is able to write the next question in the text field.

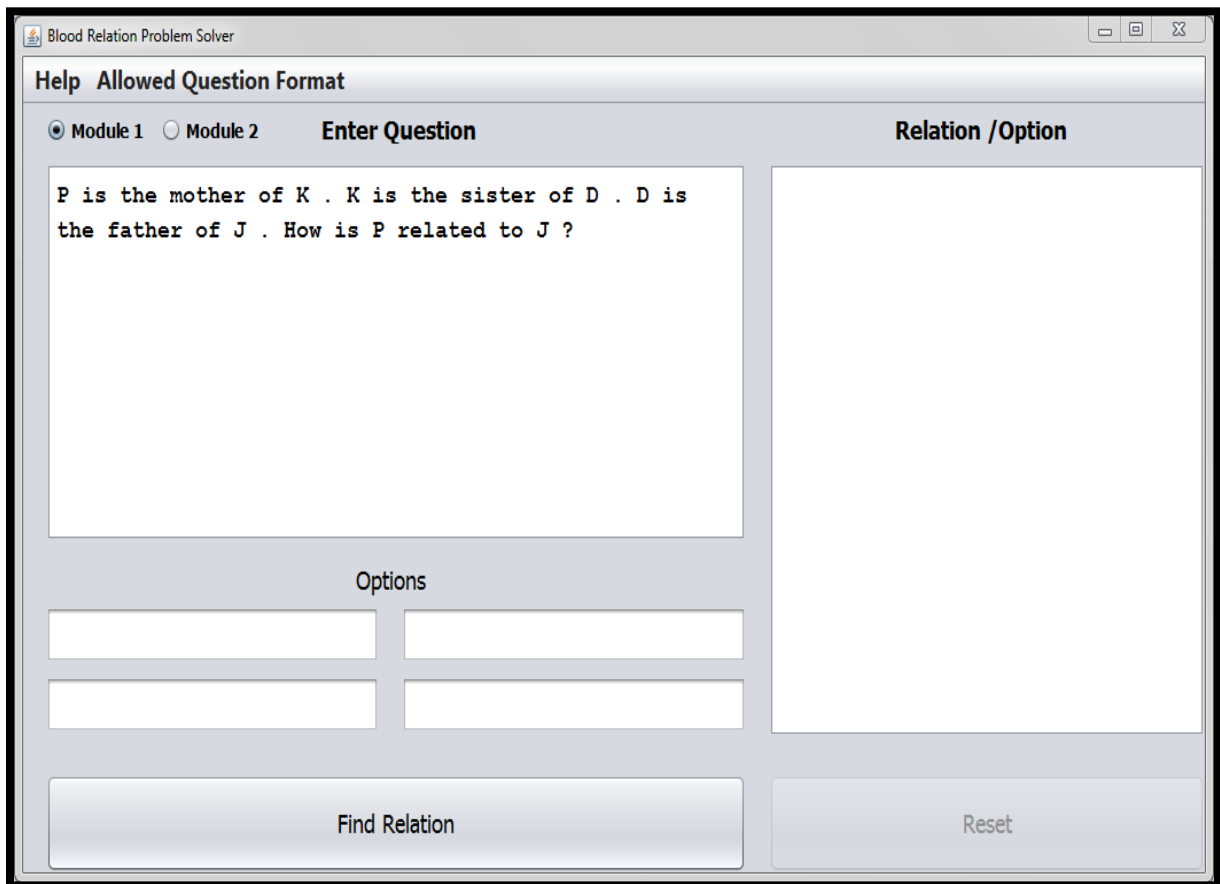


Figure 8: Interface of Module 1

Figure: 8 show the interface of module 2 and the type of the questions that can be solved by the system and entered by the user. In module 2, the user enters the question in the Enter Question text area and along with the options and then clicks on the Find Relation button. After click on the button, the relation will be displayed on the Relation text area and the Reset option gets enable. After click on the Reset option, the user is able to write the next question in the text field.

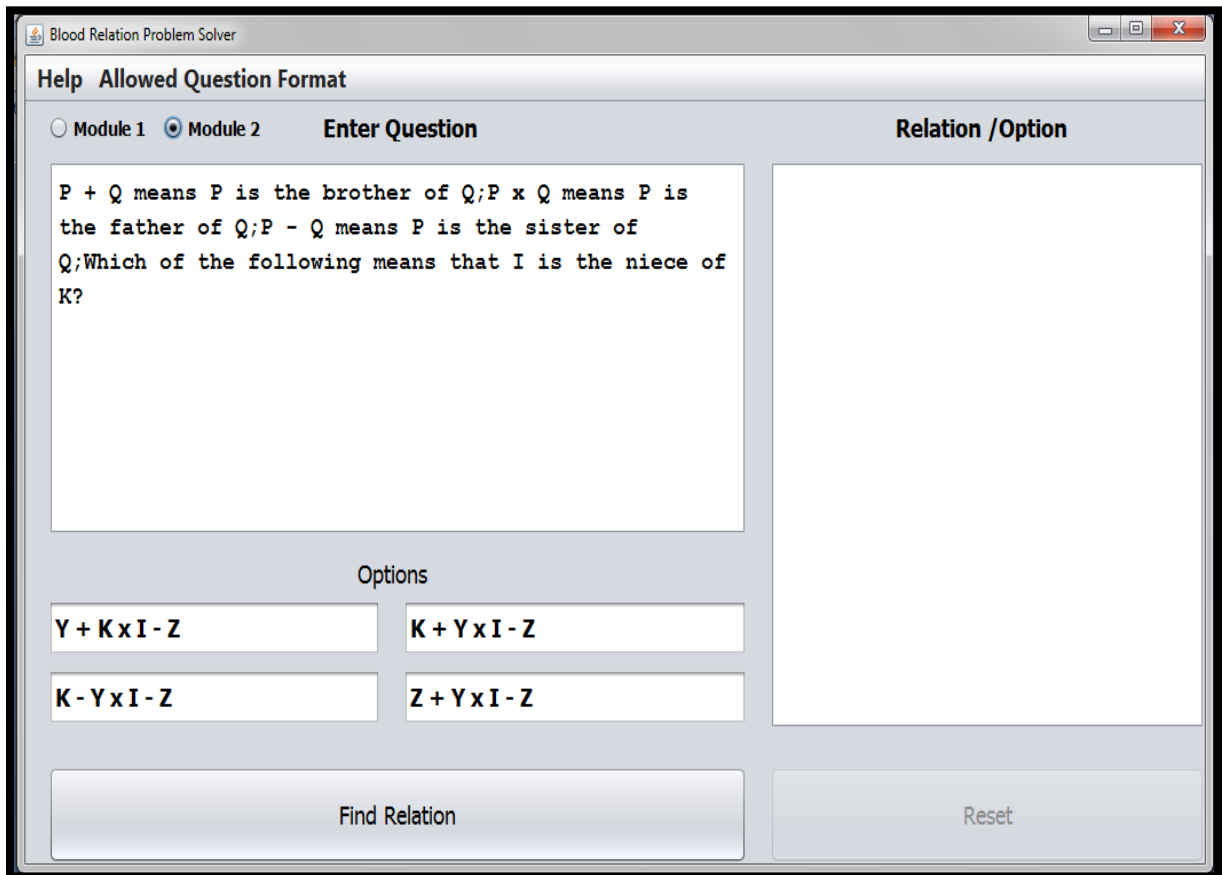


Figure 9: Interface of Module 2

Module 1: Type 1. Figure: 9 shows that the type of the questions that can be solved by the system and entered by the user. In this module, the user only enters the question in the Enter Question text area with no options added to it and then clicks on the Find Relation button. After click on the button, the relation will be displayed on the Relation text area and the Reset option gets enable. After click on the Reset option, the user is able to write the next question in the text field. The output for this relation is shown in figure: 10

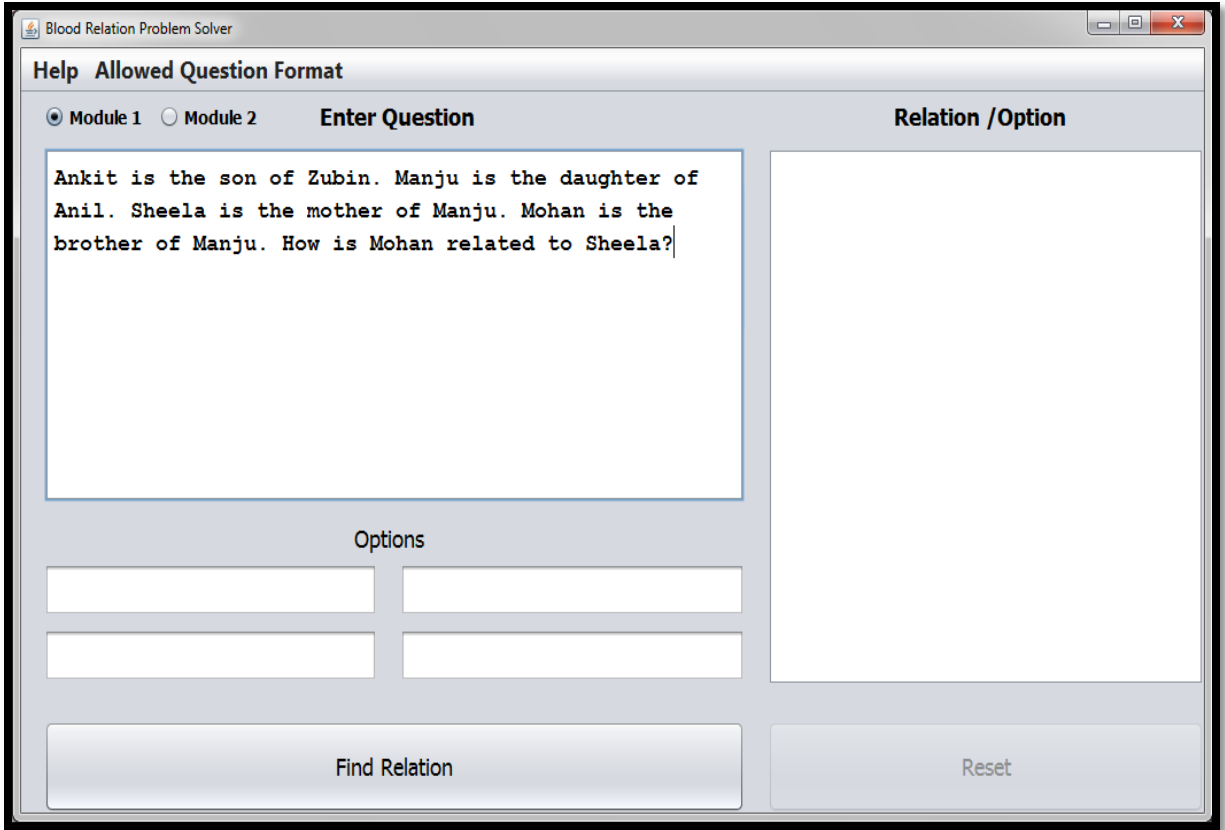
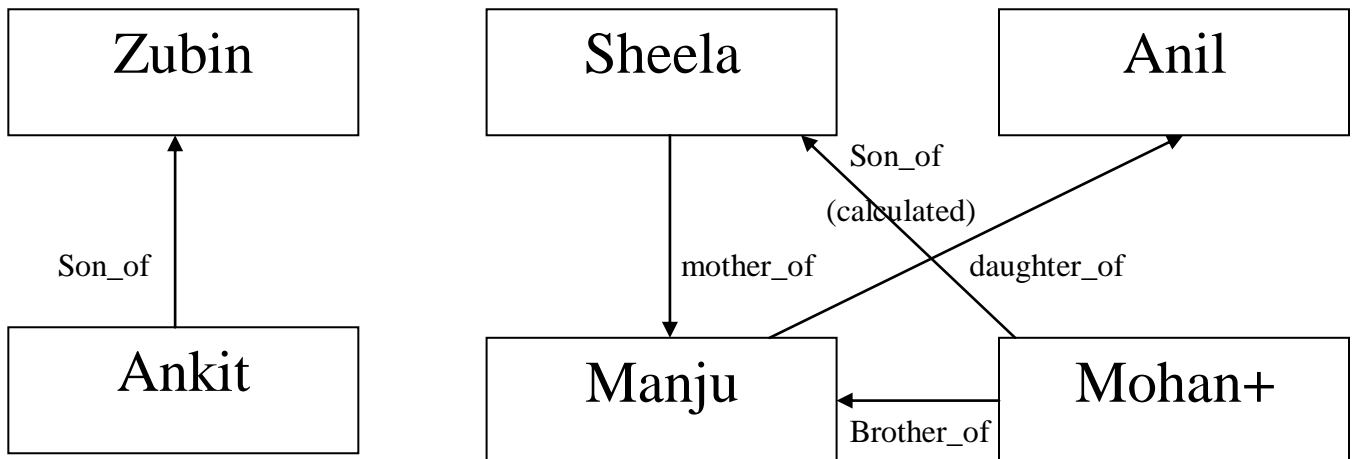


Figure 10: Input for Module: 1, Type: 1

Relationship Chart

(-) implies female

(+) implies male



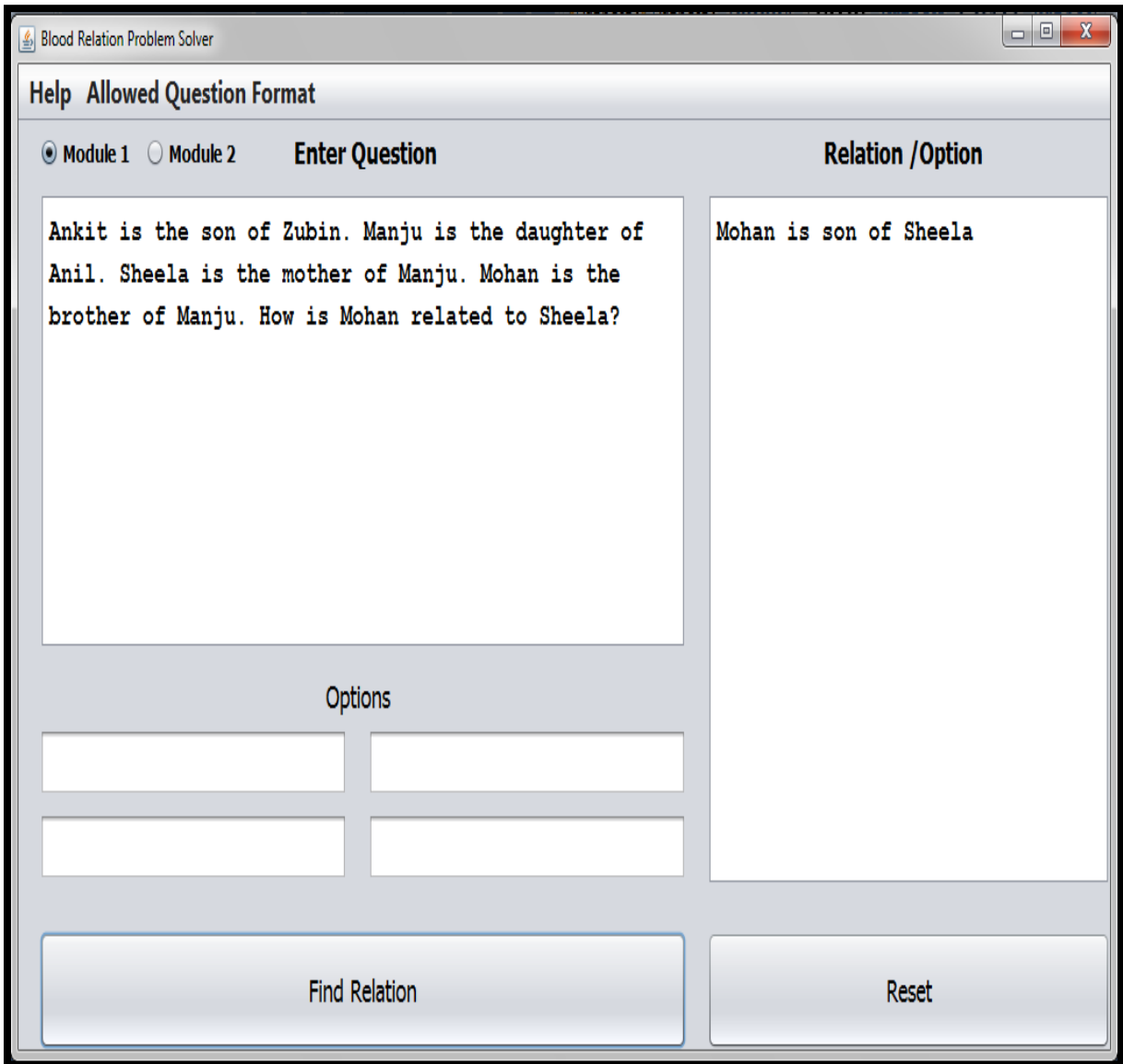


Figure 11: Output for Module: 1, Type: 1

Module 1: Type 2. Figure: 11 shows that the type of the questions that can be solved by the system and entered by the user. In this module, the user can enter the questions such as if A is B's brother instead of writing A is the brother of B. Since, there are pattern that matches the input sentence.

The user only enters the question in the Enter Question text area with no options added to it and then clicks on the Find Relation button.

After click on the button, the relation will be displayed on the Relation text area and the Reset option gets enable. After click on the Reset option, the user is able to write the next question in the text field.

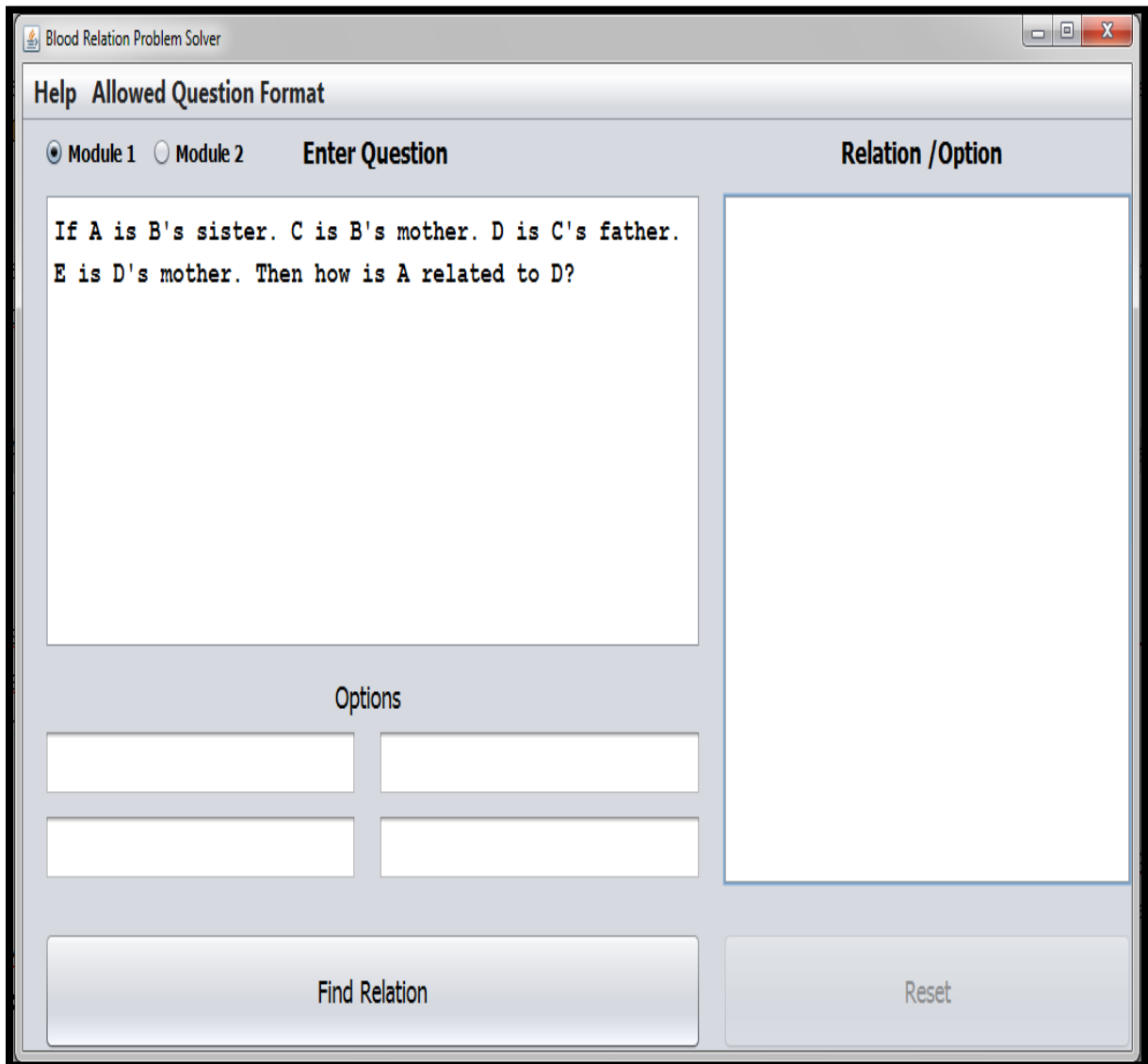
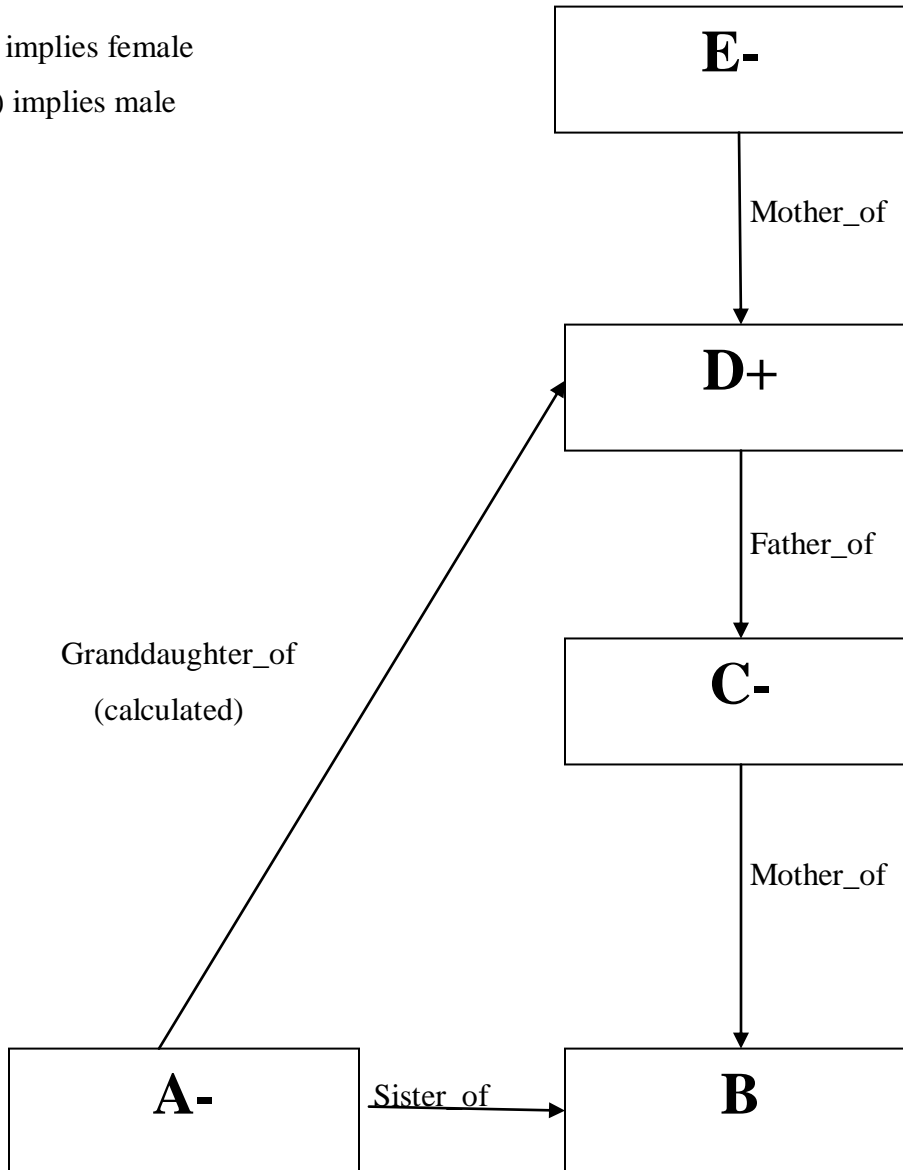


Figure 12: Input for Module: 1, Type: 2

Relationship Chart

(-) implies female

(+) implies male



The output for this relation is shown in figure 12:

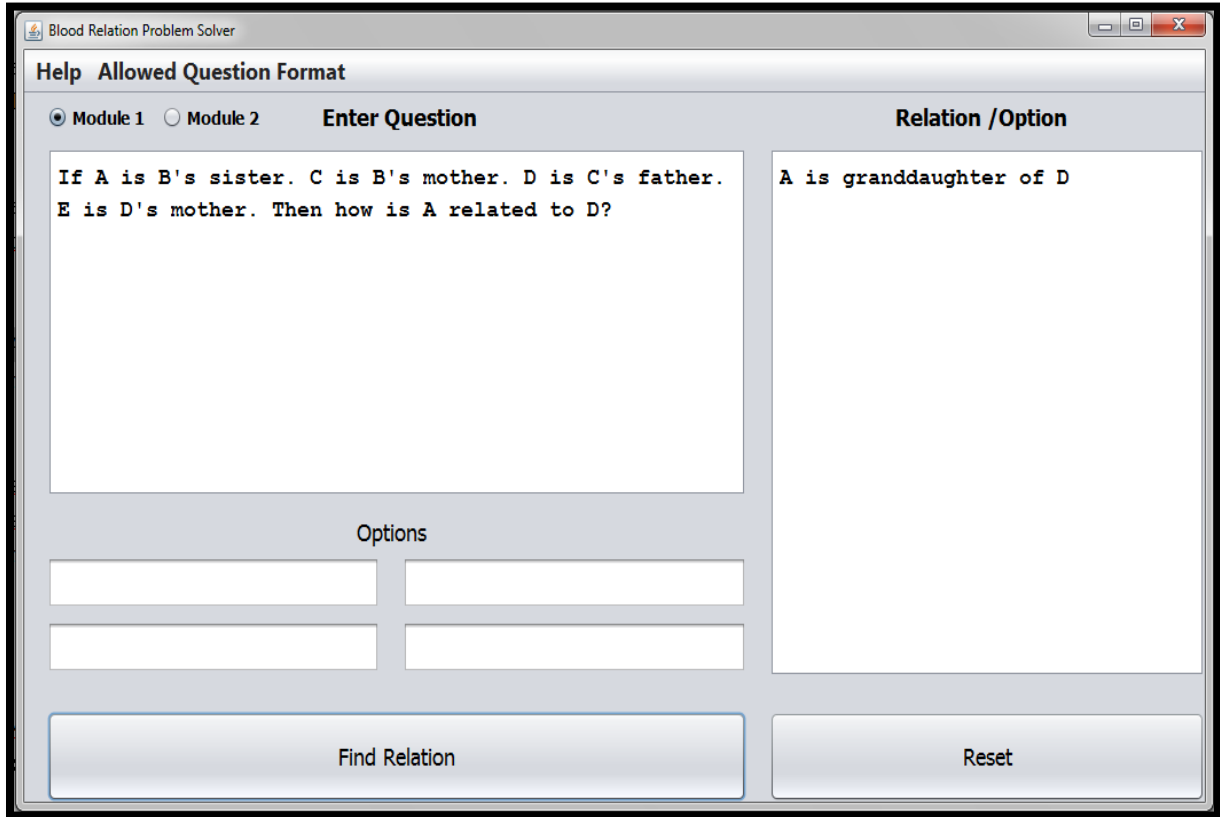


Figure 13: Output for Module: 1, Type: 2

Module 1: Type 2. Figure: 13 shows that the type of the questions that can be solved by the system. In this module, rather than finding the relation, the user can find the name of the person. The user has to enter the options in options field to find the same because the system can find the solution only with the help of options. So, it is mandatory to write the options to solve this type of questions. The user only enters the question in the Enter Question text area along with the options added to it and then clicks on the Find Relation button. After click on the button, the name of the person is display on the Relation text area and the Reset option gets enable. After click on the Reset option, the user is able to write the next question in the text field. The output for this relation is shown in figure 14:

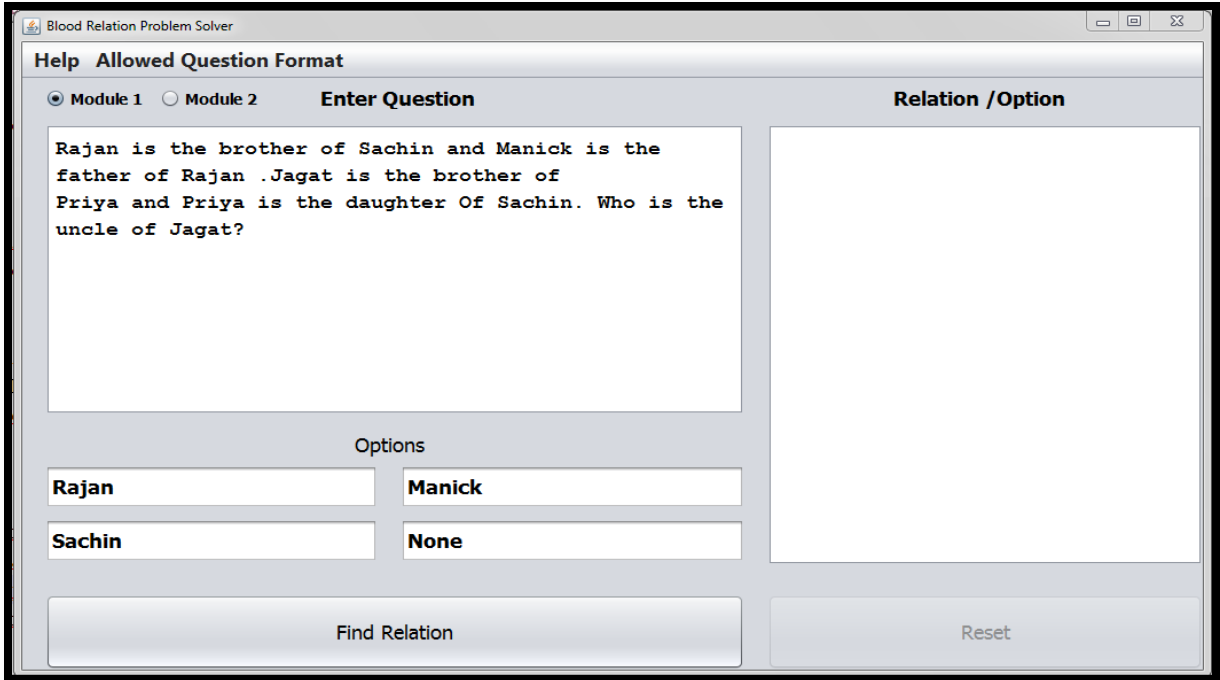
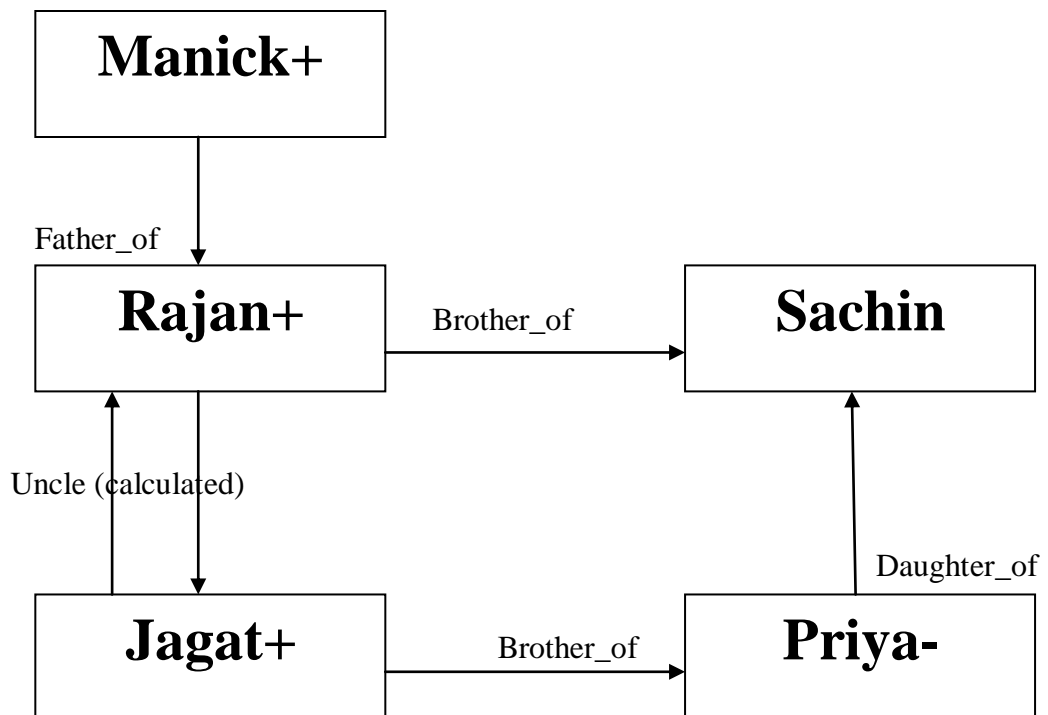


Figure 14: Input for Module: 1, Type: 3

Relationship Chart

(+) implies Male

(-) implies Female



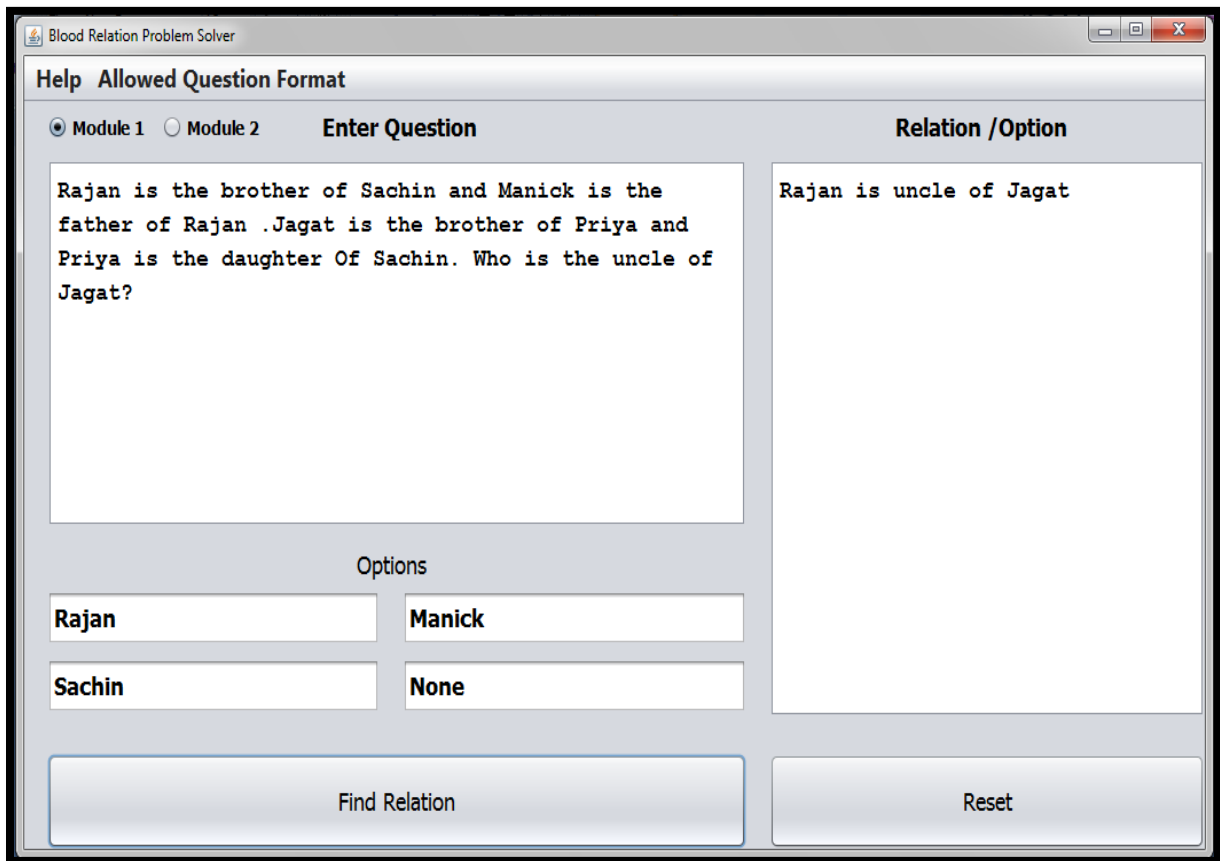


Figure 15: Output for Module: 1, Type: 3

Module : 2 Figure: 15 shows that the type of the questions that can be solved by the system. In this module, rather than finding the relation, the user can find the name of the person. The user has to enter the options in options field to find the same because the system can find the solution only with the help of options. So, it is mandatory to write the options to solve this type of questions.

The user only enters the question in the Enter Question text area along with the options added to it and then clicks on the Find Relation button. After click on the button, the name of the person is display on the Relation text area and the Reset option gets enable. After click on the Reset option, the user is able to write the next question in the text field. The output for this relation is shown in figure:

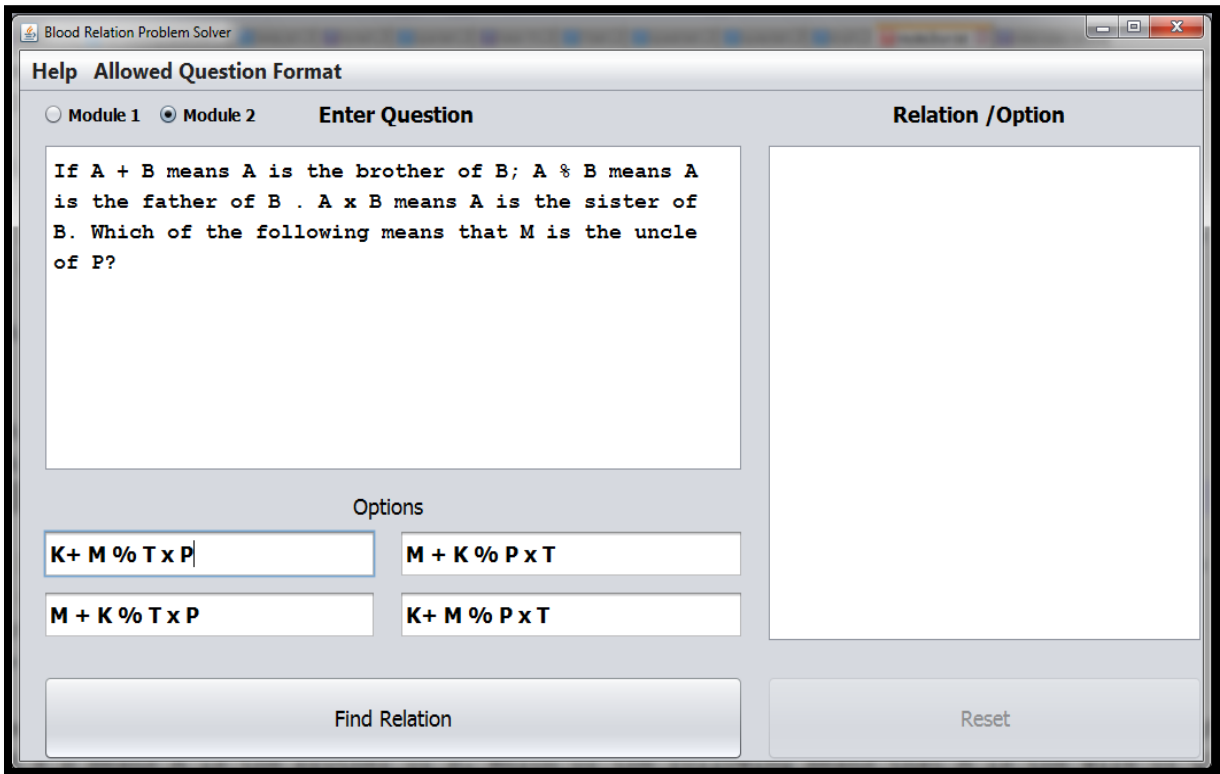
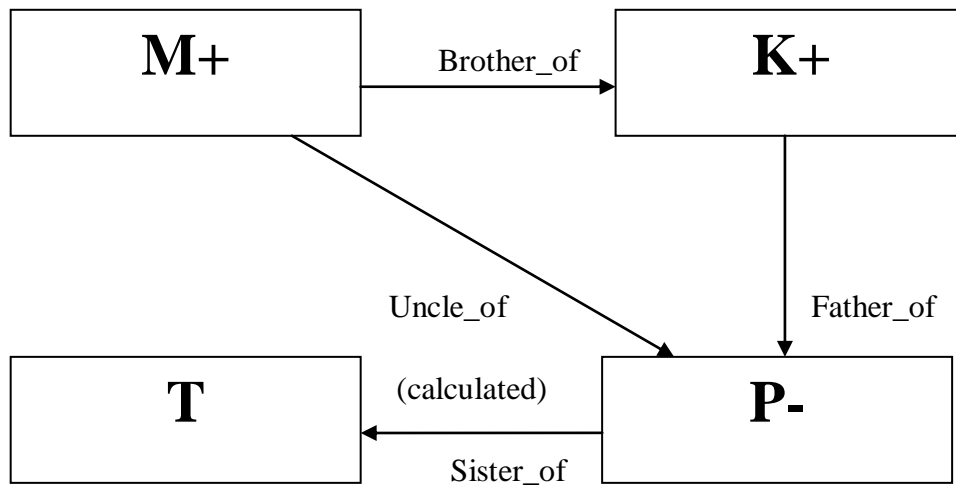


Figure 16: Input for Module: 2

Relationship Chart

(-) implies female

(+) implies male



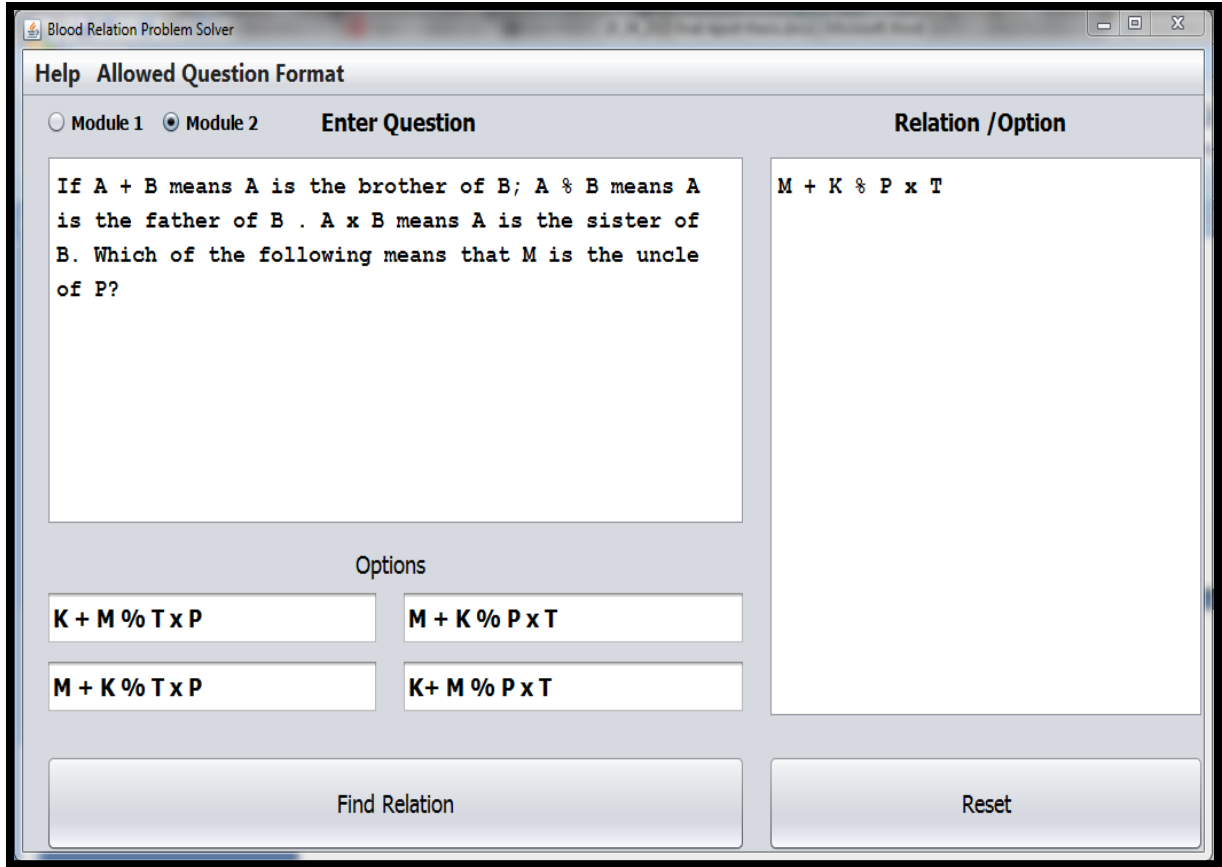


Figure 17: Output for Module: 2

Accuracy

Since the proposed system is fully dependent on Natural language processing and Stanford pos tagger so the accuracy of the system is the correct answer of questions tested over total number of questions tested.

$$\text{Accuracy} = \frac{\text{Correct answer of questions tested}}{\text{Total no of questions tested}}$$

Efficiency

The efficiency of the system entirely depends upon the number of types of questions running over total mo of available types of relationships questions. More rules may be added to the system for wider use.

$$\text{Efficiency} = \frac{\text{No. of types of questions running}}{\text{Total no. of available types of relationships questions}}$$

Table 2: Data sets for finding Accuracy and Efficiency of the system.

Module	No. of types of questions	Correct answer of questions
Module: 1	50	40
Module: 2	60	54

$$\text{Accuracy} = \frac{\text{No of questions correct}}{\text{No. of questions tested}} = \frac{94}{110} = 0.85 = 85\%$$

$$\begin{aligned} \text{Efficiency} &= \frac{\text{No. of types of question running}}{\text{Total no of available questions}} \\ &= \frac{4}{8} = 0.5 = 50\% \end{aligned}$$

CHAPTER- 5

CONCLUSION AND FUTURE SCOPE

Intelligent system plays a vital role in an education field. This report gives an overview of the expert system which is in progress. This thesis will provide the information about different relationships that may exist in a problem. It also provides the details about sentence tokenization, parts of speech tagging of a sentence, and by using the PROLOG rules that are already stored in the database, pattern matching is being done and finally the result is being displayed at the interface.

This report also provides details about how Natural Language Processing can make a machine understand different sentences and can interact with humans. This work can be extended to make a decision support system that will work in a similar manner like humans and can respond just like humans by understanding the different forms of sentences given to it as an input. By this new system, students can enjoy personalization education whenever and wherever possible. This expert system helps students to understand how to solve blood relationship problems in an efficient, easy and faster way.

The system can be used in practical applications as discussed earlier. In future, this work can be further extended to solve the different kind of problem related to blood relations. The system can be tested on most types of questions related to blood relations. It can be further improved by increasing different patterns and adding rules in the database.

I. Books

- [1]. Allen, James: Natural Language Understanding, University of Rochester, New York / Ontario / Wokingham, U.K. / Amsterdam / Bonn / Sydney / Singapore / Tokyo / Madrid/ San Juan 1995.
- [2]. D.W.Patterson, Introduction to AI & Expert Systems, Prentice Hall.
- [3]. Rich ,Knight,” Artificial Intelligence”, Tata McGraw Hill, 2009(Third edition)

II. Research papers

- [4]. Anatolii Vovk, Denys Girnyk. “Web-Based Notation of Mathematical Text Preserving Semantics for Scientific and Educational Communication”. The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 12-14 September 2013, Berlin, Germany.
- [5]. Barcala, Francisco-Mario, et al. "Tokenization and proper noun recognition for information retrieval." Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on. IEEE, 2002.
- [6]. Cinková, Silvie, Martin Holub, and Vincent Kríž. "Optimizing semantic granularity for NLP-report on a lexicographic experiment." Proceedings of the 15th EURALEX International Congress 2012. 2012.
- [7]. Fang, Ning, and Yongqing Guo. "A web-based interactive intelligent tutoring system for undergraduate engineering dynamics." Frontiers in Education Conference, IEEE, 2013.
- [8]. Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60
- [9]. Matsuda, Noboru, et al. "SimStudent: Building an Intelligent Tutoring System by Tutoring a Synthetic Student." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2008.

- [10]. M. El – Arabaty. "Knowledge Based Expert Systems Applied To Computational Techniques". Proceedings of the 1998 IEEE, ISIC/CIRNISAS Joint Conference Gaithersburg, MD September 14-17, 1998
- [11]. Mulkar-Mehta, Rutu, Jerry Hobbs, and Eduard Hovy. "Granularity in natural language discourse." Proceedings of the Ninth International Conference on Computational Semantics. Association for Computational Linguistics, 2011.
- [12]. Punyakanok, Vasin, Dan Roth, and Wen-tau Yih. "The importance of syntactic parsing and inference in semantic role labeling." Computational Linguistics 34.2 (2008): 257-287.
- [13]. Webster, Jonathan J., and Chunyu Kit. "Tokenization as the initial phase in NLP." Proceedings of the 14th conference on Computational linguistics-Volume 4. Association for Computational Linguistics, 1992.

III. Websites

- [14]. Natural Language Processing
https://en.wikipedia.org/wiki/Natural_language_processing
- [15]. Natural Language Processing,
http://www.mind.ilstu.edu/curriculum/protohiker/natural_language_processing.php
- [16]. <http://www.indiabix.com/verbal-reasoning/blood-relation-test/introduction>
- [17]. <http://www.wisegeek.com/what-is-an-inference-engine.htm>
- [18]. http://en.wikibooks.org/wiki/prolog/What_is_Prolog
- [19]. http://www.swi-prolog.org/packages/jpl/java_api/
- [20]. <http://nlp.stanford.edu/software/corenlp.shtml>
- [21]. <http://nlp.stanford.edu/software/tokenizer.shtml>
- [22]. <http://nlp.stanford.edu/software/tagger.shtml>
- [23]. http://en.wikipedia.org/wiki/Text_segmentation
- [24]. <http://www.javatpoint.com/java-regex>
- [25]. <http://en.wikipedia.org/wiki/Parsing>