Hindi Language Interface to Database

Rachana Dubey[1, *], *Tejal Kawale*[2, **], *Twisha Choudhary,*[3, ***] and *Dr. Vaibhav Narawade*[4, ****]

[1]Ramrao Adik Institute of Technology, Computer Department, Navi Mumbai, India
[2]Ramrao Adik Institute of Technology, Computer Department, Navi Mumbai, India
[3]Ramrao Adik Institute of Technology, Computer Department, Navi Mumbai, India
[4]Ramrao Adik Institute of Technology, Computer Department, Navi Mumbai, India

Abstract. In our everyday lives we require information to accomplish daily tasks. Database is one of the most important sources of information. Database systems have been widely used in data storage and retrieval. However, to extract information from databases, we need to have some knowledge of database languages like SQL. But SQL has predefined structures and format, so it is hard for the non-expert users to formulate the desired query. To override this complexity, we have turned to natural language to retrieve information from database, which can be an ideal channel between a non-technical user and the application. But the application cannot understand natural language so an interface is required. This interface is capable of converting the user's natural language query to an equivalent database language query. In this paper, we address the system architecture for translating a Hindi sentence in the form of an audio to an equivalent SQL query. The users don't need to learn any formal query language; hence it's easy to use for common people.

1. Introduction

Data is any form of information that is present in the computer memory. Database is one of the major sources of data. It is an organized collection of various types of information. In other words, database is a system that allows storage, easy access, manipulation and updating of data. We require databases for many reasons such as to manage large chunks of data, to secure the data, to maintain the accuracy and integrity of data. Mostly, a database is composed of tables containing rows and columns like a spreadsheet. Some of the famous database platforms to host or create databases are MySQL and SQLite. A database management system is interacting with the users and the database to analyze data. It is a software used to store and retrieve users' data. Users can create their own databases as per their needs using database management system. They can also perform various operations, such as adding, accessing, updating and deleting a record or the entire data. The system serves as a foundation for a database, related operations and makes it much easier to use a database and manage the data.

2. Literature Survey

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 computer 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 (ASR) [3] 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 covert the speech input into its corresponding sequence of words or transcript. The user can ask interact with the database with their voice which

[*]e-mail: rachanadubey.7@gmail.com
[**]e-mail: tejalkawale20@gmail.com
[***]e-mail: tc.twisha@gmail.com
[****]e-mail: vnarawade@gmail.com

undergoes the stages of refinement and then is passes to retrieve the details from the database. Thus, making it easier for novice user to interact with the system without having prior knowledge about the SQL queries.

Using Stanford Dependency Parser, Wibisono developed a domain-independent NLIDB as a tool in processing user input. Filbert Reinaldha and Tricya E. Widagdo [1] presents a solution to two of Wibisono's NLIDB problems, which are inability to process question-type queries and unit conversion. The major stages in processing the previous two mentioned types included Question Identification, Question Handling which again diverges to Wh-question or Yes-No type question, Tag question. Apart from this, the model also included solution for unit conversion.

Mohit Dua, Sandeep Kumar and Zorawar Singh Virk. in [2] proposed an architecture that converts the CRUD operations from text based natural language to SQL query in Hindi. The system accepts the query in the form of natural language in Hindi text, converts it into SQL query wherein the lexicon is used as a dictionary for conversion of Hindi words to equivalent English words. The system support selection, updating and deletion type of queries.

To ease the user's interaction for generation of SQL queries, Prabhdeep Kaur and Shruthi J in [3] proposed Acoustic and Language models to convert speech utterance to query in English language. The Natural Language Processing techniques were applied on English text to generate SQL query. Lexical analyzer, parser and syntax directed translation techniques were used for translation. To make use of such applications in real life, Abhilasha Kate, Satish Kamble, Aishwarya Bodkhe and Mrunal Joshi [4] proposed a system that could be used by Training and Placement cell officers who handle students' details for the placement purposes. The system proposes an architecture wherein the input can be in the form of either speech or text and after processing, it would generate the SQL query. The applications such as ATIS, Geo, Academic, Spider designs models trained on datasets that can work only for the specific database. [5] WikiSQL, a corpus of 87,000 hand-annotated instances of natural language questions, SQL queries, and SQL tables, is the task of mapping a natural language question to form a SQL query given a table from a Wikipedia article. WikiSQL consists of examples of SQL table, question, and corresponding SQL query hand-annotated by crowd workers on Amazon Mechanical Turk [6]. [7] A large-scale complex and cross-domain semantic parsing and text-to-SQL dataset annotated by 11 Yale students is Spider. The goal of the Spider challenge is to develop natural language interfaces to cross-domain databases [8]. It covers 138 different domains that consists of 10,181 questions and 5,693 unique complex SQL queries on 200 databases with multiple tables. The applications and different approaches towards processing of natural language query to SQL query were studied. LADDER, CHAT-80, NaLIX and WASP are few of the interfaces and systems which are working on NLPs [5,9].

3. Problem Statement

With the data growing exponentially day by day, there is a need to access the data in the 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. Thus, we aim at developing an application that would provide a dialogue- based environment for the users to use Hindi language as in voice command to query the required information from the database.

4. Architecture

The user is able to give command Hindi language which would be further processed through speech recognition. The data is extracted from the text and then it undergoes Lexical Analysis, Syntax Analysis and Semantic Analysis. This results in a SQL query which could be further refined (if necessary), the resulting SQL query will be fired on the database and the output will then be displayed to the user.

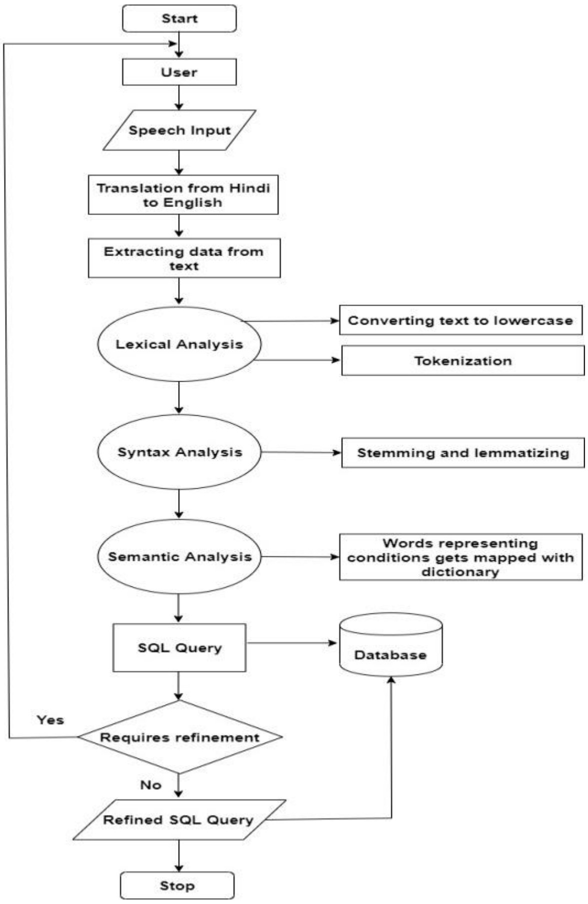


Fig. 1. Architecture of the system

5. Components of HLIDB

5.1. Linguistic Component

The linguistic component of the application handles the speech input by the user in Hindi language. The input could be of declarative type or Wh-type. It would then translate the sentence and form the intermediate SQL query which could be refined (if required) by the user. This component performs morphological analysis and syntax-level analysis to generate the SQL query.

5.2. Database Component

It is responsible to perform traditional database management functions. It executes the query and provides the result of query.

6. Methodology

The proposed method makes use of python language with MySql as a database NLTK (Natural Language Toolkit) library, Speech recognition library, CORE NLP library, PY audio library.

Following are the stages onto which the proposed methodology is based:

i. Natural Language Query

In this step, the user can query the system in Hindi language. The functioning of the system has been explained using an example:

उस विद्यार्थी का नाम और अंक बताइये जिनका अनुक्रमांक १५ है

There could be similar sentences as mentioned above, thus multiple inputs of similar kind would be mapped to the same outputs.

The sentences could be as:

उस विद्यार्थी का नाम और अंक बताइये जिनका अनुक्रमांक १५ है

उस विद्यार्थी का नाम और अंक बताओ जिसका अनुक्रमांक १५ है

उस विद्यार्थी का नाम और अंक दिखाइए जिसका अनुक्रमांक १५ है

उस छात्र का नाम और अंक बताओ जिसका अनुक्रमांक १५ है

ii. Tokenize Query

User provides input to the system in the form of simple sentence in Hindi Language. The system tokenizes that sentence and produce tokens that are to be searched in the system dictionary for mapping. The tokens for the input sentence are as follows:

उस, विद्यार्थी, का, नाम, और, अंक, बताइये, जिनका, अनुक्रमांक, १५, है

iii. Finding English words of Hindi tokens:

Using a system dictionary, the tokens are mapped to their respective English words. It only works on logical word in input sentence. So, we neglect the few words in the query that are present in the system's ignore list. The English words are then generated by the dictionary for input sentence as below:

उस- neglect
विद्यार्थी - Student
का - from
नाम - name
अंक - marks
बताइए - select
जिनका - where
अनुक्रमांक- roll_no
१५ - 15
है – neglect

iv. Map tokens for operations

After finding all the tokens, their corresponding English words are found from the system dictionary which contain the Hindi and English words. All the words of input sentences except proper noun are stored in the dictionary. After finding the English words from the dictionary, input sentence is analysed so that system understands the type of the query entered, i.e., information retrieval, update, insert or delete query.

- If the input sentence contains words like "करो", "कीजिए", "बदलिए", "कर, system interprets it is as "update" and forwards the processed input for the query formulation.
- If the input sentence contains words like "हटाइए", "हटाओ", "रद्द", "निकालये", "निकालो", system interprets it is as "delete" and forwards the processed input for the query formulation.
- If the input sentence contains words like "बताओ", "बताइए", "दिखाइए", "दिखाओ", system interprets it is as "select" and forwards the processed input for the query formulation.

v. Syntactic parsing (Map tokens for table name, columns name, conditions and operators)

The system contains the tables like TABLE-NAME, COLUMN-NAME, CONDITIONS and OPERATORS. TABLE-NAME, COLUMNS-NAME table contains the data required to map table name and column name that are found in Hindi sentence onto their corresponding English words.

vi. Generate SQL Query

After the tokens are mapped, SQL query is generated according to the type of the input Hindi sentence. After that if the input sentence provided by the user contains ambiguity the system will recommend few alternate

https://doi.org/10.1051/itmconf/20203201007

queries from which the user has to choose one so that he gets the correct information he demanded for. After that, the query is executed on the database. The SQL query generated by this function for input sentence.
Query: Select name, marks from student where roll_no = 15.

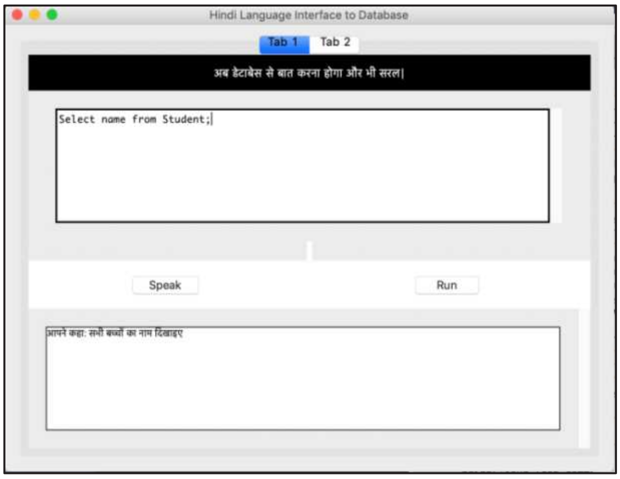
vii.Execute the SQL query:

The SQL query is executed on the database. The retrieved data from the database is displayed to the user as the output.

7. Implementation

This application accepts query in Hindi language which is an audio, parses, translates and tokenizes the sentence using GoogleTrans library and maps the Hindi words with their corresponding English words using maintained dictionary. Currently, the system supports single table named as 'Student' with basic attributes. For better accuracy, POS tagging is also applied to the translated sentence. By analyzing the combination of both, the Hindi input sentence and the translated sentence, it is tested the sentence form is of which type. The type could be of selection, insertion, count, conditional selection or deletion and is understood by a set of predefined words. After this, the table names, column names and conditions are searched in the database table's list. The tokens are then mapped with the database values. After mapping an intermediate SQL query is generated. Finally, a SQL query is generated and executed and the result is presented to the user.

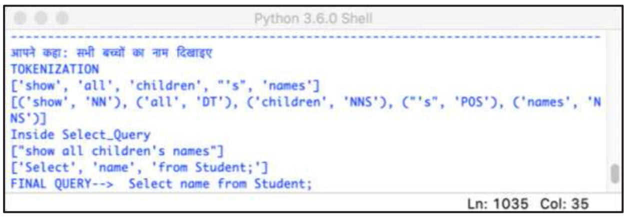
8. Results


Hindi Language Interface to Database
Tab 1
Tab 2
अब डेटाबेस से बात करना होगा और भी सरल।
Speak
Run

```
Select name from Student;
```

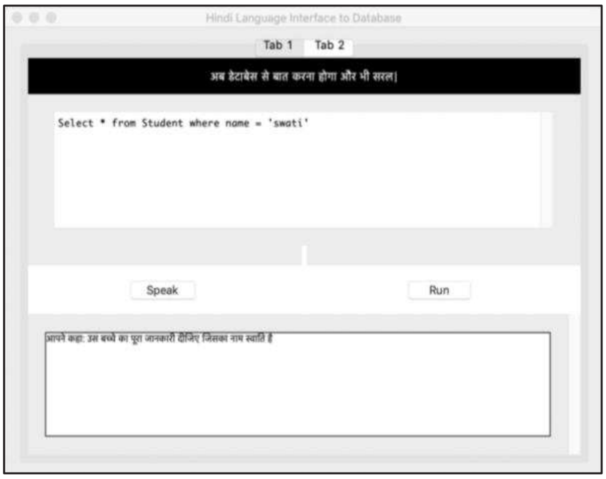
आपने कहा: सभी बच्चों का नाम दिखाइए

Fig. 2. Select record(s) from the database.



```
-----------------------------------------------------------------------------
आपने कहा: सभी बच्चों का नाम दिखाइए
TOKENIZATION
['show', 'all', 'children', "'s", 'names']
[('show', 'NN'), ('all', 'DT'), ('children', 'NNS'), ("'s", 'POS'), ('names', 'N
NS')]
Inside Select_Query
["show all children's names"]
['Select', 'name', 'from Student;']
FINAL QUERY-->  Select name from Student;
```

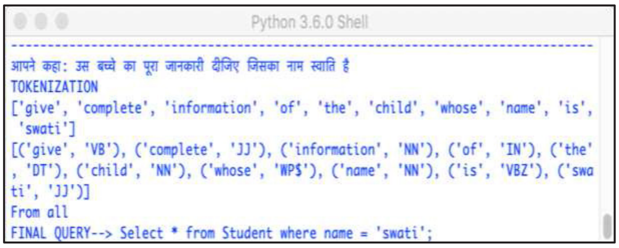
Fig. 3. Pre-processing of data.


Hindi Language Interface to Database
Tab 1
Tab 2
अब डेटाबेस से बात करना होगा और भी सरल।
Speak
Run

```
Select * from Student where name = 'swati'
```

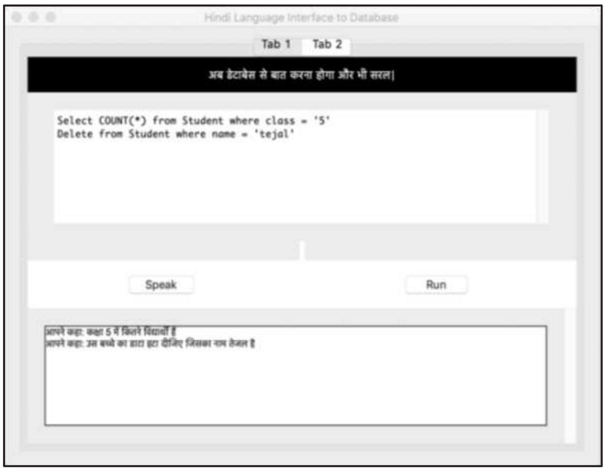
आपने कहा: उस बच्चे का पूरा जानकारी दीजिए जिसका नाम स्वाति है

Fig. 4. Query with 'where' clause.



```
----------------------------------------------------------------------------
आपने कहा: उस बच्चे का पूरा जानकारी दीजिए जिसका नाम स्वाति है
TOKENIZATION
['give', 'complete', 'information', 'of', 'the', 'child', 'whose', 'name', 'is',
 'swati']
[('give', 'VB'), ('complete', 'JJ'), ('information', 'NN'), ('of', 'IN'), ('the'
, 'DT'), ('child', 'NN'), ('whose', 'WP$'), ('name', 'NN'), ('is', 'VBZ'), ('swa
ti', 'JJ')]
From all
FINAL QUERY--> Select * from Student where name = 'swati';
```

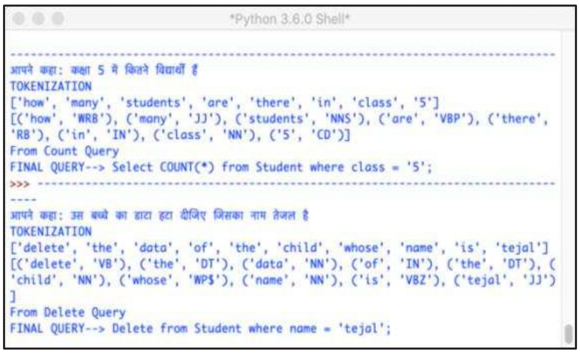
Fig. 5. Query formulation (step by step).


Hindi Language Interface to Database
Tab 1
Tab 2
अब डेटाबेस से बात करना होगा और भी सरल।
Speak
Run

```
Select COUNT(*) from Student where class = '5'
Delete from Student where name = 'tejal'
```

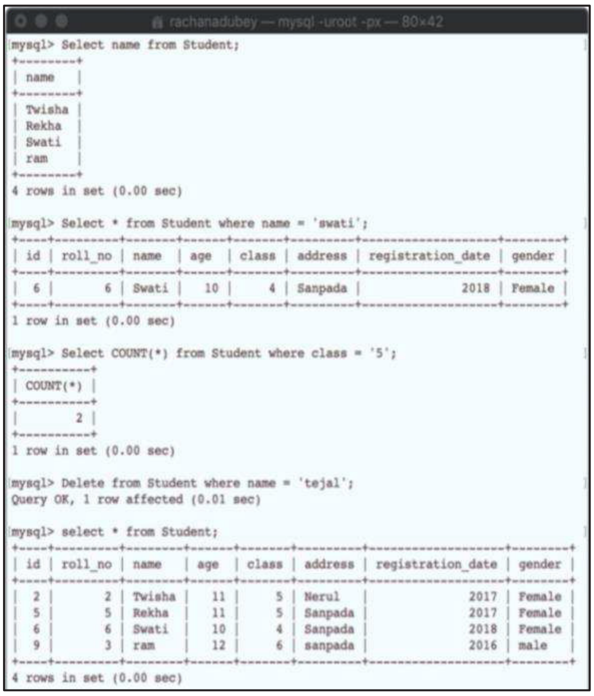
आपने कहा: कक्षा 5 में कितने विद्यार्थी है
आपने कहा: उस बच्चे का डाटा हटा दीजिए जिसका नाम तेजल है

Fig. 6. Count and Delete query



```
---------------------------------------------------------------------------
आपने कहा: कक्षा 5 मे कितने विद्यार्थी है
TOKENIZATION
['how', 'many', 'students', 'are', 'there', 'in', 'class', '5']
[('how', 'WRB'), ('many', 'JJ'), ('students', 'NNS'), ('are', 'VBP'), ('there',
'RB'), ('in', 'IN'), ('class', 'NN'), ('5', 'CD')]
From Count Query
FINAL QUERY--> Select COUNT(*) from Student where class = '5';
>>> -----------------------------------------------------------------------
----
आपने कहा: उस बच्चे का डाटा हटा दीजिए जिसका नाम तेजल है
TOKENIZATION
['delete', 'the', 'data', 'of', 'the', 'child', 'whose', 'name', 'is', 'tejal']
[('delete', 'VB'), ('the', 'DT'), ('data', 'NN'), ('of', 'IN'), ('the', 'DT'), (
'child', 'NN'), ('whose', 'WP$'), ('name', 'NN'), ('is', 'VBZ'), ('tejal', 'JJ')
]
From Delete Query
FINAL QUERY--> Delete from Student where name = 'tejal';
```

Fig. 7. Query formulation (step by step).



```
mysql> Select name from Student;
+--------+
| name   |
+--------+
| Twisha |
| Rekha  |
| Swati  |
| ram    |
+--------+
4 rows in set (0.00 sec)

mysql> Select * from Student where name = 'swati';
+----+---------+-------+------+-------+---------+-------------------+--------+
| id | roll_no | name  | age  | class | address | registration_date | gender |
+----+---------+-------+------+-------+---------+-------------------+--------+
|  6 |       6 | Swati |   10 |     4 | Sanpada |              2018 | Female |
+----+---------+-------+------+-------+---------+-------------------+--------+
1 row in set (0.00 sec)

mysql> Select COUNT(*) from Student where class = '5';
+----------+
| COUNT(*) |
+----------+
|        2 |
+----------+
1 row in set (0.00 sec)

mysql> Delete from Student where name = 'tejal';
Query OK, 1 row affected (0.01 sec)

mysql> select * from Student;
+----+---------+--------+------+-------+---------+-------------------+--------+
| id | roll_no | name   | age  | class | address | registration_date | gender |
+----+---------+--------+------+-------+---------+-------------------+--------+
|  2 |       2 | Twisha |   11 |     5 | Nerul   |              2017 | Female |
|  5 |       5 | Rekha  |   11 |     5 | Sanpada |              2017 | Female |
|  6 |       6 | Swati  |   10 |     4 | Sanpada |              2018 | Female |
|  9 |       3 | ram    |   12 |     6 | sanpada |              2016 | male   |
+----+---------+--------+------+-------+---------+-------------------+--------+
4 rows in set (0.00 sec)
```

Fig. 8. Results of the final queries.

9. Future Work

We further suggest a framework that could support complex queries like GROUP BY and HAVING which require access to more than one table to get the result. The GROUP BY clause is used to gather rows of similar data into groups. The HAVING clause is used to impose restriction(s) on the results based on certain criteria.

10. Conclusion

Through the discussed methodology we present an approach that would use speech recognition, natural language processing and SQL query generation. The user asks for an operation to be executed on relational databases in natural language which is then processed and converted to the corresponding SQL query and is fired on the database based upon the existing table names and attribute names. To make the system user-friendly and simple enough to understand, interactive environment is created through dialogue-based process.

REFERENCES

[1] Filbert Reinaldha and Tricya E. Widagdo. Natural language interfaces to database (nlidb): question handling and unit conversion. International Conference on Data and Software Engineering (ICODSE) (2014).

[2] Mohit Dua, Sandeep Kumar, and Zorawar Singh Virk. Hindi language graphical user interface to database management system. 12th International Conference on Machine Learning and Applications, IEEE (2013).

[3] Prabhdeep Kaur and Shruthi J. Conversion of natural language query to sql. 12th International Conference on Machine Learning and Applications(2016).

[4] Abhilasha Kate, Satish Kamble, Aishwarya Bodkhe, and Mrunal Joshi. Conversion of natural language query to sql. Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE (2018).

[5] Victor Zhong. How to talk to your database. https://blog.einstein.ai/how-to-talk-to-your-database.

[6] Abhijeet R. Sontakke and Prof. Amit Pimpalkar. A review paper on hindi language graphical user interface to relational database using nlp. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), **Volume 3 Issue 10** (2014).

[7] https://yale-lily.github.io/spider

[8] E U Reshma and P C Remya. A review of different approaches in natural languages interfaces to databases. International Conference on Intelligent Systems (ICISS), IEEE (2017).

[9] Tao Yu. Spider: One more step towards natural language interfaces to databases. https://medium.com/@tao.yu/spider-one-more-step-towards-natural-language-interfaces-to-databases-62298dc6df3c (2018).