A PROJECT REPORT ON

HINDI LANGUAGE INTERFACE TO DATABASES

TABLE OF CONTENTS

Chapter 1 : INTRODUCTION

This chapter presents the detailed problem definition of the project, it's scope and the hardware and software requirements.

1.1 Motivation :

- In this modern techno-crazy world, more and more laymen access various systems and applications through their smartphones and tablets.
- As more and more non-expert users are accessing relational databases, it is very important to simplify their process of accessing database records.
- Writing SQL queries can be difficult, especially when it involves complex logic.
- What if you could ask the question on your mind and we automatically convert it into a SQL- like format to retrieve the results?
- Moreover, keeping our country in mind, the percentage of English speakers in India is just 10%. That's 10% of a one billion-plus population!
- Most Indians have Hindi as their first language, followed by Marathi, Telugu, Punjabi, etc.
- Hence, there feels a need for a Hindi language interface to databases.

1.2 Problem Definition :

With the data growing exponentially day by day, there is a need to access the data in an efficient way to make the most use out of it. Natural language helps a novice to query the database in the preferred language (here, Hindi), this reduces the time and effort required to query the database such as MySQL since the person does not have to worry about the correct syntax behind a correct SQL query. Create a system which generates SQL queries from Hindi statements. The system should accept Hindi queries in both typed and voice commands. System should be generalized to work on all entity relationship models. Train the system in such a way that the exact column names and table names need not be mentioned by the user. Run the SQL queries and display the results.

1.3 Scope :

- Database will be a small version of a Library Database with book records, student records, etc.
- Hindi queries will be of interrogative type only.
- System will be trained on a smaller database and then on a broadened version and will be general for all entity relationship models.
- Input will be typed or a voice command.

1.4 Software / Hardware Requirements :

- 32 / 64 bit PC with windows 10,8,7 / macOS running
- Minimum 2 GB of RAM and 32 GN of storage
- Python 2.7 or higher
- NLTK and supplementary libraries installed

1.5 Methodology :

The proposed system will follow the following procedure :

1. Read Hindi command (via keyboard or voice).
2. Format the statement into processable form (tokenization).
3. Identify verb relations from the statement.
4. Form a dependency parse tree of the above relations.
5. Find database components (table names, column names) from semantic components.
6. Generate SQL query.
7. Run SQL query and display results.

1.6 Applications :

Hindi is a highly spoken language in north and central India, Pakistan, Fiji, Mauritius. Approximately six hundred million people speak Hindi as their first or second language. Hindi is also a national language of India. Large number of e-governance applications use databases. So to easily access the database query given by the user should be in

Hindi language. For this there should be a system that accepts the Hindi language, processes it and generates the SQL query and gives the desired results in Hindi language only. Following are some of the areas where the Hindi language interface can be applied.

- Railways : Hindi language interface to railway database system

  Hindi is the most common language used in India. So there is a need to develop Hindi language interface to the railway database system. The quarries of passengers such as reservations, timings, cancellation of journey etc., are able to be put in their native language. Passenger will give a query in Hindi and he is able to get results in Hindi language only.

- Agriculture : Hindi language interface to agriculture database system

  In India the whole population is dependable on agriculture. Farmers are generally not literate. They face lots of problems regarding irrigation, use of pesticides, time to reap the crop etc. Indian government has developed many systems to help farmers solve their queries. Data related to their queries is stored in databases but farmers are not that much literate to learn or know the SQL languages Hence the facility should be given so that the farmers can access the data from database it their own language so that they should utilize all the techniques and facilities related to agriculture.

- Weather forecasting and its related databases

  Weather report is necessary to each and every person. So providing a weather report in Hindi is very much important. Farmers may easily get the weather forecast report by asking a query in Hindi. This can be done because of the Hindi language interface to the weather forecasting database system.

- Legal matters : Hindi language interfaces to legal databases

  Lawyers' uses Hindi languages in their daily activities and hence they need to maintain their databases for clients. They just want a system which can process their request easily . The Hindi language interface to the legal database system is very useful.

- Employees : Hindi language interfaces to employee databases system

  Employees who are poor in query languages and English language, this Hindi language interface to employee database system helps them a lot

Chapter 2 : LITERATURE SURVEY

This chapter goes deep into the history of NLIDB systems, looks at the advantages and disadvantages of NLIDB.

2.1 Background :

Information storage in today's world is vastly dependent on relational databases. These databases offer the foundations for the systems like medical records, money markets, and electronic commerce. Using relational databases, a user can use a declarative language or 'Wh' type questions to describe the intended query.

The main reason behind using Natural Language Processing in any application is to make computers understand the human language either in the form of speech or text and perform the required operation. For beginners and novice who is unaware of the structure of queries and knows very little about the database query languages such as SQL, an easy approach of accessing and manipulating the data is asking questions to databases in natural language.

Automatic speech recognition is becoming more famous and hence is the reason that is being used widely in many applications. It is the process and related technology used to convert the speech input into its corresponding sequence of words or transcript. The user can interact with the database with their voice which undergoes the stages of refinement and then is passed to retrieve the details from the database. Thus, making it easier for novice users to interact with the system without having prior knowledge about the SQL queries.

The first attempts at NLP databases had been done many years back and are as old as any other NLP research. Accessing any query and information from the database in natural language is very user friendly, convenient and "free from worry" method to access the data, especially to other casual users who cannot understand database query language such as SQL.

One of the most wide and interesting areas of Natural Language Processing (NLP) is the development of a natural language interface to database systems (NLIDB). In the last few decades many NLIDB systems have been developed. Through these systems, users can interact with the database in a more convenient and flexible way. Because of this, this application of NLP is still very widely used today. Natural Language Interface has been a very interesting area of research since past times. The aim of Natural language Interface to Database is to provide an interface where users can interact with the database more easily using their natural language and access or retrieve their information using the same. We can also say that NLIDB is a system that converts the query in native language into SQL and vice-versa.

2.2 Advantages of NLIDB :

The NLIDB systems allow the people to communicate with databases in the same way they communicate with each other. The main advantages of Natural language Interface To Database are given below :

1. No requirement of Artificial Language : Users are not forced to learn an artificial communication language. Formal query languages like SQL are difficult to learn by non-computer specialists.
2. No need of Training : No special training is required before using the natural language interface. It is highly user friendly and easy to use by the end users.
3. Simple and easy to use : The natural language interface is very simple and easy to use because the end users write the query in their native language.
4. Better for some question : It has been argued that there are some kind of questions (e.g. questions involving negation, or quantification) that can be easily expressed in natural language, but that seem difficult (or at least tedious) to express using graphical or form-based interfaces.
5. Easy to use for multiple database tables : Queries that involve multiple database tables are difficult to form in graphical user interface as compared to natural language interface.

Disadvantages of NLIDB :

Many NLIDB systems have been developed so far for business purpose use but use of NLIDB systems is not broad - spread and it is not the primary choice for interfacing to databases. This lack of acceptance is mainly due to the large numbers of deficiencies which are given below:

1. Linguistics coverage is not obvious : Currently all NLIDB systems can understand some subsets of a natural language but it is quite difficult to define these subsets. Some NLIDB systems can't even handle certain queries belonging to their own subsets. This is not the case of formal language like SQL. Because the formal language coverage is obvious and provides the corresponding answers of any statements that follow the given rules.
2. Linguistics vs. conceptual failure : When the NLIDB system fails, it will not give any explanation of what causes the system to fail. Some users try to rephrase the question or just leave the question unanswered.
3. Inappropriate Medium : It has been argued that natural language is not an appropriate medium for communicating with a computer system. Natural language is claimed to be too verbose or too ambiguous for human - computer interaction.

2.3 Some Existing NLIDB Systems :

1. LUNAR System deals with the database which has the information about the specimen of rocks brought back from the moon. In this two databases are used one for chemical analysis and second for literature representation.
2. LADDER System6 is interfacing with the database which stored the data about the US NAVY ships. This system was developed in the late seventies (1978) which uses a worthwhile and meaningful grammar to analyze interrogations and doubts to query a distributed database.
3. CHAT-80 System interfacing the database which stored the data about the geographical facts (such as oceans, seas, rivers, countries, etc) basically it has the data of around 150 countries. CHAT-80 was introduced in the early eighties. This system was very optimal, result oriented, impressive and sophisticated. It was implemented in PROLOG which transformed an English Query into Prolog expressions and these expressions were evaluated against the Prolog.
4. PLANES System7 deals with the database which has the information about US NAVY 3-M (maintenance and material management) and aircraft maintenance and flight data. It was developed at the University of Illinois coordinated science laboratory in the late seventies.
5. The EUFID System consists of three important modules, without counting the DBMS. First being the analyzer module, followed by mapper module and the last being the translator module.
6. ITS (Intelligent tutoring system) assists the student or user in SQL without interruption of human teacher. It also provides immediate and customized instruction to learners, usually without interruption from human teachers.

7. The TEAM System was developed in 1987. A major section of that era was devoted to portability issues. Team was designed by the database administrator to be configured easily with least or no knowledge of NLIDBs.
8. DATALOG System based on cascaded ATN grammar, is an English database query system. It achieves a greater degree of portability and extendibility by providing distinct representation schemes for native language knowledge, general word knowledge and application domain knowledge.
9. START System uses several types of functions which are language dependent such as Parsing, Natural Language annotation to present the suitable information segments to the users.

Drawbacks of these systems :

- Although there is a little research going on, there is no proper well defined NLIDB system for Hindi language. Most of them are for English language.
- The grammar used for semantic parsing is mostly based on the subject-object relations or the agent-patient relations. Although it works for English, we find the *karaka* relations of the Computational Panini Grammar to be more efficient for Hindi language.

2.4 CPG Framework :

What is CPG ? :

- A rule based grammatical framework designed by Panini
- There are two lexical categories - nouns and verbs
- In CPG, relation between verbal nodes and its arguments(noun) called karaka relations considers information as central to the study of language.

In a general dependency model, there are two prominent lexical categories: verbs and nouns, which form verbal and nominal nodes. These nodes are connected by directed edges representing head modifier relations. Similarly, there are relations between words of other lexical categories as well.

In the CPG framework too, the verbs and the nouns are the two prominent categories which help in extracting the syntactic relations between the entities. According to it, every verbal root consists of an activity and a result. The activity consists of the actions carried out by the different participants involved in the action. The result is the state achieved on the completion of these actions. An action may consist of several sub-actions, each of which has its own semantic relation with their associated objects.

The karaka relations are syntactico-semantic relations between the verbals and their related constituents. At the karaka level, there are only six karaka relations defined between an action, represented as verbs, and its participants, represented as nominals. Even though these relations do not capture the innumerable types of semantic relations which can exist between a verb and its nominals, however, these do give the maximum necessary information relative to a verb and its nominals. Thus, they are sufficient for providing a mapping from karaka relations to semantic relations through an elegant yet compact mechanism.

Karaka Meanings :
K1 : Karta The most independent participant (usually doer of the action)
K2 : Karma Locus of the result of the action
K3 : Karna Instrument
K4 : Sampradana The beneficiary of the result
K5 : Apadan The stationary participant in the action involving separation
K6 : Adhikarna The locus of the action (space and time)

Why CPG ?

- Conventional systems use subject-object / agent-patient
- Here the semantic nature is difficult to extract
- Karaka relations are syntactico-semantic in nature
- Handles active passive voices
- CPG framework has syntactico-semantic nature, meaning CPG not only captures syntactic or lexical terms but also domain terms(which helps in connecting query with database schema) are both identified.

Conventionally, besides the CPG framework, text processing can also be done by identification of subject-object relations or a combination of subject-object with Thematic relations. However, we have chosen the CPG framework instead of the previously followed approaches because the karaka relations are different from subject-object type of relations as well as from agent-patient type of (Thematic) relations.

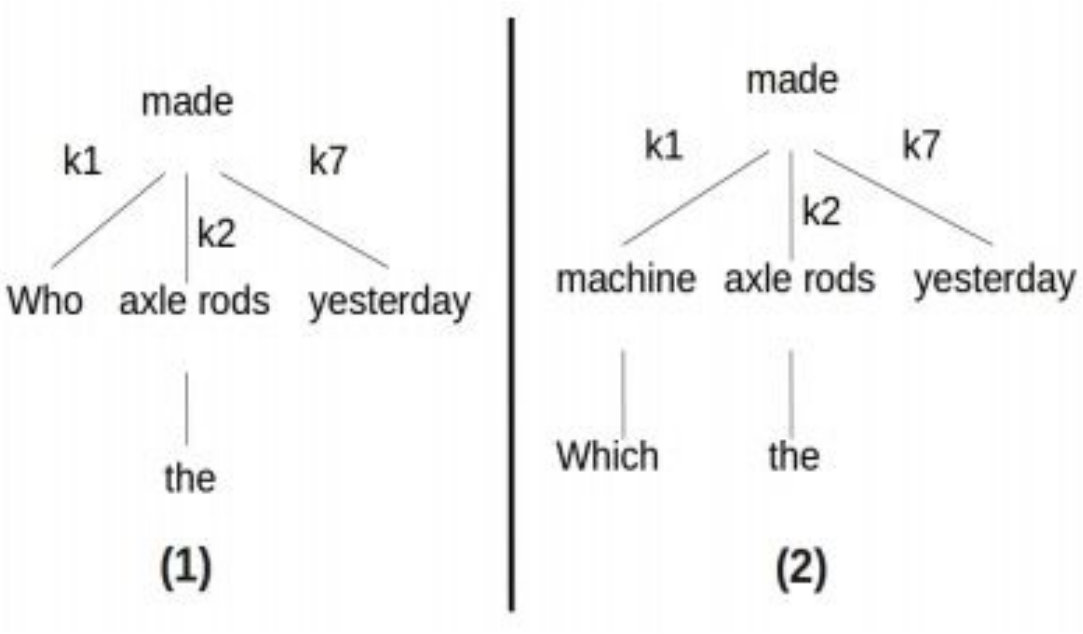
In subject-object relation based processing, the participating entities of the text are identified and then these entities are mapped to the database elements. Here, the semantic relations between the identified elements are difficult to establish. In the combination of subject-object relation with thematic role based processing, the participants are identified through the former and semantic relations, between these

participants, through the latter. In this approach developing a domain specific semantic parser is expensive in terms of time and the cost of development.

We elaborate the advantage of using the CPG framework over other approaches with the help of the following example:

If we have a database in the domain of ‘manufacturing’ where the database keeps track of batches of parts produced by a machine under the supervision of an operator, we might come across the following queries:

(1) Who made the axle rods yesterday?
(2) Which machine made the axle rods yesterday?



CPG Parse of Sentence (1) and (2)

In an approach based on subject-object type of analysis, the dependency roles filled by the question element are as shown in figure. These can be identified with relatively high accuracy by a general purpose (domain independent) robust parser, however, mapping these to the appropriate entity in the database, i.e. identifying domain semantics, requires greater effort with this variation.

Sentence	Question Element	Dependency Relation
(1)	who	subject
(2)	which machine	subject

Subject-Object based Dependency Relations

If a Theta role based approach is used, the correct output is as shown in table. Identifying such theta roles with high accuracy in a general purpose semantic parser is not possible today. As a result, such semantic parsers for a natural language become highly domain dependent and have to be built separately for each domain. This increases the effort needed to build NLIDB systems for each new domain and affects portability of the system.

Sentence	Question Element	Theta Role
(1)	who	agent
(2)	which machine	instrument

Theta Role based Semantic Relations

The CPG approach treads a middle ground which is linguistically sound and practically convenient. In the CPG approach, the relations are as shown in table. Note that in sentence (2), ‘machine’ is the karta (k1) because it is the most independent of the arguments of verbs (or participants in action). Thus, the CPG theory treats ‘person’ as k1 in (1) and ‘machine’ as k1 in (2). This ambiguity is retained in k1 and is to be disambiguated later using domain specific information.

As a result, disambiguation of certain hard things are systematically postponed to a later stage.

Sentence	Question Element	CPG Dependency Relation
(1)	who	K1 (karta)
(2)	which machine	K1 (karta)

CPG based Dependency Relations

Mapping to the semantic relations in the domain is an easier task starting from karaka analysis when compared to subject-object type of analysis. Most importantly, a broad coverage CPG parser can be used for parsing and the task of mapping to database or semantic elements is done using semantic frames.

Chapter 3 : PROJECT DESIGN

This chapter presents the detailed problem definition of the project, it's scope and the hardware and software requirements.

3.1 Proposed System Model :

The proposed system will follow 7 steps. Firstly, Hindi input will be taken from the users, either in the written form or via speaking, using speech to text. This Hindi statement will then be worked upon by tokenization, lemmatization, etc. modules. We will use the Stanford dependency parser to parse this query, which will give us a parse tree. This parse tree will be converted to a CPG parse tree by mapping the karaka relations using the karaka mapping table. This CPG parsed output will be used for the semantic analysis, where for each phrase, we will identify the domain elements and underlying relations with the help of semantic frames. The frames capture the semantics of language on one hand and the related domain concepts on the other. This will be used to solve ambiguities in the semantics. Finally, this will be converted into an SQL query using SQL Query Generator module. Query will be run on the database and output will be displayed.

Following is the procedure :

- Read Hindi commands (via keyboard or voice).
- Format the statement into processable form (tokenization).
- Identify verb relations from the statement.
- Form a dependency parse tree of the above relations.
- Find database components (table names, column names) from semantic components.
- Generate SQL query.
- Run SQL query and display results.

3.2 Software Project Management Plan

3.2.1 Overview

This section contains the software project management plan aimed at understanding the management project and to work with the various issues and to take care of changes and problems if any that may arise when the project effect goes awry.

Hindi Language Interface Database is a topic of much difficulty and has dazzled developers for a long period of time. Creating a system that gives the output by giving the input to a database in hindi language. The project delivered would be a website or an application capable of giving the output when the user gives input in Hindi language using NLP. The project spans for a period of almost 08 months from August 2020 to April 2021.

3.2.2 Included deliverables

A Semi prototype model will be delivered at the end as in this project main focus is on research work. Manage Databases for Hindi language and give output to the input given by the user in hindi language.The input that is in hindi language will be converted in sql query at backend and resulting output will be given. We will include all K1 types of karmas for our project.

Excluded

This project will exclude any training for using the software. The training of the model will require the intervention of engineers but handling the software is easy and should not require training. For the purpose of reference, a help manual can be provided.

3.2.3 Organization

The organization is simple structured with four members at the same level and interchangeably working in parallel with all the modules as and when they are vacant. The mentor guides with the line of thought and the design process and is responsible for scheduling meetings and evaluating the performances.

a. Project Manager

Role	**Name**
Project Manager	Naman Doshi
Technical Project Mgr.	Harsh Gupta Mohammad Athania Abdul Quadir

b. Project Team

1711001 : Abdul Quadir

1711002 : Mohammad Athania

1711012 : Naman Doshi

1711018 : Harsh Gupta

c. Steering Committee

The SteCo consists of the following members :

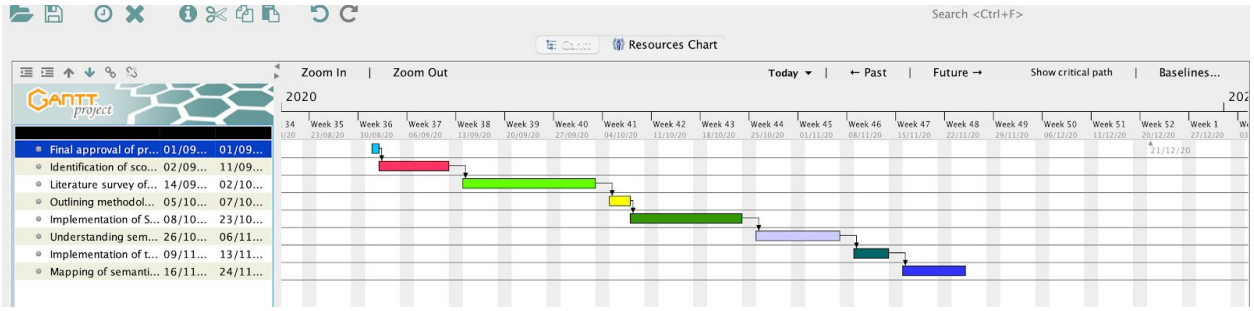
Organization	**Name**	**Comment**
KJSCE	Jyothi Rao	Project Mentor

Responsibilities SteCo :

- Determine whether the project is on point and within budget.
- Assess the quality of deliverables
- Question the project team based on the method of approach
- Suggest alternative methods as and when possible

3.2.4 Schedule and Budget

a. Schedule and Milestones



Duration	Tasks
Sept. First fortnight	• Final approval of proposal • Identification of scope and narrowing down to single application
Sept. Second fortnight	• Literature survey of NLIDB and CPG framework • Outlining methodology
Oct. First fortnight	• Implementation of Stanford dependency parser and CPG dependency parser of syntactic module • Understanding semantic analysis
Oct. Second fortnight	• Implementation of the semantic module
Nov. First fortnight	• Mapping of semantic components to database components

The objectives which will be achieved before VII semester examination :

- Literature survey of CPG framework and semantic module
- Implementation of syntactic (converting queries to dependency parser tree), semantic modules
- Mapping of semantic components to database components (tables, columns, etc.)

Timeline chart for project / thesis work completion

Duration	Tasks
Jan. Second fortnight	● Design UI outline ● Implementation of UI
Feb. First fortnight	● Implementation of text to speech ● Understanding of SQL generating module
Feb. Second fortnight	● Implementation of SQL generating module for generating SQL queries
March First fortnight	● Display of output after running query in required format ● Expansion of database
March Feb. Second fortnight	● Testing after integration of all modules and testing of UI and improvements
April First fortnight	● Final testing

April Second fortnight	● Final documentation
The objectives which will be achieved before VIII semester examination : ● UI for project will be completed ● SQL generating module for query generation and displaying of output will be completed ● Database will be expanded ● Final testing and documentation will be completed	

3.2.5 Budget

a. Budget and Price

There will be no high costing will be required as of now.Little bit expenses will be needed for server maintenance and hosting.

b. Variations, Changes, Contingencies

Costs are bound to fluctuate as we are going to find the best possible sets of algorithms and other software in the creation of the required software.

3.2.6 Development Process

For the development process the order for preprocessing is defined as follows

1. Read Hindi command (via keyboard or voice).
2. Format the statement into processable form (tokenization). Assign tokens to each and every word.
3. Identify verb relations from the statement.
4. Form a dependency parse tree of the above relations.
5. Find database components (table names, column names) from semantic components.
6. Generate SQL query.
7. Run SQL query and display results.

a. Development Environment

Item	**Applied for**
Methods	
Literature Survey	Requirements elicitation
SRS	Requirement,feasibility study
Tools	
dia	Design
nltk	Implementation
Languages	
UML	Design
Python	Implementation

3.2.7 Risk Management

All identified risks are documented, assessed and prioritized in the Risk Management Plan by the Project manager. The plan also defines the mitigation and contingency measures and who is responsible for. The Risk Management Plan is updated monthly

or on events and communicated to all affected stakeholders by the Project Manager. The risk status is reported to the line management in the monthly Project Report.

On discovering any risks, a meeting is conducted with the mentor who suggests the future course of action which can be used to plan future actions. Risks that are small are ignored, others are curbed or totally avoided if no plausible solution exists.

A risk is assessed based on its attributes i.e. the damage done and the likelihood of occurrence. A risk more likely to occur and does high damage must be tackled. A risk less likely but dangerous can be ignored.

3.2.8 Communication and Reporting

Type of Communication	**Method / Tool**	**Frequency/ Schedule**	**Information**	**Participants / Responsibilities**
Internal Communication:				
Project Meetings	Face to face Online Meeting on Zoom	Weekly and on event	Project status, problems, risks, changed requirements	Project Manager Project Team
Sharing of project data	Documents Google Drive	When available	All project documentation and reports	Project Manager Project Team Members
Milestone Meetings	Face to face Online Meeting on Zoom	Before milestones	Project status (progress)	Project team

Final Project Meeting	Face to face	Final milestone of project completion	Wrap-up Experiences	Project Manager Project Team
External Communication and Reporting:				

Project Report	Excel sheet	Monthly	Project status - progress - forecast - risks	Project Manager Sub-Project Managers
SteCo Meetings	Face to face	Quarterly	Validating the project and putting forward changes if any	Project Manager, SteCo

3.2.9 Delivery Plan

a. Deliverables and Receivers

Ident.	**Deliverable**	**Planned Date**	**Receiver**

D1	Pre processed raw code which checks the query for syntactic correctness	1-11-2019	Steering committee
D2	Trained model that does semantic analysis	15-2-2020	Steering committee
D3	Intermediate product capable of generating queries	1-3-2020	Steering committee
D4	Final project delivered	30-4-2020	Deployed to users

3.2.10 Quality Assurance

The quality assurance testing will be done as beta testing will be carried out. The user is guaranteed that the inputted data will generate appropriate output . The users are expected to know Hindi Language as it will be required for input purpose. The implementation of all the modules is based on the most efficient methods possible and is a backbone for the website/Application providing better quality to the users.

3.2.11 Configuration and Change Management

The changes will be put forward by the project steering committee (SteCo) in every quarterly meeting. The changes to be incorporated will be put forth after proper discussion between the project members and the mentor. The changes will then be incorporated by first designing a specification for changes and then using those changes to create a better version of the code or design modifications.

3.2.12 Security Aspects

In regards to the confidentiality and integrity of data, the system is liable for protecting the data. The authentication system must ensure that the database is not hacked and data is not stolen.

On breach of security, the system should notify the developers about the situation and the developers must inform the users, taking full responsibility for the breach and then take necessary action to stop and then reduce the damage caused. The information is one way i.e. from organization to the server and only the results (scores) are given back from the server to the users.

3.3 Software Requirement Specification Document

3.3.1 Introduction

a. Purpose

With the data growing exponentially day by day, there is a need to access the data in an efficient way to make the most use out of it. Natural language helps a novice to query the database in the preferred language (here, Hindi), this reduces the time and effort required to query the database such as MySQL since the person does not have to worry about the correct syntax behind a correct SQL query. Create a system which generates SQL queries from Hindi statements. The system should accept Hindi queries in both typed and voice commands.

b. Product Scope

- Database will be a small version of a Library Database with book records, student records, etc.
- Hindi queries will be of interrogative type only.
- System will be trained on a smaller database and then on a broadened version and will be general for all entity relationship models.
- Input will be typed or a voice command.

3.3.2 Overall Description

In this modern techno-crazy world, more and more laymen access various systems and applications through their smartphones and tablets. As more and more non-expert users are accessing relational databases, it is very important to simplify their process of accessing database records. Writing SQL queries can be difficult, especially when it involves complex logic. What if you could ask the question on your mind and we automatically convert it into a SQL- like format to retrieve the results?

Moreover, keeping our country in mind, the percentage of English speakers in India is just 10%. That's 10% of a one billion-plus population! Most Indians have Hindi as their first language, followed by Marathi, Telugu, Punjabi, etc. Hence, there feels a need for a Hindi language interface to databases.

a. User Documentation

The user-interface of the website will be very user friendly, and highly intuitive and would require little to no need of help and tutorial. The site would include a help section to inform the users and help them understand how to use the website and navigate through it.

b. Assumptions and Dependencies

Assumptions:

In order to keep the scope lower, we assume the initial conditions as stated in the scope but will change them as we scale the product upwards to cover more SQL query types and wider schemas.

Dependencies:

Additional libraries and tools will be used for software project monitoring and implementing complex features like thematic analysis (machine learning) and analyzing undesirables (deep learning).

3.3.3 External Interface Requirements

a. User Interfaces

User interfaces would be provided for users. There will be login provisions. Users will have to write their queries in the text box or click on microphone button to speak queries. They will receive the output..

b. Hardware Interfaces

- 32 / 64 bit PC with windows 10,8,7 / macOS running
- Minimum 2 GB of RAM and 32 GN of storage

c. Software Interfaces

- User to Website: Web Browser(modern), Operating System(any).
- Website to Database
- Python 2.7 or higher
- Web Server: Localhost/XAMPP server.
- Languages: Python, HTML5, CSS3, JS.
- APIs and Libraries: NLTK and supplementary libraries installed

d. Communications Interfaces

The HTTP protocol will be used for communication over the internet. If the website will be monetized in future then HTTPS and WWW can be used.

3.3.4 System Features

Natural Language To SQL Query

- Natural Language query will be converted to SQL queries
- This query will be run and output will be displayed

3.3.5 Functional Requirements

The user needs to have enabled Javascript on his browser. The browser must also be updated to the latest update for the features to work. In case of any issues with this feature the user can contact the developers through email.

REQ-1: Enabled Javascript on the browser

REQ-2: Browser updated to its version.

3.3.6 Other Nonfunctional Requirements

a. Performance Requirements

Since servers are maintained at regular intervals the database is highly optimized and data retrieval is much faster.

b. Safety Requirements

The data, for example product ID, price, phone number, password, etc should be kept encrypted in a database to prevent illegal access to anyone not authorized. Care must be taken so that SQL injection does not take place.

c. Security Requirements

Ensuring confidentiality and availability of databases at all times is important. Integrity checks will be carried out from time to time.

d. Software Quality Attributes

Adaptability: The developers strive to help create a website that is both beautiful to perceive and adaptable to changing times. The website will in time be made responsive to work on all kinds of devices available in the market.

Availability: The website will be available to every person across the entire globe on every modern browser.

Robustness: The organization plans to strategize back-end solutions to strengthen its foundation and thus provide seamless usability to the average user.

Testability: There is a need to maximize sticky architectures and facilitate one-to-one communities to aggregate efficient markets and accumulate usage statistics to further improve the website. This shall be done by launching the system in a closed alpha phase accessible to only a very small minority of users for efficient testing process.

e. Business Rules

The database admins will be informed to maintain the seamless working of the database system for an efficient back-end handling and server functionality. Any loss of data will not be our responsibility. This system will be only used for getting data, not

modifying data from the database. System must be updated from time to time for security updates.

3.3.7 Glossary

NLP: The field of study that focuses on the interactions between human language and computers is called Natural Language Processing, or NLP for short. It sits at the intersection of computer science, artificial intelligence, and computational linguistics (Wikipedia).

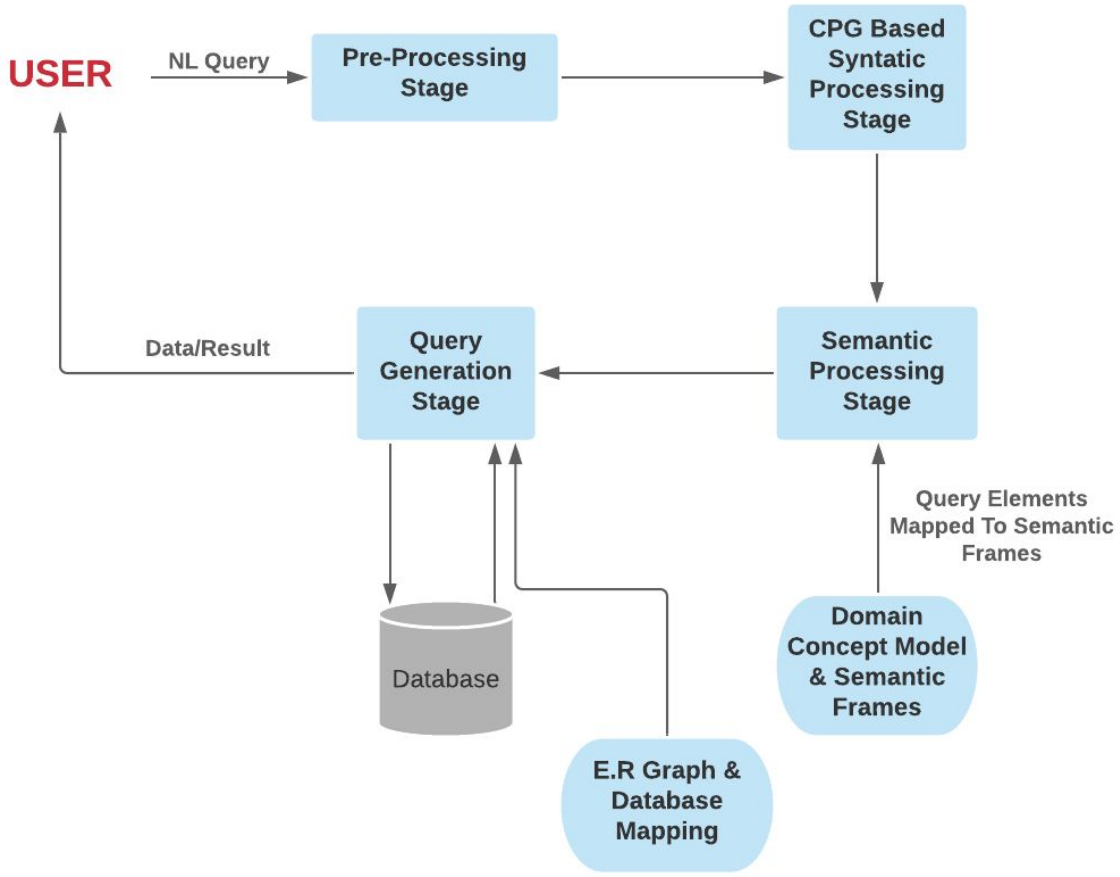
Natural Language Processing is a field that covers computer understanding and manipulation of human language, and it's ripe with possibilityis for newsgathering. Anthony Pesce said in Natural Language Processing. It is usually used in the context of analyzing large pools of legislation or other document sets, attempting to discover patterns.

NLTK: The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook.

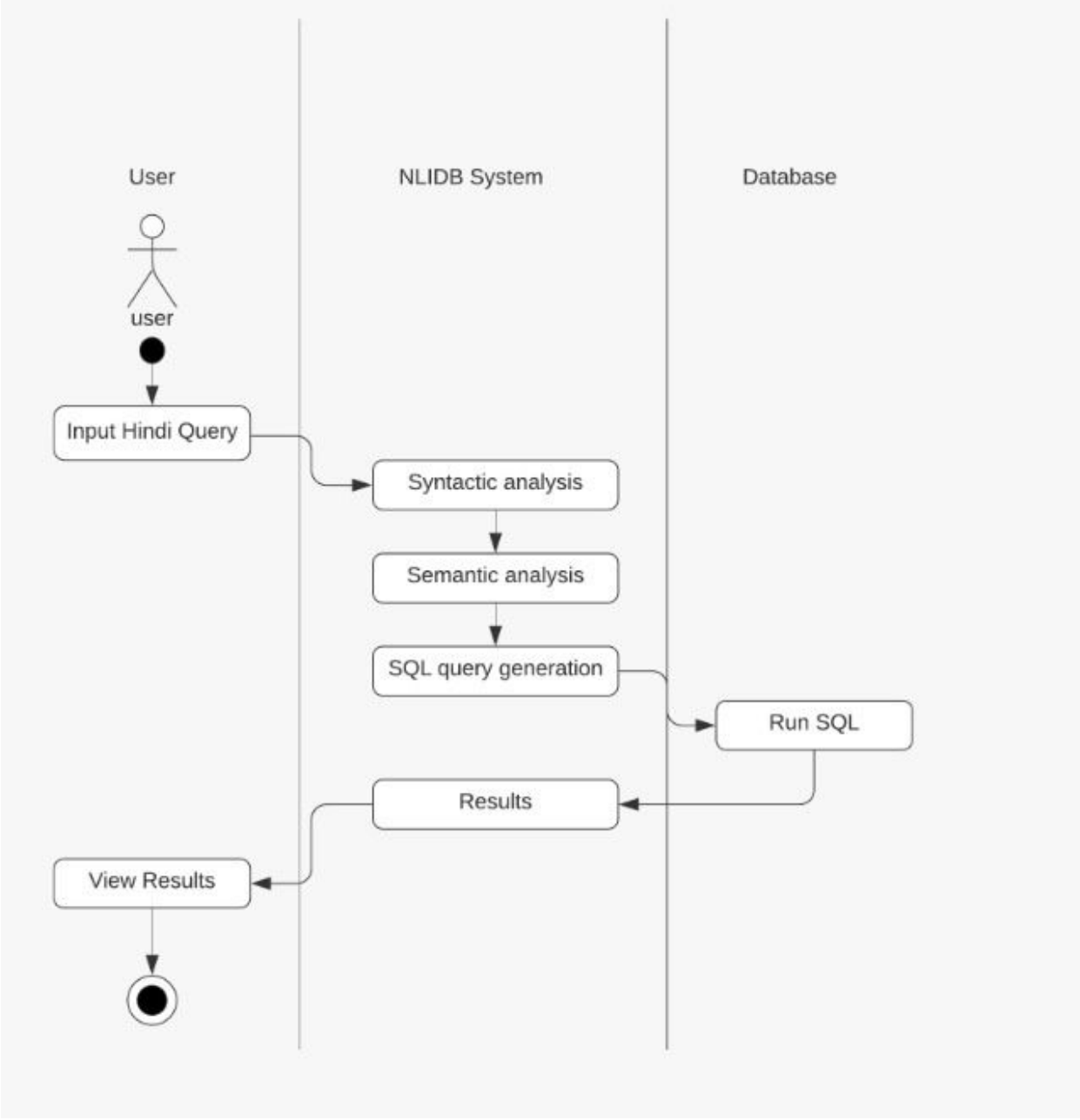
CPG : A rule based grammatical framework designed by Panini. There are two lexical categories - nouns and verbs. In CPG, relation between verbal nodes and its arguments(noun) called karaka relations considers information as central to the study of language.

3.4 Software Design Document

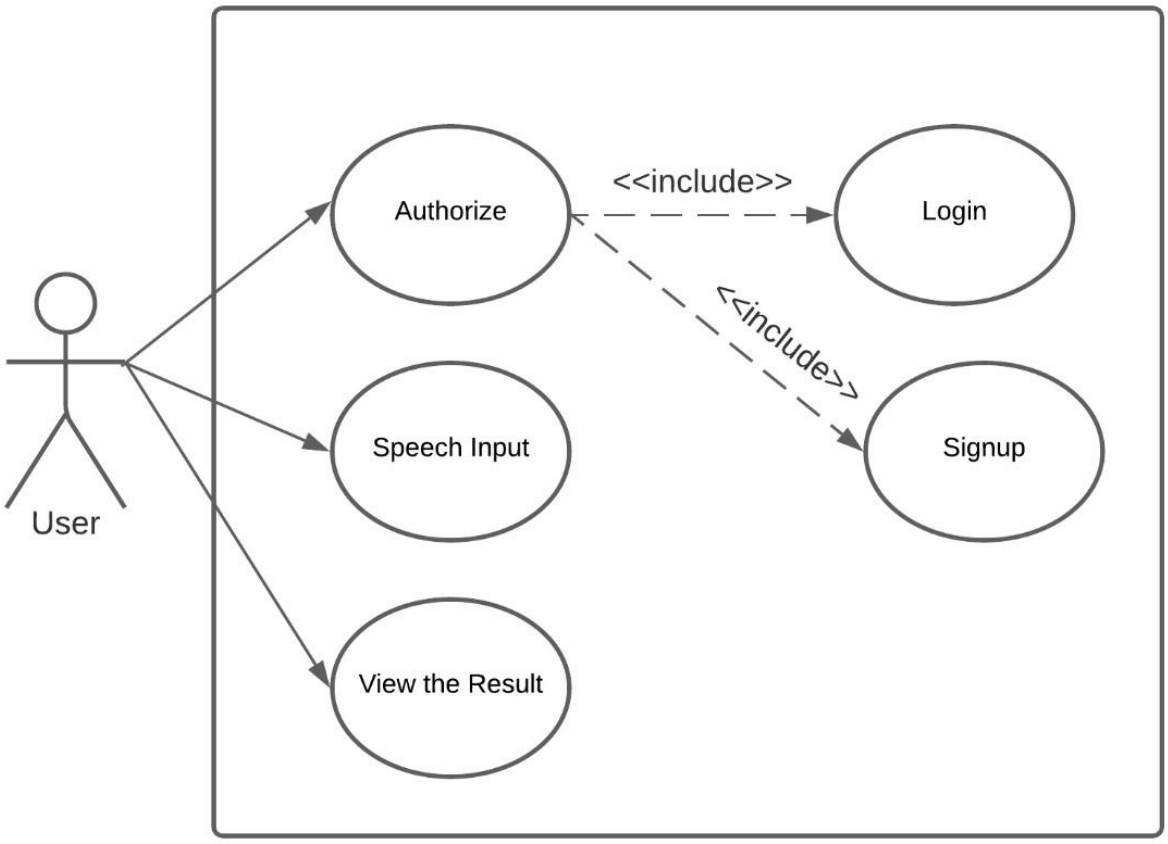
a. Workflow



b. Activity Diagram



c. Use Case Diagram



Chapter 4 : IMPLEMENTATION

This chapter presents the prototype model of proposed system model implementation, inclusion of any additional details as suggested by project guide/during progress seminar and experimental results and its analysis.

Following is the prototype of the syntactic analysis of the system :

1. **Natural Language Query Input**

   Users will be able to give Hindi queries via voice commands. These queries are natural language queries with no need of any knowledge of SQL. These queries should follow a proper template. (The Hindi queries should be interrogative and in active voice as of now).

```
1 query = 'कौन से छात्रों ने एनएलपी लिया ?'
```

2. **Tokenize and Normalize Query**

   Next step is to tokenize the query into separate phrases followed by removal of stop words from the query.

```
In [11]: from tokenizer import Tokenizer
         t=Tokenizer("कौन से छात्रों ने एनएलपी लिया?")

In [12]: t.tokenize()
         t.print_tokens()
         type(t)
```

कौन
से
छात्रों
ने
एनएलपी
लिया?

Tokenization

After removal of stop words :

```
1 query = 'कौन छात्रों एनएलपी लिया'
```

Removal Of Stop Words

3. **Syntactic Parsing Module**
   POS tags are found using Stanford NLP and corresponding dependency parse tree is formed.

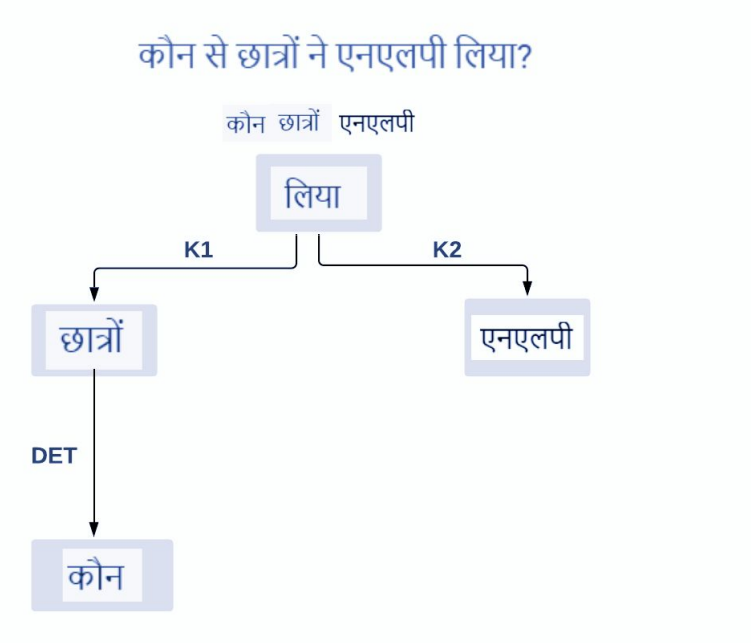
	word	pos	exp
0	कौन	WQ	NA
1	छात्रों	NN	noun, singular 'desk'
2	एनएलपी	NNP	proper noun, singular 'Harrison'
3	लिया	VM	main verb

POS Tags

```
('कौन', '2', 'det')
('छात्रों', '4', 'nsubj')
('एनएलपी', '4', 'nsubj')
('लिया', '0', 'root')
```

Stanford Dependency Parser

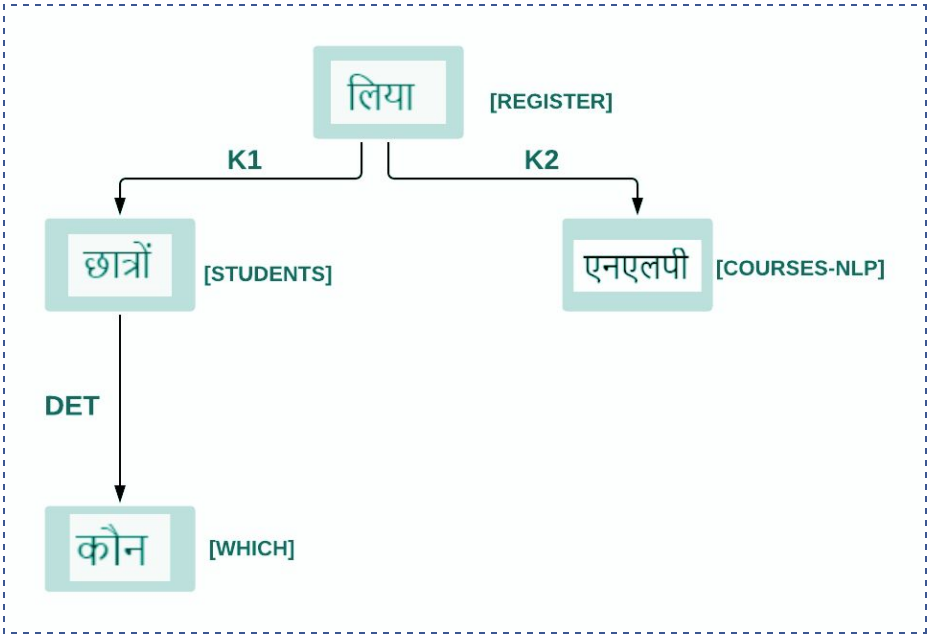
This is converted to corresponding CPG dependency parse tree by mapping the *karaka* relations.



k1 : *Karata* denotes the agent who is doing the action for the verb (*karama*).

k2 : Many dependencies can be mapped to k2. Dependencies like nsubj, dobj, iobj, prep_for can be used for k2.

4. **Semantic Mapping Module**
   From the CPG dependency parser we map the semantic frames to the nodes. We find the column names (noun frames) and appropriate action to be performed (verb frames). The table names, column names and conditions are searched in the database table's list.



Semantic Mapping

5. **SQL Query Generator Module**
   From the output of the semantic mapping module and from the entity relationship diagram of the database, a query is generated.This query is executed and the result is displayed.

   Query :

```
SELECT student.name
FROM student, register, course
WHERE
        course.name = 'NLP' AND
        student.ID = register.ID AND
        register.ID = course.ID
```

Chapter 5 : CONCLUSION AND FUTURE WORK

This chapter presents the conclusion of this report, future work for the system and references to the technicalities mentioned.

5.1 Conclusion

Natural language helps a novice to query the database in the preferred language (here, Hindi), this reduces the time and effort required to query the database such as MySQL since the person does not have to worry about the correct syntax behind a correct SQL query. The system accepts Hindi queries in both typed and voice commands. Use of CPG framework for Hindi language majorly increases the performance and correctness of the system. We have laid the groundwork for successful conversion from natural language to SQL. Further scaling and semantic analysis will be done soon.

5.2 Future Work

The system can be further made useful by adding more types of queries, hence increasing the scope of the project. System should be able to accept much wider database schemas comparatively. UI can be made accessible to specially abled people as well. In the future, the whole system can be made online as well.

5.3 References

- Computational Paninian Grammar Framework by Akshar Bharati and Rajeev Sangal, 2009 [1]
- Bridging the Semantic Gap with SQL Query Logs in Natural Language Interfaces to Databases by Christopher Baik, H. V. Jagadish, 2019 [2]
- A Novel Approach for Identification of Karaka Relations using Semantic Role Labeling Method by Amita, Ajay Jangra, 2015 [3]
- Universal Dependencies : https://universaldependencies.org/u/pos/all.html#al-u-pos/DET [4]
- University of oxford style guide for grammar rules [5]
- Stanford NLP : https://nlp.stanford.edu [6]