iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51587

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Converting Natural Language To SQL Queries Using LSTM

Venkata Adithya Vytla[1], Dr Sreedhar Bhukya[2], Dr Vijay Saradhi[3], Pabbathi Saicharan[4], K V Venkata Ramana[5], B.Vasundhara Devi[6]
[1, 4, 5]B.Tech Scholars, [2, 3]Professor, [6]Associate Professor, Dept .of Computer Science and Engineering, SNIST, Hyderabad-501301, India

Abstract: ***In today's world, not everyone is familiar with using Structured Query Language (SQL). This makes it hard for users to understand or create complex SQL queries. What we need is an improved application with a smarter interface that can bridge the gap between novice users and databases. Databases are great at managing data, but to understand their structure, users have to learn SQL. This poses a challenge for non-experts who aren't well-versed in SQL. What they need is a system that allows them to interact with databases in natural language. The system should be capable of understanding and responding to natural language commands. To achieve this objective, we utilize a range of end-to-end deep learning models, as well as probability models like conditional random field. The ambiguity in natural language makes it exceedingly difficult to determine the exact meaning of every word, therefore it is a difficult process to map individual keywords to the description of the schema and the contents of the underlying database. Accurate predictions can help us avoid unnecessary trouble. If we apply machine learning tools, we can skip the complicated process.***
Keywords: Natural language, SQL, Machine Learning.

I. INTRODUCTION

Transforming natural language to SQL is a technology that aims to bridge the gap between human language and computer language. The language used to interact with relational databases is called SQL, or Structured Query Language. Although it is a strong tool for managing, organizing, and retrieving data, using it can be complicated and challenging for non-experts.

The objective of translating natural language to SQL is to develop a system that can comprehend human language and translate it into SQL. With this, accessing and modifying data in a database would be much simpler for non-experts. For instance, a user could enter a natural language query like "Show me all the customers who bought a product in the last month" and the system would translate that query into SQL and retrieve the necessary data without the user having to learn the syntax and structure of SQL.

NLP involves analyzing text using algorithms and statistical models to find patterns and connections between words and phrases.

Data analysis, business intelligence, and information retrieval are just a few of the many potential uses for translating natural language to SQL. This project aims to explore the techniques and tools used in transforming natural language to SQL, and to develop a system that can effectively perform this task.

II. RELATED WORK

The conversion of natural language to SQL has received a lot of attention in related research. Natural Language to SQL (NL2SQL) is the name of this area of study [1], which has been active for a while.

A set of manually generated rules were used in some of the earlier NL2SQL implementations to translate natural language queries to SQL queries [2]. However, the richness and variety of spoken language, along with its inability to scale to big datasets, hampered rule-based techniques.

In more modern NL2SQL methods, the mapping between natural language and SQL has been learned using machine learning techniques like deep learning and neural networks [3]. These methods have demonstrated potential for raising the precision and scalability of NL2SQL systems.

For instance, the Seq2SQL system, created by Microsoft Research and University of Washington academics in 2018 [4], employs a neural network to learn the correspondence between SQL queries and natural language inquiries. On a number of benchmark datasets, the system delivered cutting-edge performance.

Since then, a number of other studies on NL2SQL have been published, including investigations into data augmentation methods, semantic parsing, and transfer learning.

NL2SQL is a growing field of study with a lot of potential for real-world applications in data analytics, business intelligence, and information retrieval, among other fields.

III. METHODLOGY

LSTM (Long-shot-term-memory) is a type of recurrent neural network which can be used in the natural language processing and sentimental analysis and to convert the given natural language text to the SQL query.

The memory cells in the LSTM network are responsible for storing long term information and the gating mechanism which includes the gates like input, output and forget gate which are intern used to control the flow of information in and out of the memory cells and each corresponds to a sigmoid function that outputs the value between 0 and 1.

The forget gate in the LSTM decides which information should be discarded from the memory cell based on the previous output and current input and the input gate decides which new information should be added to the memory based on current input and previous output and the output gate decides which information to output based on the current input and previous output.

Machine translation, sentiment analysis and speech recognition are few of the natural language processing tasks at which LSTM networks has demonstrated good results but they may need a lot of data to work well and they can be computationally expensive to train.

To map the natural language text to SQL queries we are using an LSTM based sequence to sequence model and This model is trained on a dataset of natural language and the corresponding SQL queries. We are using the LSTM to learn the relationships between the words in the natural language text to corresponding SQL queries.

The LSTM is used to encode the natural language query into a fixed length array or vector which is then decoded by another LSTM to produce the SQL query. The decoder LSTM is trained to predict the next word in the SQL query based on the current state and the previous word and this is repeated until the entire SQL query is generated.

IV. IMPLEMENTATION

The proposed model involves multiple steps to convert natural language to SQL query. Firstly, a database is used to train the model, followed by preparing the data by paraphrasing the natural language text and generating corresponding SQL text. The model first converts the input data into vectors of fixed length and the LSTM model is prepared by using encoder and decoder. The encoder assigns a value to the paraphrased natural language text which is then decoded to SQL words by the decoder. Finally, the SQL query is generated and returned by the model. The detailed steps involved are:

Pre-process the data: The data needs to be cleaned and pre-processed to remove any noise, inconsistencies, or irrelevant information. This involves tokenizing the input text, converting it to lower case, removing stop words, and stemming or lemmatizing the words.

Prepare the data: the data which was pre-processed needs to be converted into vectors of fixed length and then it is encoded with numbers and this involves <PAD>, <UNK>, <SOS> and <EOS>. These encoded vectors are used to train the LSTM model.

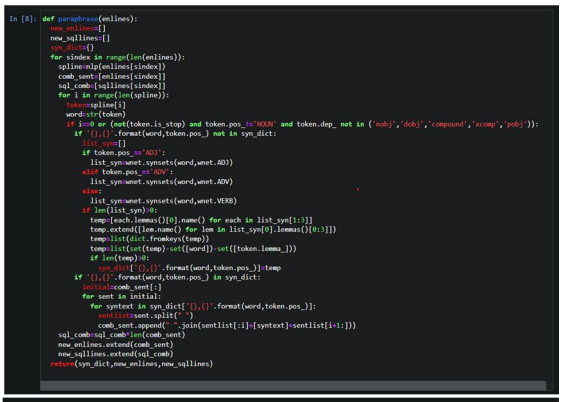
Defining LSTM architecture: This step involves to define the number of layers, the no of nodes in each layer and the activation function etc.

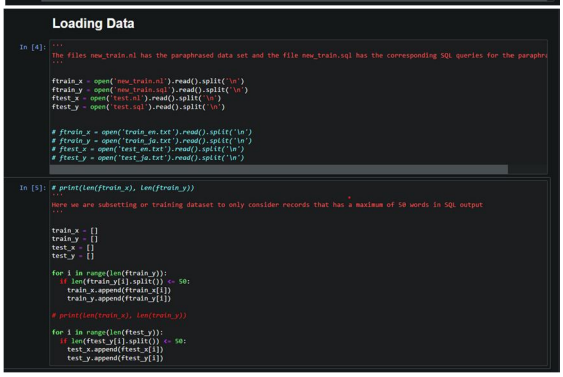
Training the LSTM: The pre-processed data is used to train the LSTM model and during the training the model updates it weights in the neural network to minimize the loss function. The LSTM model may use back propagation and gradient descent to adjust its weights to maximize its output accuracy.

Evaluation of the model: The trained model is evaluated by testing it on the testing data that was separated at the time of processing of data.

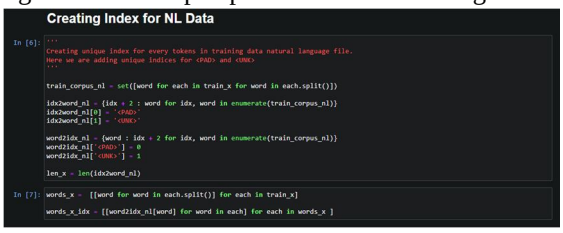
The above code imports the required packages for executing the code.

The below code is used to paraphrase the input natural language files and SQL files. Then, it analyzes the words used and identifies the nouns that can be potentially used as tables or columns and returns the train.nlp and train.sql files that can be later used to train the model.





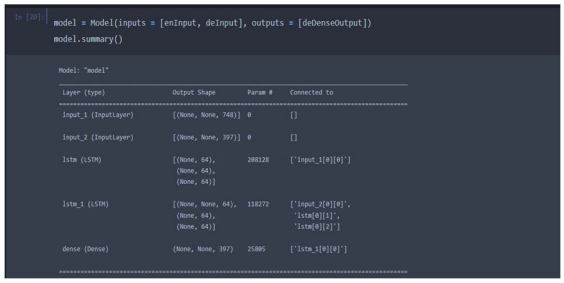
The above code is used to load the training data that was paraphrased. We are loading the train and test data for the model.



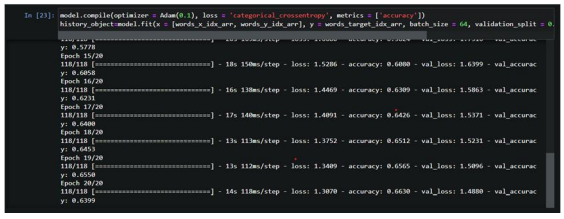
The above code creates a dictionary of all the words available in training data and assigns a unique number to these words. It collects these words from each of the natural language queries and creates another dictionary with key as unique number and value as word.

The below code creates dictionaries for the SQL query words and assigns a unique number to each of these words. It creates another dictionary with keys as unique numbers and words as values.

The above code also creates the list SQL queries with pattern <SOS> list of words in query <EOS> and also a list of the similar list but with the unique value that was indexed to the word.

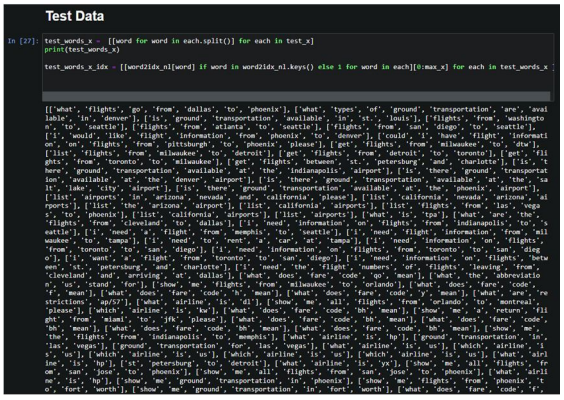


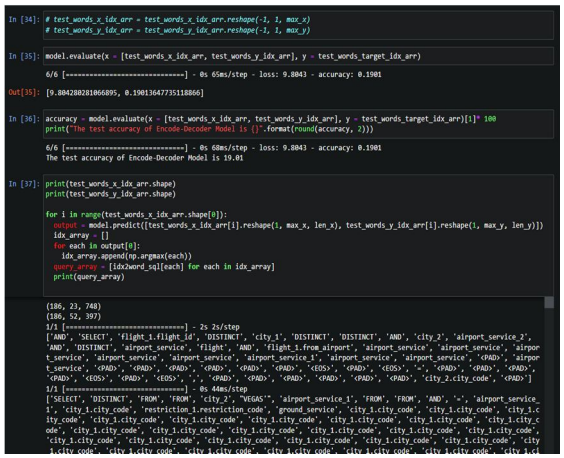
The above code is used to create a model that takes 2 inputs enInput which consists of the encoder model and the deInput consists of the decoder model.



The above code is used to train the model by providing the lists of input data that was generated by giving unique number to the words in the code.

The below picture shows the test data which is going to be indexed with unique number assigned to the words and these lists of unique numbers are given to the model for testing.





The above code consists of the code which is used while testing.

```
which airlines fly from boston to washington dc via other cities
SELECT DISTINCT airline_1.airline_code FROM airline airline_1 , flight flight_1 , airport_service airport_service_1 , city city_1 , airpor
t_service airport_service_2 , city city_2 , flight_stop flight_stop_1 , airport_service airport_service_3 , city city_3 WHERE airline_1.ai
rline_code = flight_1.airline_code AND flight_1.from_airport = airport_service_1.airport_code AND airport_service_1.city_code = city_1.cit
y_code AND city_1.city_name = 'BOSTON' AND ( flight_1.to_airport = airport_service_2.airport_code AND airport_service_2.city_code = city_
2.city_code AND city_2.city_name = 'WASHINGTON' AND city_2.state_code = 'DC' AND flight_1.flight_id = flight_stop_1.flight_id AND flight_s
top_1.stop_airport = airport_service_3.airport_code AND airport_service_3.city_code = city_3.city_code AND 1 = 1 )
```

Encoder-Decoder Output Query

```
SELECT DISTINCT flight_1.flight_id
FROM flight flight_1 ,
airport_service airport_service_1 ,
city city_1 , airport_service
airport_service_2 , city city_2
WHERE flight_1.from_airport =
airport_service_1.airport_code AND
airport_service_1.city_code =
city_1.city_code AND
city_1.city_name = 'BOSTON'
AND flight_1.to_airport =
airport_service_2.airport_code
AND airport_service_2.city_code =
city_2.city_code AND
city_2.city_name = 'PHOENIX'
<EOS>
```

The above picture is the output given for one of the query while testing.

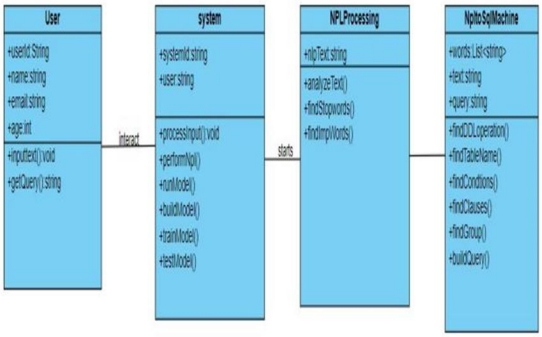
```
Epoch 10/20
118/118 [==============================] - 11s 95ms/step - loss: 0.1621 - accuracy: 0.9521 - val_loss: 0.2709 - val_accuracy: 0.9349
```

The above shows the accuracy of the model that was created.

A. UML Diagrams

1) Class Diagram

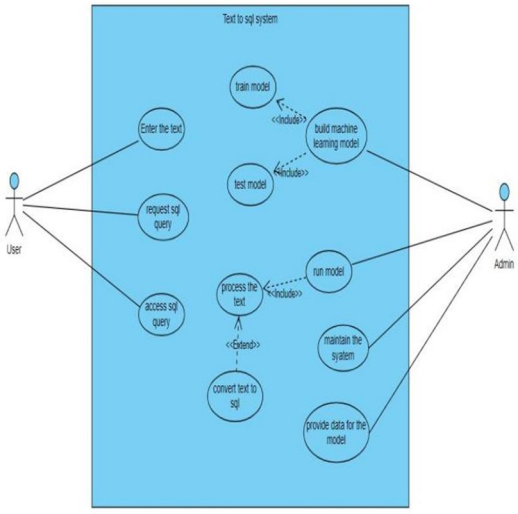
Class diagram is a graphical representation of the classes involved in the system. The class diagrams consist of classes. Each class has attributes and operations and the interactions between classes are represented by using relations. The natural language to SQL class diagram consists of 4 classes i.e user, system, NLP Processing and NLPtoSqlMachine. The NLP processing class analyzes the text and prepares it for the model. The NLP to SQL class does many operations such as finding stop words, group words etc to build the query.



2) Use Case Diagram

A use case diagram is a graphical representation of the system that shows actors involved in the process and the operations that are performed by the actors.

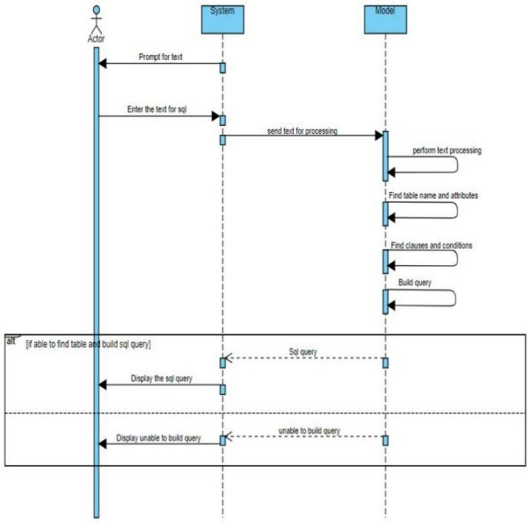
The use case diagram consists of text to SQL sub system. It involves 2 actors i.e admin and user. The user enters the text and requests for the query. The admin trains the system and provides the data required for training and tests the model as well. The admin maintains the system so that it can process the user requests.



B. Sequence Diagram

A sequence diagram is a type of UML diagram which is used to represent the interactions between the objects in a chronological order.

The text to SQL sequence diagram consists of 3 objects i.e user, system and model. The user interacts with the system by entering the text which is sent to the model to process the text. After processing, it is used to build the SQL query based on the requirements that are mentioned by the user in the text. The SQL query is built and returned to the system which is then displayed to the user.



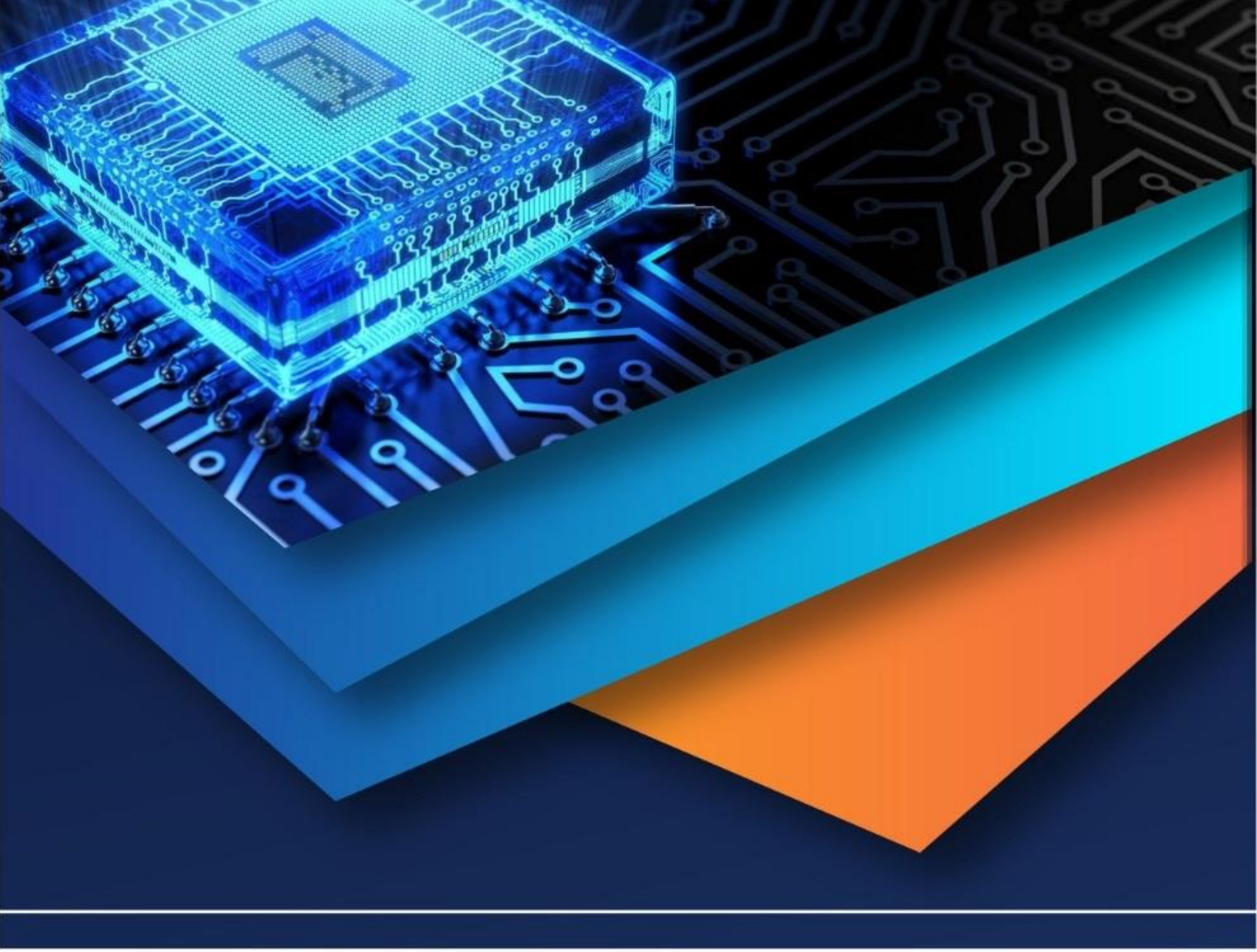
V. CONCLUSION

Transforming natural language to SQL is an exciting field that has the potential to revolutionize the way we interact with data. By bridging the gap between human language and computer language, it has the potential to make it much easier for non-experts to access and manipulate data in a database.

The development of natural language processing (NLP) technology will play a crucial role in this process. As NLP technology continues to evolve, it will become more accurate and efficient, making it easier to transform natural language to SQL. Additionally, the integration of machine learning algorithms and large language models (LLMs) will further enhance the accuracy and efficiency of this process. The potential applications of this technology are vast, including improved data analysis, enhanced decision-making capabilities, and increased productivity. It is clear that this technology has the potential to revolutionize the way we interact with data, and it will be exciting to see how it continues to evolve in the years to come.

REFERENCES

[1] A survey paper on Natural Language to SQL systems: Zhong, V., & Liu, Z. (2019). A survey on natural language processing for databases. ACM Computing Surveys (CSUR), 52(4), 1-34.
[2] Rule-based NL2SQL techniques: Popescu, A. M., & Etzioni, O. (2005, June). Extracting product features and opinions from reviews. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (pp. 339-346).
[3] Modern machine learning-based NL2SQL techniques: Xu, K., Wu, L., Wang, Z., Chen, Y., & Wu, D. (2017). SQLNet: Generating structured queries from natural language without reinforcement learning. arXiv preprint arXiv:1711.04436.
[4] The Seq2SQL system: Zhong, V., & Liu, Z. (2019). Seq2SQL: Generating structured queries from natural language using reinforcement learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 2372-2382).
[5] Other recent studies on NL2SQL: Agarwal, S., Hamborg, F., & Lehmann, J. (2021). Data augmentation for question-to-SQL generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4504-4514).
[6] Kushman, N., Erk, K., & Liang, P. (2014). Compositional semantic parsing on semi-structured tables. Transactions of the Association for Computational Linguistics, 2, 143-154.
[7] Dong, L., Wei, F., Zhou, M., & Xu, K. (2018). Neural semantic parsing with type constraints for semi-structured tables. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1035-1045).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY