A FRAMEWORK FOR CONVERTING SQL QUERIES FROM NATURAL LANGUAGE

Balapradeep K. N[1] ,Ujwal U J[2], Savitha C K[3], Prajna M R[4]

[1, 2, 3, 4] Department of Computer Science and Engineering, KVGCE, Sullia DK

Corresponding Author
Balapradeep K.N,
deepkatoor@yahoo.com

DOI
10.29121/shodhkosh.v5.i5.2024.2672

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Copyright: © 2024 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License.

With the license CC-BY, authors retain the copyright, allowing anyone to download, reuse, re-print, modify, distribute, and/or copy their contribution. The work must be properly attributed to its author.

ABSTRACT

The database interface, called the "natural language interface", uses natural language to allow users to access data without the need for SQL queries. Using query language to interact with databases is always a sophisticated issue. Due to this complexity, the client must use the clear reports that are included in some pre-executed programming projects when using information that is already present in database points of confinement. But, users can make it possible for each nonprofessional user to ask questions and provide requirements in natural language, which the computer can then process to produce the appropriate data. This study introduces a system developed using the "Natural Language Interface to Database" (NLIDB). This approach has led to the emergence of "experts" who can detect compound sentences in English and use them in lessons. After parsing the input sentences, the natural language is converted to SQL.

Keywords: SQL; NLP; database; ANN; Natural Language Interface for Databases (NLIDB)

1. INTRODUCTION

Due to the strength of the Internet in the present era, a wide range of people can access a massive data source. Internet usage surged as a result of the development of smart phones, which also made it exceedingly simple to use. Nowadays, everyone with a smartphone has direct or indirect access to the Internet. In order to ensure that such large amounts of information can be searched, stored and stored quickly, it needs to be organized. Most data is stored using relational database management systems.

The Relational Database Management System (RDBMS) uses a specific language called SQL (Structured Query Language) to store and retrieve data. A user must therefore learn. We can now communicate with computers using human languages thanks to the scientific field of natural language processing (NLP)[1]. We can develop a completely new area of trans disciplinary research by fusing linguistics, computer science, and NLP.

How to cite this article (APA): K.N, B., U J, U., C K, S., and M R., P. (2024). A Framework for Converting SQL Queries From Natural Language. *ShodhKosh: Journal of Visual and Performing Arts, 5*(5), 1034–1041. doi: 10.29121/shodhkosh.v5.i5.2024.2672

The average person uses natural language to communicate with others, but to communicate with a computer, one must learn a computer language. Whether used by humans or machines, language has its own rules and symbols. Humans use symbols to communicate, and there are rules to ensure that the symbols are used according to a set pattern.

Manual translation is done by a multilingual person who is proficient in both languages. Although it appears Straight forward, computers have trouble performing the same task. Word-by-word translation produces subpar results. Here, the difficulty is in developing computer software that can grasp a statement in Understand the meaning of the input, translate it into the language, and work with it in the same way as a human [2].

The main purpose of machine translation is to use technology that will produce different sentences while keeping the meaning the same. The language that individuals naturally use to communicate with one another. However, relational management uses a special language called SQL (Structured Query Language) to record and store data in a relational database management system. The natural language database interface will help those who want to submit questions in natural language. This connection converts natural expressions into corresponding SQL statements, executes them in the database, and displays the results. It is necessary to normalize the text that has been entered for processing so that it is uniform and prepared for additional processing.

The goal of this project is to create a system that can convert customer queries into SQL. This enables users to quickly obtain the contents they need without having to understand any intricate details or SQL language syntax. Lastly, the system runs a SQL query on the database and outputs results to the user. Due to a lack of system advice when an interpretation error occurs, users frequently become trapped and are unable to recover. We offer a framework for natural language query recommendation to address this issue.

Section 2 describes related work. Section 3 explores planned method. Section 4 provides an experiment detail, and Section 5 provides the conclusion.

2. LITERATURE SURVEY

Several steps were taken in creating NLIDB (Natural Language Interface for Databases), so end users no longer need to know SQL to access the database. The assessment evaluates these efforts based on the design and construction of the NLIDB system, its internal components and composition, and its operational functions.The scope of upcoming NLIDB research is also presented.

The standard NLP system takes any one natural language input, then processes it and converts to another language, called target language, like English, and outputs that target language. A variety of machine translation-based methods exist for translating between different languages. We may need to alter the output because it might not produce adequately accurate results.

Challenges and approaches to querying using natural language is proposed by Quamar et. al[3]. It explores a well-liked NL interface technique. The procedure is a structured conversion methodology based on a parse tree. A parse tree is created from the source text and then converted into a parse tree in the destination language. In Duan et al. [4] receive input that contains at least one sentence, they employ a parsing table to go on to the next stage. The parser can do a reduce operation or a shift action. The first one (shift action) is used to insert an additional item into the intermediate data structure from the input text.A new parse node is then built with a lexical feature attached to it. Using this feature as a model, move the objects extracted by the morphological analyzer. Newly created nodes are transferred to this data center. Moreover, the feature structure related to a grammar rule is changed in accordance with the decrease action. If this technique is successful, a new node will be created with a new feature structure. The accept action is carried out after a successful execution. Another approach suggested by Usta et al. [5] involves a hybrid explainable translation pipeline that gives the user both textual and visual explanations of the choices made along the route. Quantitative assessment of xDBTagger has done in three real- world relational databases. The evaluation findings show that methodology is completely interpretable, accurate, and up to 10000 times more efficient at translating queries than competing pipeline-based systems that are at the cutting edge of technology. A structured intermediate representation of the provided text is created in intermediate stages. These interpretations are processed to produce hypotheses about intermediate structures. The hypothesis for it can be scored using either of two models. The first model to give intermediate structure a proper score is the language model. The second model that assigns a suitable score to a source translation event is the translation model. Combining both of these ratings for each intermediate structure hypothesis yields an aggregate score. The target structure hypotheses that scored the highest are chosen. Leibniz, Descartes, and other philosophers attempted to demonstrate a connection between words from various languages in the seventeenth century. There were no actual applications for these theoretical ideas.

Georges Artsrouni later in the mid-1930s submitted a patent for a bilingual dictionary. Peter Troyanskii, a Russian philosopher, also developed a method for treating connections between language grammars and a bilingual dictionary. Alan Turing proposed the "Computing Machines and Intelligence" book and the "Turing Test," a popular intelligence assessment, in 1950. Many natural language processing methods have been developed in the past, the most famous of which is the "Georgetown Experiment" conducted by Georgetown University and IBM in 1950.[6]

Another well-known chatbot with the moniker Racter that could write English prose was created by William Chamberlain and Thomas Etter. The Augmented Transition Network[7] served as the foundation for the operation. The system was made public is explored in Sujatha et al [8]. Actually, it was a machine that answered questions. We can inquire about the moon rock samples with this gadget. The dataset that was utilised to create the system is where it earned its name. Two separate databases were employed for the practical implementation; the first is to use information about writing, and the second is to use analysis. LUNAR's general architecture was divided into three sections. The Syntax

Analysis was used to determine the sentence's grammatical structure. To convert a sentence into a question, a reference word is used to determine its meaning. When the query is sent for processing, the data is retrieved from the source [9]. There are grammatical errors that are not corrected [10].It performed just as well as the previous system, with the exception that it could not handle incorrect questions and was not flexible.

Uma M., et al.[11] created a system whose main objective is to convert NL queries into DB language queries. This system uses the Word Pair Mining Method to remove spelling mistakes from user queries as an added functionality. Then create the SQL query equivalent to the English query. The morphological, syntactic, and semantic levels of the system are all involved. The user will ask a question, which is then converted into text using Android's scripting language (SL4A). The parser is then used to analyze the natural query.

The acronym Natural Language Processing (NLP) can be translated into SQL using the ideas given by Kombade et al. [12]. In general, it is not possible for people who do not know SQL to retrieve information from the database.SQL is a domain-specific language used for storing, altering, and retrieving data saved in relational databases. To solve this issue, we suggested converting the NLP model to SQL. But, under this system, users can also utilize abbreviations to retrieve data from the database. SQL language experts can access the database and retrieve the information they need, but non-technical users cannot. Due to the prevalence of databases, this technique is applicable everywhere.

B.J. Grosz et al. [13] developed the TEAM architecture. There are two modes of operation for the structure. There were two modes: one for database administrators and the other for end users. The first option is the purchase mode, where the database administrator provides some information and makes some settings. These settings will be made when new files are connected to the system. Once connected, the system can be enhanced and expanded with additional information. File names, field names, content or words, and other information about the attached file are displayed in a simple menu.

The Natural Language Database Interface (NLIDB) array is one of its more complex components. The research community is currently addressing and supporting temporal concerns at both the popular annotated dataset (such as Spider) and the most recent sophisticated model levels, as this work explores. Vo et al[14] present a brand-new dataset and a collection of supporting databases for NLIDB's temporal queries.

Rohit Agrawal et al.[15] created an intelligent system whose operations are based on semantic maps. Semantic graphs are always generated by the system. The machine's ingenious engine allows it to "manage pragmatic actions and control conversations." This form of management is necessary to address ambiguity issues that may commonly arise in queries pertaining to several database tables.

For the purpose of creating SQLs from NLS, Yaghmazadeh N. et al[16] suggested the type-based, database content-driven system known as SQLizer. This framework created a system of rules for its own grammar, a relational algebra version. Amandeep Kaur et al.[17] created a Punjabi language interface. It was connected to a database for agriculture. The effort to create an interface for Punjabi is commendable; however it was specifically designed for an agriculture database and did not follow a domain-neutral design. Quamar et al[18] concentrate on leveraging conversational interfaces for data analytics, giving non-technical users and line-of-business owners rapid insights into the data.

In Khurana et al[19], the state of the art in natural language querying was provided. The techniques used and explained in this survey cover a wide range of issues from the communities of databases and natural language processing.

3. PROPOSED FRAMEWORK

Which will method a question posed in natural language through turning it right into a sq. query, this effort will convert natural language to question language (square) in preference to converting one natural language to another. The paintings is extra laborious since sq. queries use English phrases in a rigid, exact syntax. The English language inquiry have to be transformed into a sentence making use of English language phrases to be able to be in a sq. format string. The techniques proven in parent 1 may be employed by using the gadget.. Execution includes various steps. Before and after tokenization, preliminary and intermediate steps to be followed as below.

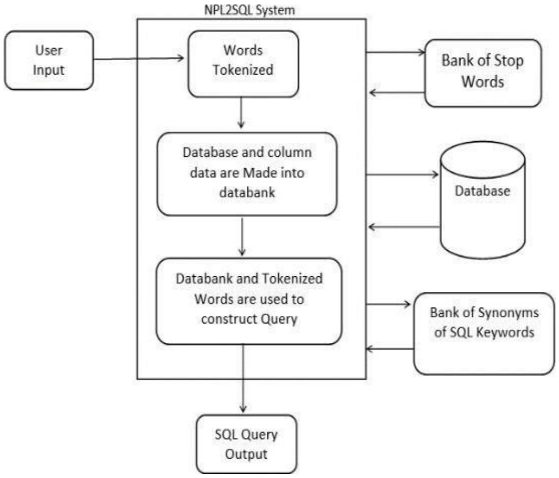

NPL2SQL System
User Input
Words Tokenized
Database and column data are Made into databank
Databank and Tokenized Words are used to construct Query
Bank of Stop Words
Database
Bank of Synonyms of SQL Keywords
SQL Query Output

Fig 1. NPL to SQL system

NORMALIZATION: Normalization of the textual content entered for processing is necessary to make certain consistency and put together it for additional processing. The procedure of normalization adjustments the textual content into a completely unique canonical shape. cleaning, changing, tokenizing, and stemming are all a part of text normalization.

CLEANING: Between-word unique characters are removed. Each phrase has been unmarried-spaced, meaning that extra areas had been eliminated.

SUBSTITUTING: After processing the wiped clean sentence, sure complex phrases are swapped out for his or her less complicated alternatives. moreover, single-word opposite numbers are used to alternative multiword formulations.

TOKENIZATION: Text tokenization is completed with the aid of identifying word delimiters, like areas. it's miles the most broadly used divider for tokenization because most languages make use of spaces among words. The sentence is divided into tokens, which might be discrete phrases.

STEMMING: Sure tokens which can be determined in herbal language texts as plurals are changed to singular through stemming them. The final stem word is confirmed for accuracy in a dictionary after any suffixes to the root natural words are eliminated.

DATA ELEMENT FINDER: An synthetic Neural network (ANN) machine gaining knowledge of technique is used to discover the records elements from the listing of tokens generated at some stage in normalization. For accurate identification of each data detail, which includes entity, characteristic, and condition tokens, the back propagation set of rules is applied. To locate the right series of information pieces in the token set, it is again and again traversed backward and forward.

TRANSLATING: A herbal language to English dictionary is used to convert the tokens into corresponding English tokens. Translation and transliteration are the 2 subprocesses that make up the interpretation procedure. Nouns (such call or city) are transliterated into Roman character, even as phrases from the dictionary inside the local language are translated into their English equivalents.

SQL PREPARATION: With ANN, English language tokens are used to build a sq. question. The column and database names are blended with the stop phrase-filtered enter. in order for ANN to pick out that a phrase within the input relates to a specific column and database, the enter includes both the column names and the database names. An important part of the system is the artificial Neural community's potential to discover the column and database call supplied within the enter.

The neural network technology itself is not a strategy; instead, it serves as a framework that allows numerous machine learning algorithms to cooperate and process large amounts of complex data. Such systems "learn" to execute tasks by taking into account examples, typically without having any task- specific rules written into them. ANN consist of connected nodes with link. Like synapses in the human brain, every link can transmit a signal from one artificial neuron to every other.

Following sign processing, an artificial neuron can talk with its related synthetic neurons. In standard ANN implementations, the sign at a link among synthetic neurons is a actual wide variety, and each synthetic neuron's output is determined by means of a non-linear feature of the sum of its inputs. synthetic neurons are linked by edges. Weights of artificial edges and neurons regularly vary with studying. Weight adjustments the signal power of a connection through making it more potent or weaker. For the signal to be transferred from synthetic neurons, a threshold may additionally want to be crossed. synthetic neurons are typically grouped together into layers. several ranges may additionally revel in exceptional varieties of input changes. perhaps signals journey throughout the layers. In our case we use Deep Neural Network architecture or neural network architecture with more than one hidden layer. Once the input is given to the Neural Network model, it will generate the SQL sequence from the past learning experience.

4. COMPONENTS OF THE SYSTEM

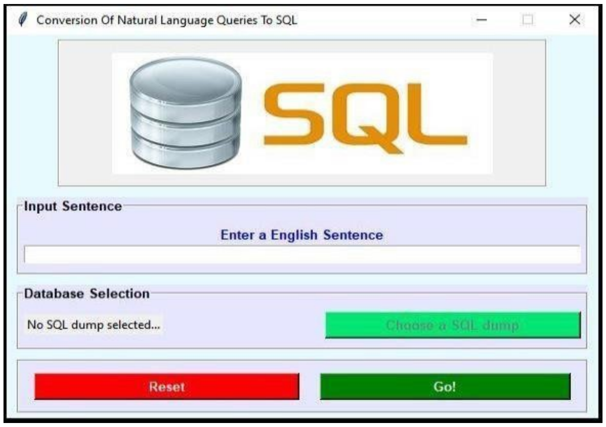
Any given system, software or hardware contains many components of itself for it to work effectively and efficiently. System components are:

GUI: The visible portion of the device that communicates with the person is occasionally known as the Graphical consumer Interface. From the standpoint of the user, that is full-size. to improve the consumer's revel in, our GUI includes buttons, an input textual content discipline, and other peripherals. The input text field in the GUI is for giving the input Query in text. The next button is used to select the database on which operations should be applied. In this we check for the path to the database. After the SQL Query is generated, the query is popped up in a dialog box for the user and also an execute button and don’t execute button is available for the user. Execute button is provided to execute the Query on the database.

Parser: Analyzing a string of symbols in natural language, pc languages, or information systems to ensure they observe the guidelines of a proper grammar is called parsing, syntax evaluation, or syntactic evaluation. Formatted output is produced the use of templating, which is complementary to parsing. moreover, a parser can also confirm that each one required enter has been furnished. Our device makes use of a parser to retrieve characteristics and different data, on the whole from the database. additionally, it is used to run the square query at the user-distinct database.by way of parsing the training information from the database to construct a schooling set, the parser is also extensively used in artificial neural community schooling.

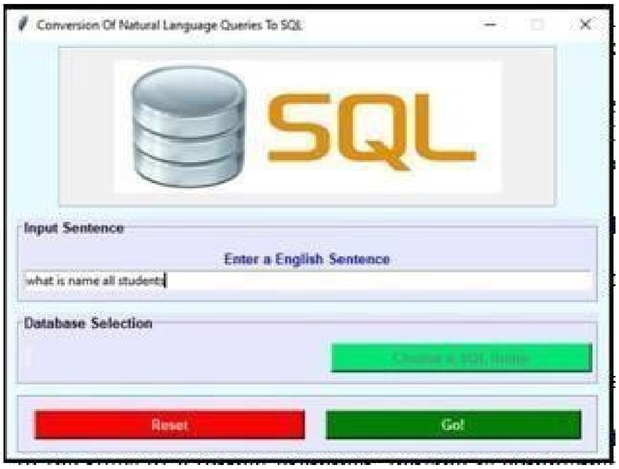
ATTRIBUTE GENERATION: This component examines all the tables stored and finds out all the attributes such as table name and columns name of all the database. After that, it locates each belongings that became included within the user- furnished natural language question. This is essential for the next component.

Input generation: In this component, the database names mentioned in the Input by the user is examined and the columns of those table are retrieved. With a view to build the enter a good way to be fed into the artificial Neural community so that it can generate the sq. query, the key traits of the person's enter are extracted and matched with the information about the column and databases which are available.



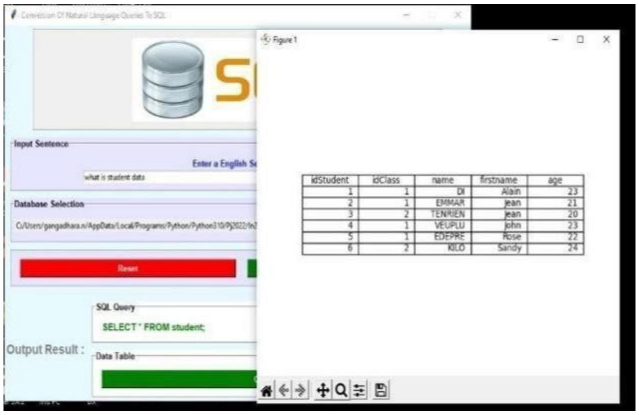
No SQL dump selected...

SQL Query Generation: The Neural Network is fed with the Attribute values of the user input and database input. This generates the SQL Query from the pre-trained Neural Network



what is name all students

model. This is then popped out on the GUI for the user to see.Fig 2.Text entered in Natural language



what is student data

C:/Users/gangadhara.n/AppData/Local/Programs/Python/Python310/Pj2022/In2

SELECT * FROM student;

idStudent	idClass	name	firstname	age
1	1	DI	Alain	23
2	1	EMMAR	Jean	21
3	2	TENRIEN	Jean	20
4	1	VEUPLU	John	23
5	1	EDEPRE	Rose	22
6	2	KILO	Sandy	24

Fig 3.Predicted result page

5. EXPERIMENTAL RESULTS AND ANALYSIS

Natural language enter inquiries from many domains are processed. The produced outputs are recorded in a desk, in which they may be tallied and assessed.

A. DATASET AND TRAINING NEURAL NETWORK

Attribute names appear as columns in the data collection,and record values appear as rows. The object's height and weight for each set member may be used as values for each variable. It is said that every value is a fact. The total range ofrows inside the data series indicates the capacity number of individuals' facts. The dataset which we have used to train theNeural Network is called WikiSQL which is readily availableon Github. This contains around 82000 separate questions andrespective query in the form of Json.

The machine's architecture consisted of two wonderful components: the database and linguistic additives. The linguistic thing is in rate of searching through the input phrases for predicates and constraints. This essential records is generated in order that other machine components can use it. After receiving this statistics, the database thing makes use ofa translation technique to transform query words into sq.tokens. The database component makes use of the templates to create the proper sq. query from the enter words. building a valid and well-formed sq. question does now not continually require that the data observed is suitable.

PERFORMANCE ANALYSIS

A. DISCUSSIONS

Three different databases had been utilized to assess thegadget. those databases are part of exceptional domain names. For a selected area, college students have been advised to jot down records queries in simple language. text files included the ones inquiries. For every of the three textual content files,the applicable square statements were created, and that they had been stored in three awesome text files.

Three text files containing the queries have been applied for checking out. The device-generated square statements and the manually-organized sq. statements have been contrasted. Out of the 448 English inquiries on this test, 425 of the output queries were accurate.

Additionally, the machine's take into account and precision areevaluated. To get records for the person, it can hyperlink two orextra tables. three awesome area databases were used to evaluate the interface, and the outcomes confirmed an accuracyof 94.6%.

B. CONCLUSION

The interface model with design and development details for retrieving information from relational databases through herbal language queries was made to be had by means of this attempt. to satisfy the need, the interface version includes numerous modules which might be coupled to one another. If necessary, the interface can mechanically be part of or extra tables to create a sq. declaration from the question that became entered. it's far an interface that can be connected to any database without requiring any coding modifications and is not restrained to any specific database. three one of a kind units of enter queries are used to test the interface towards 3 exceptional domain databases. The evaluation's accuracy rate for creating square statements from queries turned into 94.6%.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Lehmann, Claude, Dennis Gehrig, Stefan Holdener, Carlo Saladin, João Pedro Monteiro, and Kurt Stockinger. "Building Natural Language Interfaces for Databases in Practice." In Proceedings of the 34th International Conference on Scientific and Statistical Database Management, pp. 1-4. 2022.

S. Karimi, A. A. Rasel and M. S. Abdullah, "Natural Language Query and Control Interface for Database Using Afghan Language," 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Biarritz, France, 2022, pp. 1-8, doi: 10.1109/INISTA55318.2022.9894168.

Quamar, Abdul, Vasilis Efthymiou, Chuan Lei, and Fatma Özcan. "Natural Language Interfaces to Data." Foundations and Trends® in Databases 11, no. 4 (2022): 319-414.

Langote Manojkumar S, Kulkarni Sweta, Mansuri Shabnam, Pawar Ankita, Bhoknal Kishor (2014),—Role of NLP in Indian Regional

Languages||, IBMRD's Journal of Management and Research, Vol. 3, No. 2, pp. 123-128.

Usta, Arif, Akifhan Karakayali, and Özgür Ulusoy. "xDBTagger: Explainable Natural Language Interface to Databases Using Keyword Mappings and Schema Graph." arXiv preprint arXiv:2210.03768 (2022).

Garvin, P. L. (2016), "The Georgetown-IBM Experiment of 1954: An Evaluation in Retrospect", William Mandeville Austin (ed.): Papers in linguistics in honor of Leon Dostert, Berlin, Boston: De Gruyter Mouton, doi:10.1515/9783111675886-006

Lorpaiboon, Chatipat, Jonathan Weare, and Aaron R. Dinner. "Augmented transition path theory for sequences of events." The Journal of Chemical Physics 157, no. 9 (2022): 094115.

Sujatha B, Viswanadha S.R, Shaziya H (2012), —A Survey of Natural

Language Interface to Database Management System||, International Journal of Science and Advance Technology, Vol. 2, No. 6, pp. 56-61.

Woods, W. A. (1977), —Lunar Rocks in Natural English: Exploration in Natural Language Question Answering||, A. Zampolli (ed.): Linguistic Structure Processing., New York. NY: North-Holland Publishing Co., pp. 521.

Nihalani N., Silakari S., Motwani M. (2011), ||Natural language interface for database: a brief review||, International Journal of Computer Science, Vol. 2, No. 8, pp. 600–608..

Uma, M., V. Sneha, G. Sneha, J. Bhuvana, and B. Bharathi. "Formation of SQL from natural language query using NLP." In 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), pp. 1-5. IEEE, 2019.

Kombade, C., More, M., Pujari, A. and Patil, S., 2020. Natural language processing with some abbreviation to SQL. International journal for research in applied science and engineering technology, 8(5), pp.10461048.

B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira (1987),

—TEAM: An Experiment in the Design of Transportable NaturalLanguage Interfaces||, Artificial Intelligence, Vol. 32, No. 2, pp.173– 243.

Vo, Ngoc Phuoc An, Octavian Popescu, Irene Manotas, and Vadim Sheinin. "Tackling Temporal Questions in Natural Language Interface to Databases." In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 179-187. 2022.

R. Agrawal, A. Chakkarwar, P. Choudhary, U. A. Jogalekar and D. H. Kulkarni (2014), "DBIQS — An intelligent system for querying and mining databases using NLP", International Conference on Information Systems and Computer Networks (ISCON), Mathura, pp. 39-44.

Yaghmazadeh, Navid, Wang, Yuepeng, Dillig, Isil and Dillig, Thomas. (2017, Feb). "Type and Content-Driven Synthesis of SQL Queries from Natural Language". Computer Science - Databases, Computer Science

- Programming Languages, eid =arXiv:1702.01168. Available:

https://arxiv.org/abs/1702.01168.

Kaur J, Saini JR (2016),—Punjabi Stop Words||, Proceedings of the ACM Symposium on Women in Research 2016 - WIR '16, pp. 32-37, doi:10.1145/2909067.2909073

Quamar, Abdul, Vasilis Efthymiou, Chuan Lei, and Fatma Özcan. "Natural Language Interfaces to Data." *Foundations and Trends® in Databases* 11, no. 4 (2022): 319-414.

Khurana, Diksha, Aditya Koli, Kiran Khatter, and Sukhdev Singh. "Natural language processing: State of the art, current trends and challenges." Multimedia tools and applications 82, no. 3 (2023): 37133744.