CHAPTER – 2

Study related to Natural Language Interface to Database

2.1 Survey on Natural Language Processing (NLP) Applications
2.2 Survey on Natural Language Interface to Database (NLIDB)
2.3 Summary
2.1 SURVEY ON NATURAL LANGUAGE PROCESSING (NLP) APPLICATIONS

The chapter discusses the literature review on NLP applications such as Information Retrieval System, Question Answering System, Dialogue System and Natural Language Interface to Database Systems.

2.1.1 Information Retrieval (IR) System

The Information Retrieval system is a scientific discipline which deals with analysis, design and implementation of a computerized system that addresses representation, organization, and access to large amounts of heterogeneous information encoded in a digital format. The search engine is the well known application of IR which accepts query from user and returns the relevant document to user. It returns the document, not the relevant answers; users are left to extract answers from the returned documents. The figure 2.1 represents Information Retrieval Model and its vertical taxonomy [33].

![Figure 2.1: Vertical Taxonomy of Information Retrieval model.](image-url)
2.1.2 Question Answering (QA) System

It is a specialized case of Information Retrieval System. It is a process of finding specific answers of a question posted by user from a large collection of text. The user can input question in natural language and receive concise answer. The dimension of Question Answering system is based on different sources to generate an answer. For example, it can be an open or closed domain Question Answering system or it may be syntactical or semantic based Question Answering system.

2.1.2.1 Open & Closed Domain QA System

The first Question Answering System was developed in the 1960s and it was basically a natural language interface to an expert system with a specific domain, such as BASEBALL [10]. There are many other systems that researchers have developed in the field of the Question Answering system such as AnswerBus, START, TextMap, EaGLi, AQUA, NSIR and WolframAlpha.

Zheng [100] developed open domain Question Answering system AnswerBus, based on sentence level information retrieval. It actually gives a list of answers, each of which is a hyperlink to the source page. It accepts user’s natural-language questions in English, German, French, Spanish, Italian or Portuguese and extracts possible answers from the Web. It uses a specific dictionary. It classifies all words of retrieved documents in two categories: matching and non-matching according to its predefined formula. It provides web based interface to user.

Katz et al. [54] developed a first web based open domain Question Answering system START (SynTactic Analysis using Reversible Transformations). Questions are asked in English about the MIT AI laboratory, geography and other topics. It has been in operation since
1993. It uses semantic parsing. It adopts a triplet data value “object-property-value”. It handles all varieties of media, including text, diagrams, images, audio and video clips, data sets, web pages, etc. The major problem in this system is that it accepts only simple questions related to its domain like Geography, Science & Reference, Arts & Entertainment, History and Culture as shown in figure 2.2 [73]. It uses the concept of template <subject relation object>. The START system gives a proper answer when the query is asked and the answer appears in frequently asked database. In other cases, the user has to navigate to web page to get an exact answer.

The open domain intelligent Question Answering assistant TEXTMAP [38] focused on developing an algorithm that automatically mines vast amounts of data in order to answer the question posed in natural language. It uses answering techniques like factoid question, cause question and event question. It supports English, Spanish and German language. It provides web based interface to user.
Another open domain Question Answering system, *EAGLi* [37] (Engine for Answers in Genomics Literature) retrieves relevant answers from selected taxonomy. It uses browser and predictive model and also includes advanced search. The EAGLi interface is shown in figure 2.3. It supports only biomedical questions. It provides web based interface to user.

Dragomir et al. [24] developed a web based question answering system - *NSIR Question Answering System*, which allows users to choose from a list of search engines such as Yahoo, All the Web and Excite. User can also specify the number of documents to be processed and the number of answers to be returned as shown in figure 2.4.
The computational knowledge engine - WolframAlpha [41] is an online service that answers factual queries directly by computing the answer from an external source rather than providing a list of documents or web pages. The user interface of WolframAlpha is shown in figure 2.5. It is composed of a toolkit such as mathematica, computer algebra, symbolic, numerical computation, visualization and statistical capabilities. Its limitation is that it only deals with facts and not with options. It limits a computation time for each query.
2.1.2.2 Statistical or Semantic QA System

There are many similarities between open-domain system and semantic question answering system. Both need to find synonyms and also their morphological variants for the keywords. Despite, there are two major differences between these two systems [53]:

a) Open domain question answering classifies the queries based on hierarchies or heuristics to recognize named entities, whereas the semantic information needs the ontology.

b) Semantic systems can classify the query based on the answer and equivalent semantic representation of question rather than only on the type of question submitted. Thus, there is no need of complex hierarchies of answers.

However, almost all the work in this research area has been done using a process of applying semantic and syntactic analysis to get a logical representation of a sentence followed by an appropriate conversion into corresponding representation to a database query. The following are the major systems carried out by the researchers:

P. Haase and P. Heizmann [79] developed a system - ORAKEL which translates wh-queries into F-(frame) logic and evaluates them with respect to the given knowledge base. It uses compositional semantic construction which handles questions involving quantification, conjunction and negation.

Roldolfo A. and Pazos Rangeli [84] discussed various assumptions that database satisfy and suggested the generation of domain dictionary from synonym dictionary and database metadata. The author implemented the system using North wind and pubs database of Microsoft SQL server 7.0. The author gives the various limitations in the form of assumptions related to entering question and converting natural language question into a structured query form.
Venessa Lopez and Motta, E. [57] developed the multi-ontology based question answering system called *PowerAqua*. Inputs are in natural language queries leading to answers using resources of open domain question answering on the Semantic Web. The system uses learning component which ensures that the performance of the system improves over time. It provides web interface to user [43].

Avinash Agarwal [12] discussed semantic analysis of natural language queries using domain ontology. For experimental purposes, railway inquiry is entertained. A toolkit with Python is used for preprocessing. The author also discussed types of questions and types of answers.

Nguyem Tuan Dang and Do Thi Thanh Tuyen [75] developed a system for free eBook library Gutenberg called *eLSSNL* (eLibrary Searching System by Natural Language). The author proposed a method to build a specific Question Answering system which is integrated with the search system for eBooks in the library. Limitation of the system is, it uses predefined syntax structure to input the query in natural language.

Dekang Lin and Patrick Pantel [22] discussed the inference rule for processing the question in natural language. Unsupervised algorithm is proposed for discovering inference rules from text. For parse corpus, the dependency tree concept is used.

Rajendra Akerkar and Manish Joshi [82] in their paper discussed the natural language interface which accepts questions in natural language and generates textual responses. It uses a keyword matching approach. It presents the rules to tackle the phenomenon using shallow parsing technique. The experimental result shows that approach used provides high accuracy and produce reasonable textual responses.
2.1.3 Dialogue System

It consists of a dialog between human and computers in natural language. The following are the list of systems developed by researchers in field of dialog systems namely ELIZA, HAPPY ASSISTANT, BIRD QUEST, CHAT 80, PLANES, etc.

Weizenbaum [97] developed a first dialog system – ELIZA. Eliza’s task was to talk to a human in his or her own language and appear to understand and give meaningful and appropriate replies as shown in figure 2.6. ELIZA script was written in English and German. ELIZA is not restricted for the recognition of a particular set patterns or responses. It provides web interface to user.

Joyce Chai [52] developed a dialogue system - HAPPY ASSISTANT. It helps user to find relevant information about products and services for e-commerce applications. It uses the XML concept to manage domain lexicon and knowledge based business rules.
Another system which combines information extraction and dialogue system called, *BIRDQUEST* [42]. It is a website developed for users who watch programs on TV and ask questions related to Nordic birds. A limitation of this system is that it does not allow free text search. It gives information which can be extracted from its own database only.

David Warren and Fernando Pereira [20] developed a dialogue system - *CHAT-80*. It provides a fascinating sample application, comprising a natural language interface to a geographical database. The user interface of CHAT-80 is shown in figure 2.7 [20]. It gives information about facts like oceans, major seas, major rivers and major cities. The system uses semantic grammar techniques and it is implemented in Prolog language. Major problem in this system is that it can be used in case of specific database only.

![Figure 2.7: Screen shot of CHAT-80 Dialogue System](image)

D.L. Waltz [23] developed *PLANES* (Programmed Language-based Enquiry System) at the University of Illinois Coordinated Science Laboratory. PLANES include an English language front end with the ability to understand and explicitly answer user requests. It carries
out clarifying dialogues with the user as well as it answers vague or poorly defined questions. The system was developed using database related to information of the U.S. Navy 3-M (Maintenance and Material Management), which is a database of aircraft maintenance and flight data. The idea can be directly applied to other non-hierarchical record-based databases.

The next section discusses the survey of NLIDB system, as a special case of NLP application.

## 2.2 SURVEY ON NATURAL LANGUAGE INTERFACE TO DATABASE (NLIDB)

NLIDB systems provide means to access the database independent of its structure knowledge. There are many systems developed of which some are commercial system while others are free systems developed by researchers.

### 2.2.1 Commercially Available NLIDB Systems

Some of the commercially available Natural languages Interface to Database System are EasyAsk, English Query, English Language Front End, etc. The architecture and design of these systems are not in the public realm.

Dix J. [26] developed a system called “The English Wizard (EW) or EasyAsk system” which has its own parser and spell check capabilities. It uses WordNet. The software operates in site search queries and in Original Equipment Manufacturing Embedded systems. A query can be expressed in English, French or any other spoken language system. It does semantic parsing, based on elements at run time. The system considers the data to automatically construct a contextual dictionary which in turn is used to identify words that correspond to the values or catalog attributes and it generates an SQL
statement. It incorporates approximate word matching, stemming and synonyms. The system was tested by taking Northwind database samples using SQL Server 7.0. The major problem encountered in this system was that it does not preprocess the schema.

Microsoft Corporation [66] developed system *The English Query (EQ) system* as a part of SQL server 7.0. It works for any domain. It is best for normalized database. It uses relationship based on verbs. It consists of runtime engine and authorizing tools. It includes database objects, semantic objects and relationships between them. The major problem of this system is that it is available for Win32 platform only. The system supports only a few relationships and it supports the OLE database services like Oracle and Microsoft Access.

Alkama S. [09] developed *The English Language Front End (ELF) system* in Microsoft Access and VB. It uses pre-processed knowledge and context free grammar rather than a programming language. The parser builds the parse tree from the input, by swapping and substituting the words with the SQL keywords. The basic model is to build automatically, extracting most of the relationships of the database. The major problem of this system is it does not have its own documentary method [39].

Mark Watson [61] developed *the NLBEAN system*. The system is based on objects which include input in the form of natural language and it uses a database. The system was tested in sunJDK 1.3. The system parses the input syntactically and tries to match the words to database terms. It uses predefined SQL templates to build an actual SQL query to the database. It has limited semantic and syntactic parsing capability. The user interface of NLBEAN system is shown in Figure 2.8 [73].
Antonija Mitrovi and Martin as cited in [7] [65] developed system SQL-TUTOR. The system tutors students by taking the number of database questions from different databases such as movie databases. It also records the successful and unsuccessful query. Three levels of information are identified in the system namely, database independent information, database structure and database semantics. The information assists the user for generation of natural language query to access the database. Figure 2.9 shows screen shot of SQL-TUTOR [7].
2.2.2 Literature Survey on NLIDB Systems

A lot of work has been done in the area of NLIDB. Asking questions in natural language is very convenient and easy method of data access, especially for casual users who do not understand complicated database query language such as SQL.

Hendrix et al. [30] designed a natural language interface to database system \textit{LIFFER/LADDER}, which gives information about US Navy ships. This system uses a semantic grammar to parse questions and uses distributed database. The system consists of three major components: (a) INLAND (Informal Natural Language Access to Navy Data), (b) IDA (Intelligent Data Access) and (c) FAM (File Access Manager). It has various limitations such as: it supports either one-table queries or multiple table queries with easy join conditions. It can accept only considered set of questions. It does not make use of context free grammar. It provides no general mechanism for dealing with syntactic ambiguity. Only limited Yes/No questions are covered.
It works in limited context problem. Proper nouns are not properly identified. Elliptical inputs are not covered.

Woods W. A. [99] developed a system LUNAR which answers about rock samples brought back from the moon. The system makes use of two databases such as chemical analysis and literature reference. The program used is an Augmented Transition Network (ATN) parsers and procedural semantics. It consists of three components: (i) general purpose grammar (ii) Parser for a large subset of natural English (iii) a rule driven semantic interpretation of component. The first component is responsible for transforming natural language input into the disposable program to carry out its intent and third component deals with executing programs against a database to determine answers to queries. It does not handle ungrammatical sentences and is not flexible. Semantic ambiguity is not fully considered.

G. Bhalotia et al. [32] developed BANKS (Browsing And Keyword Searching) system. The user interface is shown in figure 2.10. The system enables data and schema browsing together with keyword-based search for relational databases. BANKS enables a user to get information by typing a few keywords which also include some templates, following hyperlinks, or by interacting with controls on the displayed results. It is developed in Java using servlets and JDBC, and can be run on any schema without any programming. The problem encounter is the use of limited metadata keywords and non-metadata keywords which could not match with larger database.
Popescu A., M., Etzioni, O., Kautz, H., A [78] developed system PRECISE that uses both an inverted index and a directed schema graph to generate a new database and a personalized natural language synthesis of result. Besides, keyword question answering is implemented using huge inverted indexes and the symbol tables that need to be rebuilt whenever the database is updated. The approach used by the precise system is not suitable for very large databases. It only retrieves single value answer.

Mohsin Ahmed and Milind Gandhe [68] discussed the implementation of natural query processor based on Pattern Matching in prolog language. The author discussed some of the examples which showed the major problems found during implementation and anaphoric disambiguation. The developed system cannot detect semantic contradictions and is suitable for only simple applications.

Chauhari S. et al [17] developed a system DBXplorer which describes a multi-step system to answer keyword queries using relational databases. It proposes methodology which uses a symbol table to
store tables, columns, and rows of all data values that are looked up during the search to identify the locations that contain all the keywords appearing in the question. The system does not consider the solutions that include two tuples from same relation. It only considers exact match of keywords with database attributes.

B. J. Grosz and D.E. Appelt [15] developed system TEAM (Transportable Natural language Interface to Database). It has three major components: (a) acquisition component; (b) DIALOGIC language system (c) data access component. The acquisition component is designed to interact with two kinds of users: Database expert and End user. The DIALOGIC language system deals with the translation of English query into two steps: First the DIALOGIC system converts a logical form of query for input sentence and second the logical form is converted to the formal database query using third component. It has limited capability of using verbs. It assumes that the database should be in third normal form. Time oriented information is not well defined.

Vesper and Shamkant Navathe as cited in [95] and [96] designed the Conceptual Query Language in Natural Language (CQL/NL) which allows users to formulate the database queries in natural language. CQL/NL queries are filtered for search predicates derived from conceptual schema constructs. It uses the associated semantics of semantic data models to construct query solution paths. The prototype is developed in the Oracle relational database system. Based on the identified search predicates, CQL/NL uses a set of predefined natural language templates to compose a natural language explanation of the query. The applicability of conceptual query language is limited to single database only.
Motro [67] proposed a system FLEX, which is a relational database interface that adapts to varying levels of "correctness and well-formulation" of queries, and then returns its construction of queries to the user for acceptance, rejection, or abandonment. It uses a connection sub-graph to construct a query path and this sub-graph helps in converting it to SQL which is based on query template and returned an answer to the user.

B. Sujatha et al.[13] discussed the novel architecture of natural language interface to database which uses a pragmatic approach with illustrations. A special language features that increase system usability such as spelling correction, processing of inputs and run-time system performance were also discussed.

S. Kalpan [89] discussed in his literature review paper, the types of architectures used in NLIDB such as pattern matching systems, syntax based systems, semantic grammar systems and intermediate language representation system. The author also discussed the overview of classification of systems based on various architectures.

Georgia Koutrika and Alkis Simitsis [34] discussed graph based approach to query translation problem. It discussed new approach related to translation of SQL to Natural language. The author discussed domain independent graph traversal strategy for exploring query graphs as phrases in natural language. It uses a general SQL parser that reads SQL queries and generates an XML representation of the parse tree.

Seung En Shin and Young Hoon Kim [90] discussed the approach to automatically generating ParaSQL from a natural language using semantic features and also introduced a methodology of a concept and rule-based ParaSQL generation from natural languages. The approach
uses agricultural datasets. The ParaSQL is a query language for parametric data model. It consists of three expressions: relational expression, domain expression and boolean expression. Relational expression is assigned by the SELECT statement. Domain expression restricts the domain of tuples. Boolean expression qualifies or disqualifies the tuples. It uses a concept dictionary.

Xian Li and Weiyi Meng [93] developed EasyQuerier: a natural language query interface where the user needs to provide keyword that of provided to asking questions to search engine. The translated query is mapped to query interface of Web databases of the domain. The author assumed that integrated query interfaces for each domain would be constructed using existing technique such as a WISE – Integrator.

Rashid Ahmad et al. [80] proposed an algorithm that efficiently maps a natural language query entered in Urdu language to convert it into structured query language. The algorithm has been implemented in Visual C#.NET and tested on a database containing student and employee data.

Bogdan Czejdo and Venugopal Reddy [16] proposed the method of implementing an interactive graphical query interface to a relational database management system and also explain how the graphical operator was defined and implemented efficiently.

Jiawei Han, Yandong Cai and Nick Cercone [51] suggested the prototype system that dealt with an attribute-oriented technique. The system, namely DBLEARN is implemented in C with UNIX software packages LEX and YACC. The system takes a learning request as input, applies discovery algorithm with the assistance of conceptual data stored in
the database. Knowledge rules extracted from the database are the output of the learning system.

Chung-Chian Hsu [18] proposed generalized traditional approach by introducing major value threshold. A synthetic data set and a real data set were used for experiments and the results showed that the proposed methods were feasible and could induce more precise rules.

Nikos et al. [72] discussed a tool for transforming simple queries in natural form into their equivalent SQL queries. The tool supports three languages—English, German and Greek. The tool is developed in Web ontology language that translates text of natural language to SPARQL query and exports the results from OWL database. The author discussed the architecture of the system and conversion methodology with major conditions such as table name ought to be provided first, followed by a punctuation mark, table elements are written followed by a comma, and followed question mark and foreign key.

In-Su Kang, Seung Hoon Na [50] presents a light weight approach to natural language interface in which translation knowledge is semi-automatically acquired. The approach is based on information retrieval framework where most translation knowledge is in the form of documents. It uses only one database domain. It makes use of syntactic analyzers for conversion of question nouns to its domain objects.

Sonia Bergamaschi [92] discussed the novel technique for translating keyword queries into SQL based on mukras algorithm. The approach used by author helps in identifying significant improvement in the identification of semantically meaningful SQL queries that describe the intended keyword query semantics. The author had made the
assumption that each keyword cannot have more than one meaning in the same configuration, no two keywords can be mapped into same database.

Marek Lipczak and Evangelos Milos [64] discussed the problems of e-mail databases. The paper gives introduction of grammar used in Natural Language Search Queries and probabilistic methods for recognizing the query structure. The author used two methods: CYK parsing algorithm which recognize accuracy level for the queries that strictly follow grammar rules and Hidden Markov Model for extraordinary queries. For testing purpose, the author uses Enron datasets. The author maintains the simplicity and intuitiveness of NLIDB by addressing a limited and specific area of search in consumer oriented database.

M. Samsonova, A. Pisarev and M.Blagov [62] developed NLIDB in which they uses conceptual schema that is mapped into logical schema. The authors applied the method to query FlyEx database in natural language. FlyEx database contains information on the expression of segmentation genes in Drosophila melanogaster. The author uses semantic and syntactic approach.

Anh Kim Nguyen and Phuong Hong Nguyen [11] in their paper construct a natural language interface to relational databases, which accept fuzzy questions as inputs and generates answers under the form of tables or short answers. By using derivation evaluation mechanism, the author construct a set of translation rules for all possible structures in standard trees of user questions to translate it into SQL query.
H. V. Jagdish et al. [35] developed NALIX system- a generative interactive Natural Language Query Interface to an XML database. The system can accept an English language sentence as query input, which can include aggregation, nesting, and value joins, among other things. The system can be classified as syntax based system. The transformation process has three steps: (a) generating parse tree (b) validating parse tree (c) translating parse tree into a Xquery expression. It reformulates the input query to XQuery expression and translates it by means of mapping grammatical proximity of natural language parsed tokens to the nearness of corresponding elements in the resulted XML. The system makes little attempt to understand natural language itself.

Runvanpura [85] has developed system - SQ-HAL. It is platform independent and has multi-user support. The user interface is shown in figure 2.11. It is written in Perl, which has a powerful string manipulation capability. It uses top down parser methodology. The system has various limitations such as; string values need to be quoted, date value needs hashes (#) around the values, it has limited thesaurus, the user has to manually entered the relationships, there is no direct method of retrieving column name, the system cannot determine synonym for table name and column names, user has to manually enter synonym words, it does not support Microsoft Access.
Androutsopoulos et al. [4] has proposed a system – MASQUE (Modular Answering System for Queries in English). The system is powerful and portable natural language front end for Prolog databases. It answers written English questions referring to certain domain knowledge such as geography and airplane. Each question is transformed to suitable database query using Prolog database. It uses an extra position grammar parser. It transforms each question into a single SQL query; there is no way to set off the knowledge stored in relational databases by prolog rules. Also the author discussed the database interface concept which gives information about the central issues of NLIDB, typical architecture of NLIDB, the facilities, methods and problems faced in NLIDB in his handbook cited in [5].

Michael Minock [59] proposed a system - Step (Schema Tuple Expression Processor) that adopts a phrasal approach; an administrator couples and phrasal patterns of elementary expressions of tuple relational calculus. The user interface is shown in figure 2.12. It uses a geographical database using relational queries to PostgreSQL.
database. The system does not use a domain independent grammar for syntactic analysis, but rather a phrasal lexicon developed specifically over the PostgreSQL database.

Figure 2.12: Screen shot of Step system

Porfirio P. Filipe et al. [77] discussed a Natural Language Interface for Database. It allows user to formulate multimedia queries. Here the questions are first translated into logic language and then to SQL which is processed by database management to return the answer.

Reis, P., Mamede, N., Matias, J [81] presented a system – EDITE. It provides natural language interface to databases, a new dimension to old approach with the expressiveness of graphical interfaces. The system is a multi-lingual (supporting Portuguese, French, English and Spanish) natural language front-end for relational databases. It answers written questions about tourism resources by transforming them into SQL queries. The answer depends on the type of question. It can be a nominal list of resources, text, images or graphics. It accepts only simple questions. The system does not deal with rich metadata. It does not answer elliptical and anaphora resolution.
Neelu Nihalani et al. [70] discussed the concept of natural language interface to database using semantic matching. The system was tested for Northwind Database and compared with MS English Query product. The main functionality is based on semantics and rules. It is composed of two modules: a pre-processor and run time processor. It does not deal with genitive relations of the sentences. For example, the system generates the SQL query for the question get details of an employee, whereas it does not generate the SQL query for the question get employee details. It does not give answers to elliptical questions.

Niculae Stratica [73] developed a querying system CINDI (Concordia Virtual Library System). The system uses natural language input and gives structured representation of the answer in the form of structured query language. It uses Link Parser to semantically parse the query. It uses WordNet to build the conceptual knowledge base from the database schema. The system was tested using information contained in the virtual library. There are many limitations given by the author such as values should be in double quotes, table name and attribute name should be specified, template should be specified.

M. Jose Suarez- Cabal [58] proposed a methodology to define the coverage criteria for assessing the data retrieved by SQL query. The query is classified into SQL specific and non-SQL specific. The query structure is used to determine the set of situation c-values and r-values to execute against the test suite. The SQL faults were detected such as null values, joins and aggregation functions.

N.D. Karande and G. A. Patil [71] discussed the natural language statement which was converted into a machine understandable form such as what-type questions. Ambiguity among words was taken care of while processing the natural language. The assumptions which are
given for input the natural language query such as all the names should be in double quotes.

Veera Boonjing and Chang Hsu [94] proposed a metadata search approach to provide practical solutions to the natural language database query problem. Here the metadata grew in a largely linear manner and the search was linguistics-free. A new class of reference dictionary integrated four types of enterprise metadata: enterprise information models, database values, user-words, and a query cases. The interpretation of input could be easily identified with the help of the graphical representation method. It uses branch-and-bound method to identify the optimal interpretation that led to SQL generation. The necessary condition was that the text input contained at least one entry in reference dictionary, and the input was to complete and correct grammar which led to correct single SQL query.

Ong Sing Goh and Arnold Depickere [76] discussed the technique for implementing conversational intelligent agent system on the web. It included various layers such as (i) spelling check, (ii) natural language understanding & reasoning, (iii) Frequently Answer Questions (FAQ), (iv) Chat, (v) metadata index search, (vi) pattern matching & case based reasoning and (vii) supervised learning. The author worked on selected pandemic crisis websites which were used to perform knowledge extraction for Domain-Specific databases on the server.

Rukshan et al. [86] proposed the natural language interface for database, which allowed input in the form of an English query through a convenient interface over the internet. A limited data dictionary was used where all possible words related to particular system would be included.
Faraj et al. [28] developed a natural language interface for database system known as GINLIDB (Generic Interactive Natural Language Interface to Database). The author proposed a design by the use of UML and architecture of GINLIDB. The experiments of the query were tested using VB.NET 2005. The system dealt only limited domain and answered a small set of queries. The limitation of the system depends on the size and content of the system’s knowledge base. When the query is not available as per ATN rule, then the query is rejected and the user has to rephrase the query and enter again into the system. Besides, in the input query, user has to explicitly specify the attribute name. For example, the user asks “display employee location”, but the system does not recognize this query. However, if the user asks display employee address, the system has recognized the query and gives a response by generating SQL query.

Amardeep Kaur [2] presented the design and implementation of natural language (Punjabi) interface to (agriculture) database. The system uses MS Access database. The system accepts input in specific template only. The author considers the limited words. Joining of tables is not considered.

David Nadeau [19] presented work which identifies references of the text using Named Entity Recognition (NER) by extracting proper names. The author discusses the Baseline Information Extraction, a system that learns to recognize the named entity in autonomous manner. The author discussed the word-level features, look-up features, document and corpus features. The author assumes various constraints. It does not recognize the entity which is out of its lexicon.
Zsolt et al. [101] designed system for Hungarian as well as it was adopted for English language. The paper discusses the genitive relations used to express a wide range of different relations between entities. It discusses the transformation methods for syntactical analyzed sentence to SQL statements. The proposed method does not resolve the metaphoric expressions, idiomatic expressions, etc.

F. Siaser [29] had presented method for building a Natural Language Interface to Database system. The system prepares an expert system implemented in prolog language which identifies synonyms words in any language. The author discussed the shallow parsing, synthetic parsing and uses Link Parser. The system was tested in Ms-Access and VB.Net. The limitation of the system is it has limited semantic processing capabilities.

Srikanth Thaneeru [88] presented the concept of natural language interface to TIMEBASE- an intelligent temporal database management system. It detects and analyzes the temporal influence of natural language query which is converted to ETSQL query- a query language of TIMEBASE. It is implemented in Java Servlet and JDBC for database access. Complex questions are not handled by the system.

Himani Jain [36] developed Hindi Language Interface to Database. Hindi Shallow Parser which uses Shakti Standard Format is considered for parsing a sentence. The system was developed in Java with MySQL as backend. For testing of the developed system, employee database is used. The system does not deal with linguistic components. It directly maps user keywords to database entity names. It does not deal with complex queries, range queries, order by queries, group queries, and temporal queries. It does not support DDL statements.
Looking at the limitations of the various NLIDB systems developed by a large number of researchers in this field, we have designed and developed the Natural-English Language Interface to Database (N-ELIDB) system that fulfills this knowledge gap. Some of the silent features of N-ELIDB are: (a) value not necessary to be in quotes, (ii) can enter the sentence in free form, (iii) direct mapping to column name with user words are not necessary, (iv) semantic of words are considered, (v) knowledge reuse is maintained, (vi) lexicon analysis is done in detailed manner, (vii) not only SELECT statement but SQL of DDL and DML are also generated by the system.

### 2.3 SUMMARY

This chapter discusses the study related to NLP applications like information retrieval, question answering system and dialogue systems. It then discusses the survey of NLIDB systems in which the limitations are identified and most of this are fulfilled by N-ELIDB system.