Received: January 27, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Syntax and Table Aware Parsing Based Naturalized Structured Query Language

Praveena Mydolalu Veerappa[1]* **Ajeet Annarao Chikkamannur**[2]

[1]*Dr Ambedkar Institute of Technology, Karnataka, India*
[2]*R L Jalappa Institute of Technology, Karnataka, India*
* Corresponding author's Email: mv_praveena@rediffmail.com

Abstract: A database is characterized as accumulation of data that is organized to access, manage, and update information effectively. All information is store in a database and there are numerous approaches to interface with the database to access our information. A client needs some specialized knowledge to extricate information from the database. They have to utilize Structured Query Language (SQL) for information definition, information manipulation, or information control. Although, the vast majority of the clients who need to separate information from a database are not technical experts. Hence, there is a tremendous communication gap between the database and its core client. With the advancement of Natural Language Processing (NLP), a client would now be able to converse with their database in their characteristic language without learning database language. The communication gap between the client and the database has begun to vanish with this stunning capacity. In this paper, the methodology addresses this issue by considering the structure of table and the syntax of SQL. The nature of the generated SQL query is fundamentally enhanced through finding out how to recreate content from column names, cells or SQL keywords; and enhancing the generation of WHERE clause by utilizing the column-cell relation. Analyses are directed on WikiSQL, a recently released dataset with the biggest query SQL sets. Our approach significantly improves the state-of-the-art precision, recall and F-measure from 76.5% to 82%.

Keywords: Database, Structured query language, Natural language processing, WHERE clause, WikiSQL dataset.

1. Introduction

In the present fast computing situation, computer based data recovery innovations are in effective way to encourage scholastic and education institutions associations, organizations to deal with their data frameworks and procedures [1]. These are utilized to manage information that is capable for managing various types of information which are stored in the databases otherwise called Database Management System (DBMS) [2]. Despite, Information recovery of a vast information being effective in social databases, the client still needs to master the DB schema to figure out the inquiries. Artificial Intelligence (AI) and Linguistics can be consolidated to create programs that can help to understand and deliver data in a NL. NL is the dialect that is utilized by every single individual for communication in reality [3, 4]. The term real world makes the issue considerably more troublesome. While the term NL known to be extremely convenient dialect and talked by people, NLP is a piece of AI that deals with frameworks and programs and convey in natural dialect [5]. Frameworks that are reasonable for the processing and understanding NL connect between the man-machine communications boundaries to a great extent. The primary reason for NLP is to empower communication among human and computers without execution of complex Commands and techniques [6].

NLP is the systems that can make the computer to comprehend the common dialects utilized by people [7]. The applications that will be conceivable when computer would have the capacity to process NL translating dialect precisely in real time, or extracting and outlining data from an assortment of information sources relying upon the client's demand. NLP frameworks catch important data from a contribution of words (sentences, passages, pages and so on.) as an organized output. Dialect handling is most basic part of our framework [8]. This framework is preparing the NL which is English entered by the client. With that information our framework anticipated the appropriate consequence of that sentence. Presently, the requirement of business framework is to extracting information from a DBMS, namely, MS Access, Oracle and others [9]. Today, one of the most focused on issues in the field of AI (Computer Science) is to make machine this much fascinating so then it can nearly behave like a person. The behaviors of individuals have been refined during machine usage e.g. presently days' machines can hear with the use of microphone, talk by creating sound, see with the use of cameras, yet at the same time there are a few zones where this machine improvement isn't totally effective and some of them are to comprehend NL, gaining from experience and making autonomous decisions in real time environment etc. [10, 11].

The proposed framework is intended to limit the communication gap between a human and computer. It is produced to encourage enhanced collaboration between the two. As it is known databases can only react to standard questions written in SQL and it is less feasible for a common individual to know SQL. The proposed approach a Syntax-and Table-Aware seMantic Parser (STAMP) encodes the inquiry into persistent vectors, and integrates the SQL question with three channels. The model realizes when to produce a column name, a cell or a SQL key phrase. The technique additionally consolidates column-cell connection to moderate the poorly shaped results. The strategy analyses on WikiSQL to approve the execution of the proposed technique. The paper describes as Section 2 represents the auxiliary question dialect, whereas Section 3 represents the review of ongoing innovations related to the present work. The proposed philosophy methodology can be described in Section 4, additionally, the investigations and validated outcomes are clarified in Section 5. At last, Conclusions are made in Section 6.

2. Structure query language

SQL is a popular query language for relational database management systems (RDBMS) according to ANSI (American National Standards Institute). In the course for beginner, two types of SQL usually introduce to learners, i.e., DDL and DML [12]. Data Definition Language (DDL) statements are used to define the database structures in a relational database model. CREATE, ALTER and DROP are examples of DDL that are used for creating objects, altering structure of objects and deleting objects in a database, respectively. For Data Manipulation Language (DML), the set of statements is used to manage data within schema objects. For example, SELECT is applied for retrieving data from a database. INSERT, UPDATE and DELETE are used for modifying data in a database. Main RDBMSs such as Oracle, Microsoft SQL Server, PostGreSQL and MySQL use SQL as a language for managing their databases. Although these RDBMSs use SQL as their primary language, most of them also have some proprietary extensions that are normally only applied on their system. However, standard SQL commands can be applied on most of RDBMSs [13].

3. Literature review

Numerous methods have been proposed by researchers in NLQ. In this section, a brief review of some important contributions to the existing methodology of NQL is presented below.

N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig [14] This paper executed a procedure for synthesizing SQL questions from NL. The procedure was another NL-based program synthesis system that consolidated semantic parsing strategies from the NLP community group with sort coordinated program synthesis and computerized program repair. The technique was completely computerized, worked for any database without requiring extra customization, and did not expected clients to know the fundamental database outline. There were two key points of interest in this methodology: First, the system was utilized to answer questions on a database on which it has not been previously trained. Second, the utilization of sketch refinement enabled to deal with circumstances where the client's description does not precisely reflect the hidden database pattern. This heuristic may not function admirably if the client's NL question does not precisely coordinate the substance of the database, in such situations where the client's description utilizes a condensing or contains an incorrect spelling.

X. Xu, C. Liu, and D. Song [15] proposed a novel methodology, i.e., SQLNet, generally to tackled this issue by avoiding the sequence structure when the request does not make a difference. Specifically, the technique utilized a sketch based methodology where the sketch contains a dependency diagram so that one forecast would be possible by considering the past expectations that it relies upon. Moreover, the strategy proposed a sequence to-set model and additionally the column consideration instrument to synthesize the inquiry according to sketch. The execution accuracy was sensitive to the information in the technique, which contributes to the distinction between inquiry coordinate precision and execution accuracy.

S. Chander, J. Soundarya, R. Priyadharsini, and B. Bharathi [16] This project pointed in solving the issue of connecting and bringing the data from the database by consolidating NL called the NL Interface Relational Database System (NLIRDS). It joined the features of the AI with the RDBM). Prior frameworks manually created the semantic maps, whereas in this framework it was created naturally. This approach performed concept identification by utilizing event related ideas accessible in word-net to discover user's event from NL requirements. The detailed output that passed on the data was additionally represented in the form of tables, diagrams and charts. Extraction of valuable information out of an enormous database was made simple by this framework. The strategy focused only on bringing the data from the database. The strategy neglected to clarify the method for analyzing any NL where NLP is utilized in the interpretation of the NL to the standard dialect (English) which in turn converted into SQL.

H. van der Aa, H. Leopold, A. del-Río-Ortega, M. Resinas, and H. A. Reijers [17] proposed a combined technique such as Hidden Markov Models (HMM) and semantic matching techniques for transforming an unstructured natural language Process Performance Indicators (PPIs) description into a structured notation. The method collected the data from industry with a number of process models and PPIs from Supply Chain Operations Reference (SCOR) framework. The experimental results evaluated that the HHM method significantly tackled a problem with a more limited scope, which was highly time intensive and manual task. But, the method faced the limitations related to the transformation approach which provides an automated alternative to a highly complex task.

V. Lertnattee, and P. Pamonsinlapatham, [18] the goal of this exploration was to present blended learning for enhancing adaptability of learning SQL. The learning procedure was examined and arranged. Contents in this theme and models of RDBS were outlined and implemented. The SQLite was chosen as a RDBS administration framework because of its adaptability of utilizing and managing. After a face-to-face class, students could rehearse their activities with any of their registering gadgets, i.e., tablets and cell phones. In addition, these gadgets could work with or without Internet association. Students could rehearse and deal with their activities at any time and from anyplace. With assessments from students, the outcomes demonstrated that the adaptability of the framework upgraded their capacity for learning SQL. The technique utilized just three spaces of adapting, such as learning, psychological aptitudes and data innovation abilities. In this technique, pharmacy students were not acquainted with command lines on the server. Along these lines, no student utilized terminal customers to associate the RDBMS server.

J. Pérez, [19] proposed a Semantically Enriched Database model (SEDBM) for the Semantic Information Dictionary (SID) to solve the problems occurred in natural language interfaces to databases (NLIDBs). The experiments were carried on ATIS databases and results stated that the method was robust for obtaining the required information of most queries. The two group of undergraduate students were undergone for the experiments and NLIDB achieved 44.96% recall for correctly answered queries and 11.83% recall was obtained by ELF in first group of students. When customized by second group of students, ELF obtained 13.48% recall whereas NLIDB obtained nearly 78% recall. The method provided poor performance in complex database and also for very difficult queries.

To overcome the above issues, the focus of this work is the design of the neural architecture called STAMP method could be easily adapted by incorporating additional SQL keywords to minimize the communication gap between human and computers.

4. Proposed methodology

This paper presents the utilization of NLP for connecting with the database by utilizing NL. In this work, the paper utilizes English dialect for giving the information. Fig. 1 shows the basic design of the proposed structure. In this design the Preprocessing stage comprises of four modules, for example, Morphological examination, Semantic investigation, Mapping table and Retrieval of reports. Here the NL sentence is given as an input by the client.

- **Morphological analysis:** In Morphological examination, the client gives a NLP sentence as input and it is sent to Tokenizer. The Tokenizer split the sentences into Word depends on whitespace character. The tokenized words are taken to extractor for stemming process. In stemming process, the extractor maintains the gathering of predefined words which is utilized for correlation with the approaching new words. Predefined words are most utilized words in the report for questioning. It contrasts the tokenized words and the predefined and extract the fundamental keywords. i.e., the keywords are words that are available in the predefined list of words. At this point, from the extricated words, the root words are distinguished.

Received: January 27, 2019

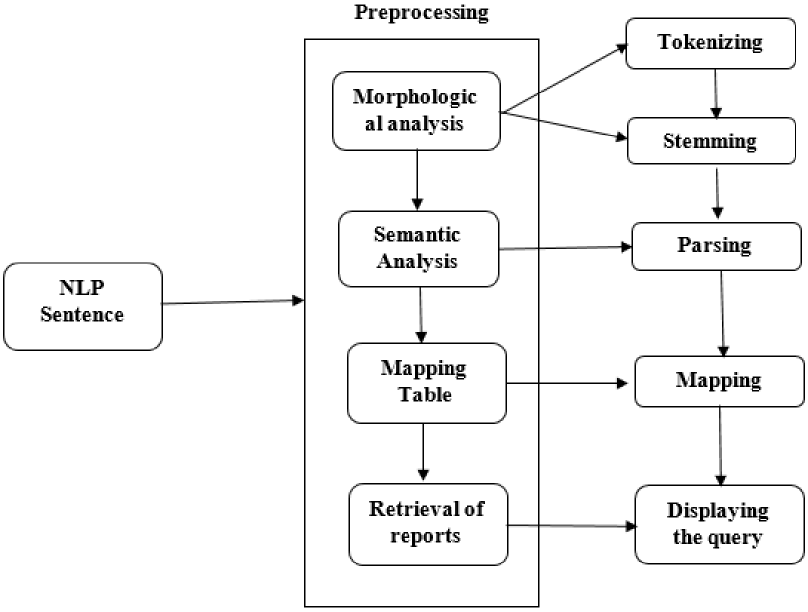


Figure. 1 An architectural layout

- **Semantic Analysis:** In the semantic investigation, the distinguished arrangement of words will be given as input. The parse tree is produced through parser and subject, object and verb present in the arrangement of words is recognized. The output of this investigation will be the accumulation of distinguished words.
- **Mapping and Retrieval:** In Mapping, the mapping table comprises of predefined set of SQL inquiries along with the most extreme possibility of NLP words. Map the accumulation of distinguished words with the mapping table and locate the best inquiry. The SQL question is created toward the end as a report from which the inquiry is picked and addressed semantically.

4.1 Task formulation and dataset

In this work, the WikiSQL task proposed is utilized. Different from most past NL2SQL datasets, the WikiSQL assignment has a few properties. To start with, it gives an extensive scale dataset so a neural system can be successfully trained. Second, it utilizes crowd-sourcing to gather the NLQ made by people, so it can beat the issue that a very well trained model may over fit to layout combined depictions. The WikiSQL is the biggest hand-annotated semantic parsing dataset to date which comprises of 87,726 inquiries and SQL questions appropriated over 26,375 tables from Wikipedia.

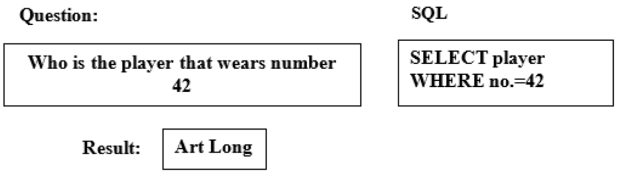
4.1.1. The WikiSQL task

Specifically, the information contains two sections: an NLQ expressing the inquiry for a table, and the construction of the table being questioned. The diagram of a table contains both the name and the sort (i.e., real numbers or strings) of every column. The output is a SQL inquiry which reflects the NLQ for the questioned table. Note that the WikiSQL task considers synthesizing a SQL question concerning a single table (Table 1). Along these lines, in an output SQL inquiry, only the SELECT condition and the WHERE statement should be anticipated, and the FROM clause can be excluded. The paper presents a model in Fig. 2.

In Fig. 2, the subject of NLQ is introduced, whereas the information is available in the Table. The consequence of SQL is portrayed in the above figure.

Table 1. Sample data for SQL database

Player	No.	Position	Year in Toronto
Antonio Lang	21	Guard-Forward	1999-2000
Voshon Lenard	2	Guard	2002-03
Martin Lewis	32,44	Guard-Forward	1996-97
Art Long	42	Forward-Centre	2002-03



```
SELECT player
WHERE no.=42
```

Figure. 2 An example task of WikiSQL

In table, the SQL results are checked. The WikiSQL assignment makes promote presumptions to make it tractable. To begin with, it expects that every section name is a significant NL depiction so the synthesis assignment is tractable from the NLQ and column names. Second, any token in the yielded SQL inquiry is either a SQL keyword or a sub-string of the NLQ. For instance, while creating a constraint in the WHERE provision, e.g., name='Martin Lewis', the token 'Martin Lewis' must show up in the NLQ as a sub-string. Third, every constraint in the WHERE statement has the type of COLUMN OP VALUE, where COLUMN is a column name, OP is one of "<, =, >, ≥, ≤", and VALUE is a substring of the NLQ.

Even though from these suspicions, the WikiSQL task is difficult to report that the state-of-the-art task-agnostic semantic parsing model [20] can accomplish an execution precision of only 37%, while the best in existing model for this task can accomplish an execution accuracy of around 60%.

4.2 STAMP: syntax- and table- aware seMantic parser methodology

Fig. 3 and 4 outlines a review of the proposed method, which is abbreviated as STAMP. There are three "channels" in STAMP, among which the column channel predicts a column name, the esteem channel predicts a table cell and the SQL channel predicts a SQL keyword. Basically, the probability of creating an objective token is figured out in Eq. (1),

$$p(yt|y_{<t},x) = \sum_{z_t} pw(yt|zt,y_{<t},x)\, pz(zt|y_{<t},x) \qquad (1)$$

Where, where zt stands for the channel selected by the switching gate, $pz(\cdot)$ is the probability to choose a channel, and $pw(\cdot)$ is a probability distribution over the tokens from one of the three channels.

One advantage of this architecture is that it inherently addresses the problem of generating partial column name/cell because an entire column name/cell is the basic unit to be generated. Another advantage is that the column-cell relation and question-cell connection can be naturally integrated in the model.

Specifically, our encoder takes a question as the input. Bidirectional Recurrent Neural Network (RNN) with Gated Recurrent Unit (GRU) is applied to the question, and the concatenation of both ends is used as the initial state of the decoder. Another bidirectional RNN is utilized to evaluate the representation of a column name (or a cell), that that every unit contains various words [21]. Basically, each channel is an attentional neural system. For cell and SQL channels, the contribution of the attention module contains the decoder hidden state and the representation of the token to be figured out in Eq. (2),

$$p_w^{sql}(i) \propto exp\left(W_{sql}\left[h_t^{dec}; e_i^{sql}\right]\right) \qquad (2)$$

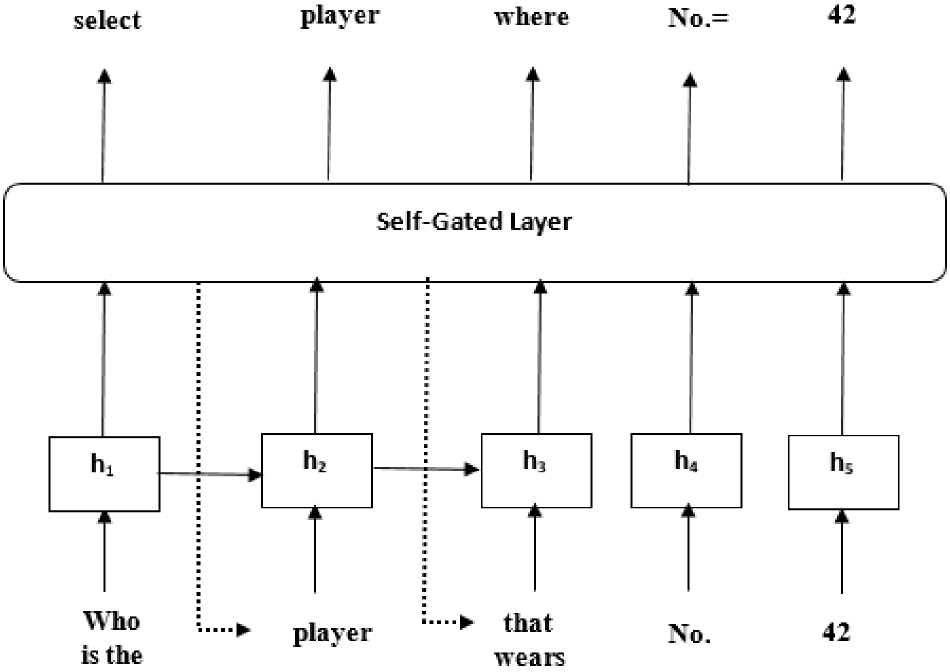
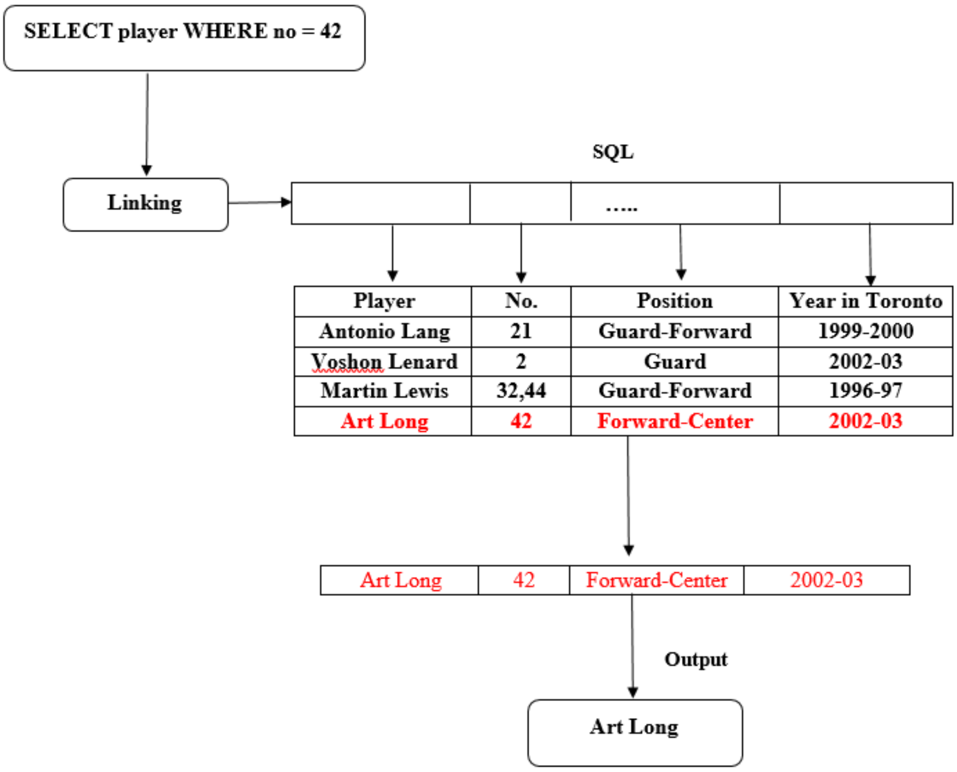


Figure. 3 Block diagram of Self- gated encoder



Player	No.	Position	Year in Toronto
Antonio Lang	21	Guard-Forward	1999-2000
Yoshon Lenard	2	Guard	2002-03
Martin Lewis	32,44	Guard-Forward	1996-97
Art Long	42	Forward-Center	2002-03

Art Long	42	Forward-Center	2002-03

Figure.4 An illustration of the proposed approach

Where, e_i^{sql} stands for the representation of the $i-th$ SQL keyword. The strategy additionally connects the inquiry representation into the contribution of the column channel to enhance the precision of the SELECT section. The strategy executes the exchanging gate with a feed-forward neural system, in which the yield is a softmax work and the input is the decoder hidden state h_t^{dec}.

4.2.1. Improved with column-cell relation

In this work, the analyst additionally enhances the STAMP method by considering the column cell connection, which is critical for forecasting the WHERE condition. On one hand, the column cell connection could enhance the expectation of SELECT column. The strategy observes that a phone or a piece of it normally appears at the inquiry acting like the WHERE esteem, for example, "anna nalick" for "anna christine nalick"). In any case, a column name may represent with a different expression, which is a "semantic gap". Assume the inquiry is "How many number of schools did player number 3 play at?" and the SQL question is "Select count School Club Team where No. = 3". Consequently, the column names "School Club Team" and "No." are not quite the same as their corresponding utterances "schools", "number" in NLQ. Therefore, table cells could be viewed as the pivot that interfaces the inquiry and column names (the "linking" segment in Fig. 4).

4.2.2. Improved with policy gradient

The model depicted so far could be expectedly learned by means of cross-entropy loss over question-SQL sets. Although, extraordinary SQL questions may be executed to yield a similar outcome, and conceivable SQL inquiries of various varieties couldn't be thoroughly covered in the training dataset. Two conceivable approaches to deal with this are rearranging the WHERE statement to create more SQL inquiries, and utilizing reinforcement learning (RL) which respects the correctness of the executed output as the goodness (reward) of the produced SQL inquiry.

5. Experimental result

The proposed method conducts experiments on the different dataset such as industry and SCOR [17] which is collected for evaluating the performance of STAMP method in terms of parameters such as precision, recall and F-Measure.

5.1 Performance measure

The method uses several evaluation metrics such as precision, recall, F-Measure to predict the performance of proposed method.

5.1.1. Precision

Precision is the extent of the anticipated positive case inquiries which are right. Precision is characterized in the Eq. (3).

$$Precision\ (P) = \frac{A}{(A+C)} \tag{3}$$

Where, A is True Positive and C is False Negative.

5.1.2. Recall

The extent of positive case inquiries which are effectively recognized. The condition of the recall can be characterized in Eq. (4).

$$Recall(R) = \frac{A}{(A+B)} \tag{4}$$

Where, B is False Negative.

5.1.3. F-measure

The metric F-Measure computes some average of the data recovery precision and recall measurements. The accompanying Eq. (5) can be depicted as below:

$$F - Measure\ = {}^{2PR}/_{(P+R)} \tag{5}$$

Where, P is Precision and R is Recall.

5.2 Evaluation measures

The experiments analyze the STAMP model from different perspectives in this part. Since SQL to all the more fine-grained assessment measurements over these aspects. The evaluation for precision, recall and F-Measure of the proposed STAMP method can be compared with existing method such as HMM [17].

5.2.1 Evaluation of precision, recall and f-measure

Table 2 shows the comparison between the performance of HMM in [17] and STAMP for various parameters such as precision, recall and F-Measure. The graphical representation are presents in Fig. 5. The existing method were evaluated in datatypes such as Industry and SCOR, whereas the proposed method also evaluated in same datatype.

Table 2. Comparison between performance of HMM and STAMP

Methods	Precision		Recall		F-Measure	
	Industry	SCOR	Industry	SCOR	Industry	SCOR
HMM [17]	0.78	0.98	0.76	0.88	0.77	0.93
Proposed-STAMP	**0.80**	**0.99**	**0.81**	**0.90**	**0.81**	**0.94**

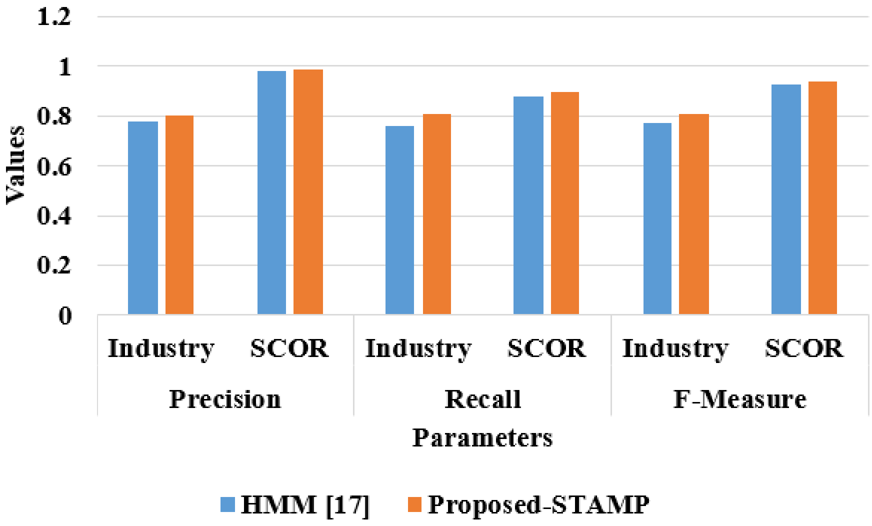


Figure. 5 Performance of proposed method

Table 3. Evaluation results for first experiment

Author	Methodology	Total Query	Recall with customization (%)	Recall with fine-tuning (%)
Joaquin Pérez, [19]	ELF	70	10	11.83
	NLIDB	70	24.28	44.69
Proposed Method	STAMP	70	27.34	52.41

Table 4. Evaluation results for second experiment

Author	Methodology	Total Query	Recall with customization (%)	Recall with fine-tuning (%)
Joaquin Pérez, [19]	ELF	70	10	13.48
	NLIDB	70	24.28	77.05
Proposed Method	STAMP	70	27.34	86.54

As it can be seen from the Table 2, STAMP has 80% precision for industry data whereas SCOR achieved 99% precision. The recall of the proposed STAMP method achieved 81% and 90% for both dataaset. The existing method achieved nearly 79% recall, F-Meaure and precision in industry dataset. In SCOR dataset, the existing method achieved nearly 95% in both precision and F-Measure.

Thus STAMP better values for recall, F-measure as compared to HMM in both datasets. The proposed STAMP method achieved 0.81 F-measure in industry dataset, whereas 0.94 F-measure in SCOR dataset. However, there is a minimal increase in Precision which can be further improved in the future work. Hence the proposed method STAMP achieves better performance than the existing system in terms of recall, precision and F-Measure.

5.3 Comparative analysis

After discharged, WikiSQL dataset has pulled in a considerable measure of attention from both industry and research networks. SEDBM [19] is compared with the proposed STAMP method. The SEDBM presented an experiment in which two groups of undergraduate students customized NLIDBs and ELF. The method provided poor performance in complex database and also for very difficult queries. Table 3 describes the performance of proposed STAMP method with existing method in terms of recall for first experiment. The first and second experiments were conducted by using 70 queries and the customization was performed by a group of 28 engineering students for customizing NLIDB.

In first experiment, the existing methods such as ELF and NLIDB method achieved 10% recall and 24.28% recall in customization for 70 queries, whereas the STAMP method achieved 27.34% recall with customization. The STAMP method achieved 52.41% recall with fine-tuning process, whereas the existing method achieved nearly 45% recall in NLIDB for 70 queries. Table 4 describes the proposed STAMP method performance in recall with customization and fine-tuning process for 70 queries.

From Table 4, the outcomes showed that STAMP performed superior compared to existing frameworks by achieving nearly 87% recall with fine-tuning process for 70 queries. From the above table, the experimental results stated that there is no improvement in second experiment for recall with customization. The large difference in the performances (52.41 and 86.54 % for STAMP) reported in Tables 3 and 4 is explained by the fact that the students of the second group are better (as revealed by their academic grades). Notice that the performance for existing methods such as ELF and NLIDB obtained by the second group is also larger than that of the first group. The existing method ELF achieved 11.83% recall in first experiment, whereas it achieved 13.48% recall in second experiment. In first set of experiment, the NLIDB method achieved 44.69% recall, but it achieved 77.05% recall in second experiment. The proposed method performed well when compared with these existing methods in both experiments because it can able to solve the complex queries from different database.

6. Conclusion

In this paper, the proposed STAMP, a Syntax-and Table-Aware seMantic Parser framework is intended to deal with difficulties in NLQ handling. The aim is to assess correct SQL interpretations for NLQ. The paper demonstrated how a modelled algorithm can be utilized to make a client friend non expert search process. The modularity of SQL change was additionally appeared. The algorithm naturally maps NL inquiries to SQL questions, which could be executed on web table or RDBS to find the solution. STAMP has three channels, and it figures out how to change to which channel at each time step. STAMP thinks about cell data and the connection among cell and column name in the generation procedure. Examinations are led on the WikiSQL datasets and results demonstrate that STAMP accomplishes the best in class execution on WikiSQL. The experimental results concluded that proposed STAMP method achieved 0.80 precision, 0.81 recall and 0.81 F-measures for industry dataset, whereas 0.99 precision, 0.90 recall and 0.94 F-measure for SCOR dataset. The WikiSQL assignment is a more suitable challenging task than others considered previously. The technique considers constructing and handling the SQL synthesis task of more complex questions as vital future work and furthermore plan to enhance the precision of the column prediction component.

References

[1] S. Kanhe, P. Bodke, A. Chikhale, and V. Udawant, "SQL Generation and PL/SQL execution from natural language processing", *International Journal of Engineering Research and Technology*, Vol.4, No. 2, 2015.

[2] A. Falle, S. Panhalkar, A. Jadhav, K. Kamble, A. Salunkhe, and D. Mirajkar, "Knowledge Extraction from Database using Natural Language Processing", *International Research Journal of Engineering and Technology*, Vol.4, No.4, pp.904-907, 2017.

[3] P. Anand and Z. Farooqui, "Rule based Domain Specific Semantic Analysis for Natural Language Interface for Database", *International Journal of Computer Applications,* Vol.164, No.11, pp.21-28, 2017.

[4] A. O. Enikuomehin and D. O. Okwufulueze, "An algorithm for solving natural language query execution problems on relational databases", *Editorial Preface*, Vol.3, No.10, 2012.

[5] E. Tinelli, S. Colucci, F. M. Donini, E. Di. Sciascio, and S. Giannini, "Embedding semantics in human resources management automation via SQL", *Applied Intelligence,* Vol.46, No.4, pp.952-982, 2017.

[6] X. Hu, D. Dang, Y. Yao, and L. Ye, "Natural language aggregate query over RDF data", *Information Sciences*, Vol.454, pp.363-381, 2018.

[7] W. Solihin, C. Eastman, Y. C. Lee, and D. H. Yang, "A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database", *Automation in Construction*, Vol.84, pp.367-383, 2017.

[8] F. Basik, B. Hättasch, A. Ilkhechi, A. Usta, S. Ramaswamy, P. Utama, and U. Cetintemel, "DBPal: A Learned NL-Interface for Databases", In: *Proc. of International Conf. On* Management of Data ACM, pp. 1765-1768, 2018.

[9] D. Saha, A. Floratou, K. Sankaranarayanan, U. F. Minhas, A. R. Mittal, and F. Özcan, "ATHENA: an ontology-driven system for natural language querying over relational data stores", In: *Proc. of International Conf. on* VLDB Endowment, Vol.9, No.12, pp.1209-1220, 2016.

[10] W. Zheng, H. Cheng, L. Zou, J. X. Yu, and K. Zhao, "Natural Language Question/Answering: Let Users Talk with Knowledge Graph", In: *Proc. of ACM International Conf. on Information and Knowledge Management*, pp.217-226, 2017.

[11] K. J. Sathick and A. Jaya, "Natural language to SQL generation for semantic knowledge extraction in social web sources", *Indian Journal of Science and Technology*, Vol.8, No.1, pp.1-10, 2015.

[12] E. C. Foster and S. Godbole, "Overview of SQL", *Database Systems*, pp. 205–209, 2016.

[13] D. Kar, S. Panigrahi, and S. Sundararajan, "SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM", *Computers & Security,* Vol.60, pp.206-225, 2016.

[14] N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig, "Sqlizer: Query synthesis from natural language", In: *Proc. of ACM International Conf. on Programming Languages*, Vol.1, No.OOPSLA, 2017.

[15] X. Xu, C. Liu, and D. Song, "Sqlnet: Generating structured queries from natural language without reinforcement learning", *arXiv preprint arXiv:1711.04436*, 2017.

[16] S. Chander, J. Soundarya, R. Priyadharsini, and B. Bharathi, "Data Analysıs of Natural Language Queryıng Usıng NLP Interface", *International Journal of Applied*

Engineering Research, Vol.13, No.8, pp.5792-5795, 2018.

[17] H. van der Aa, H. Leopold, A. del-Río-Ortega, M. Resinas, and H.A. Reijers, "Transforming unstructured natural language descriptions into measurable process performance indicators using Hidden Markov Models", *Information Systems*, Vol.71, pp.27-39, 2017.

[18] V. Lertnattee and P. Pamonsinlapatham, "Blended Learning for Improving Flexibility of Learning Structure Query Language (SQL)", In: *Proc. of International Conf. on* Blended Learning, Springer, Cham, 2017.

[19] J. Pérez, "Comparative study on the customization of natural language interfaces to databases", *Springer Plus*, Vol.5, No.1, pp.553, 2016.

[20] K. Schweinsberg and L. Wegner, "Advantages of complex SQL types in storing XML documents", *Future Generation Computer Systems*, Vol.68, pp.500-507, 2017.

[21] L. Dong, F. Wei, H. Sun, M. Zhou, and K. Xu, "A hybrid neural model for type classification of entity mentions", In: *Proc. of Twenty-Fourth International Joint Conf. on Artificial Intelligence*, pp.1243–1249, 2015.