ICSEE2019 JULY 2019 | International conference on systems energy and environment

Text to SQL Query Conversion Using Deep Learning: A Comparative Analysis

Anisha T S
PG Student, Dept. of CSE
GEC Palakkad
anishats21@gmail.com

Rafeeque P C
Professor, Dept. of CSE
GCE Kannur
rafeeqpc@gcek.ac.in

Reena Murali
Professor, Dept.of CSE
GEC Palakkad
reenamurali@gecskp.ac.in

***Abstract*—In relational databases, a significant amount of world's knowledge is stored. The ability of users to retrieve facts from a database is limited due to a lack of understanding of query languages. Intersection of Natural Language Processing (NLP) and human-computer intersection is the Natural Language Interface (NLI) which provides humans to interact with computer through the use of natural language. Here, NLI applied to relational databases translating Natural Language Queries to Structured Query Language (SQL). The proposed system is a NLP based system for converting natural language queries to database queries using Deep Neural Network. Spider, a new large-scale, cross-domain semantic parsing and text-to-SQL data set annotated by 11 college students released on November 2018 is used here. The system uses a deep learning framework such as basic sequence to sequence (Encoder-Decoder Model) and sequence to sequence plus attention. The model takes word embeddings of the query as the input and then sequence to sequence modeling is applied. The system is evaluated on the two models based on the exact matching and hardness criteria of questions. The evaluation results on Spider dataset shows that the proposed two deep learning model improves the results of text-to-SQL task. Spider dataset poses a major challenge for future research.**

***Index Terms*—Deep Learning; Natural Language Interface (NLI); Natural Language Processing (NLP); Semantic Parsing; Seq2Seq; Seq2Seq+Attention**

I. INTRODUCTION

A Semantic parsing (SP) is one of natural language processing (NLP)'s most important tasks. It requires both understanding the meaning of natural language sentences and mapping them to meaningful executable queries such as logical forms, SQL queries, and Python code. Nevertheless, prior tasks in this sector have easy but difficult definition of tasks because most of these outcomes are expected by matching semantic rather than semantic parsing. Nearly all applications in today's world use data collected to meet the intended requirements and improve their functionality. Parsing natural language questions into SQL can be viewed as a special form of machine translation. Machine translation is one of the important problems in machine learning, attracting extensive research efforts in the last few decades.

Traditional machine translation [1] is used to translate text from one language to another. In this age of digitization, relational databases are used in almost every industry to store information. Generating SQL queries from natural language has wide applications. A large amount of today's information is stored in relational databases that form the basis for key applications such as medical records, the financial market and customer relations management. The automated synthesis of database queries is a promising application domain for such computer-aided programming methods. Natural language is used in day-to-day life, and if information sharing can be made easy with the use of natural language, it will reduce the cost of learning and understanding the technology used for it. Converting natural language to structured query language is a form of machine translation. The text-to-SQL task has various application and which has been investigated in both academia and industry. It attempts to fix the issue of information recovery so that natural language questions can be used to collect the data needed.

Neural networks and deep learning were commonly applied to multiple apps for natural language processing and generate effective outcomes. There are different approaches for converting natural language to SQL query. They are Pattern-Matching Systems, Syntax-Based Systems, Semantic (linguistic) Grammar Systems. But here we are using deep neural network. A sequence to sequence model is intended to map a fixed length input with a fixed length output where the length of the input and output may differ which described in [2]. For example, translating English to Chinese or to another language cannot use a regular LSTM network to map each word since source language and target language have different length of words. That's why the proposed system uses sequence to sequence modelling.

Spider, which consists of 200 databases with various tables, 10,181 issues and 5,693 associated complicated SQL queries, all written by 11 university learners spending a total of 1,000 man-hours, addressing the need for a big and high-quality information set for a fresh complicated and cross-domain semantic parsing assignment.

First, finding many internet databases with various tables is difficult. Second, given a database, annotators must comprehend the complicated database scheme to generate a set of issues so that all SQL patterns are covered by their respective SQL queries. In addition, writing distinct complicated SQL queries is even more difficult. In addition, it requires a considerable quantity of time to review and quality-check questions and SQL pairs. All of these procedures involve very particular database understanding.

Since Spider has 200 databases with foreign keys, it is possible to divide the dataset with complicated SQL queries to avoid overlapping of train and test database. Models need to take questions and systems for databases as inputs and predict invisible queries on fresh databases. Spider, a big, complicated, cross-domain weekly parsing and text-to-SQL task that benefits NLP and DB communities directly.

Text-to-SQL assignment is being suggested based on spider, a fresh difficult and realistic semantic parsing task. The proposed system uses a deep neural network. Both sequential to se- quential and sequential to sequential plus attention model is used. Figure 1 shows the basic overview of converting natural language query to structured query language.

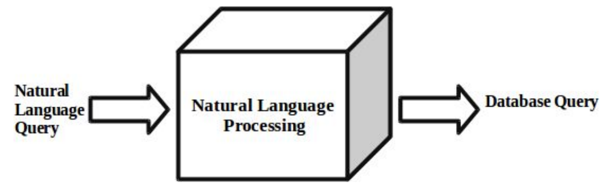

Fig. 1. Overview of NLP to SQL

To assess the task difficulty, we experiment with the above two models. The evaluation results on Spider dataset shows that the proposed two deep learning model improves the results of text-to-SQL task. Spider dataset suggests that there is a great chance for improvement for this semantic parsing task.

This paper is organized as follows: Section II gives the related works of this semantic parsing task. Section III discusses about the methods of text-to-SQL task and the stages of converting NLP query to SQL query. It also provides a comparison between the two models. Section IV discusses the results and discussion of the system. Section V provides a brief conclusion and future scope of the text-to-SQL task.

II. RELATED WORKS

Various methodologies are proposed for identifying the text-to-SQL. They are rule based approach and deep neural network based approach. Figure 2 shows the classification of proposed task of NLP-to-SQL based on the type of approach they use. Table I shows the comparison of rule based and neural network based approach. The SQL queries are labelled by [3] [4] in ATIS, Geoquery datasets. The other datasets are Restaurants [5], Scholar [4], Academic [6], Yelp and IMDB [7], Advising [8] and WikiSQL [9]. In the proposed system, *Spider* dataset is used. Annotation, collection, creation and construction of corpus is explained in [10].

People are not speaking the language of the database. Machines are Structured Query Language (SQL) while people are speaking. We all want to make better decisions by using data, which means we need to overcome major obstacles such as tedious dashboards or SQL scripts. One way to solve it would be to convert natural language to SQL like Google Translate is translating Japanese to English.

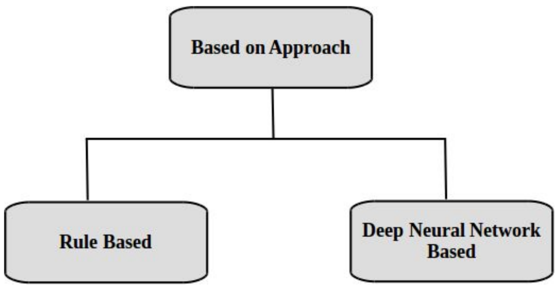

Fig. 2. Classification of Approaches

Traditionally, the grammar-based approach implies that a human is involved in the process of developing and improving the system. The biggest advantage of formal grammar is that there is always a way to check if a user's query could be processed by the system. Rules are written by people, so it is easy to locate and fix any reported bug. The Rule Based System (RBS) requires deep domain understanding and a lot of manual job. That is the main disadvantage of RBS. It is quite difficult and time consuming to create guidelines for a complex system. The scheme will produce the outcome according to the regulations, so that the system's learning ability alone is much less. If an application you want to construct is too complicated, it can take a lot of time and analysis to construct the RBS. Identifying complex patterns is a difficult job in the RBS.

Amit Pagru et. al. [11], the objective is to develop a system that converts a natural language statement into a MySQL query to retrieve data from the respective database. Various OpenNLP natural language processing techniques like tokenization, parts of speech tagging, stemming and lemmatization is used to get the statement in the desired form. The statement is further processed to extract the query type, the basic clause specifying the required entities from the database, and the condition clause specifying the basic clause constraints.The final query is generated by converting the basic and then concatenating the condition query to the basic query. It only works with MySQL database.

Tanzim Mahmud et.al. [12] proposed an approach for NLP based query processing. Proposes an easy-to-use approach to access the database with no knowledge of the query language. It makes it easier for ordinary people to use a great promise for computer interfaces. Because of this, People can communicate to the computer in their own language. For testing, a prototype system is developed.

Since rule based system requires time consuming and lots of manual work, deep learning came into existence. There are various methods in deep neural networks for neural machine translation and also there are so many applications. Machine Translation is a technology which runs behind Google Translate. It has changed the world by allowing people to communicate. The deep neural networks are still learning in parallel texts. The technology behind this machine translation is **sequence-to-sequence learning.**

X. Xu et.al. [13] proposes SQLNet to convert nlp to SQL. It uses a sequence-to-set model to generate SQL queries when order does not matter. It introduces the column attention mechanism to improve the accuracy. P.S.Huan et.al. [14] proposes a new learning protocol that reduces a supervised learning problem to the few-shot meta-learning. It creates pseudo-tasks with the help of a relevance function. It is evaluated on WikiSQL dataset.

V. Zhong et.al. [9] suggested Seq2SQL, a profound neural network for SQL query translation. The model uses the structure of the SQL database to considerably decrease the room produced for the query output. It utilizes in - the-loop query execution rewards across the database to learn a strategy for generating unordered components of the query. A policy-based reinforcement learning is implemented to WikiSQL using a query execution environment. The model Seq2SQL is better than the sequence of attention to sequence models. Here is used WikiSQL consisting of issues and SQL queries which is a magnitude order bigger than similar datasets.

TABLE I
COMPARISON OF TWO APPROACHES

Approach	Paper	Merit	Demerit
Rule Based	A Rule Based Approach for NLP Based Query Processing	1. Dictionary method 2. Efficient and fast related to response	1. Difficult to handle complex sentences 2. Slow
	Automated SQL Query Generator by Understanding a Natural Language Statement	1. Handles complex queries	1. System works only with MySQL database.
Deep Neural Network	Seq2SQL: Generating Structured Queries from Natural Language using Reinforcemen	1.Reinforcement learning is used	1.WikiSql is difficult to parse

	t Learning		
	Natural Language to Structured Query Generation via Meta-Learning	1. Few-shot meta-learnin g is used	1.WikiSql is difficult to parse

Li Dong and Mirella Lapata [15] proposes an encoder-decoder neural network model for mapping natural language descriptions to their meaning. Using Recurrent Neural Networks (RNN) with Long Short-term memory (LSTM), the model encodes natural language utterances into vectors and generate their corresponding logical forms as sequences or trees. SEQ2TREE model to related structured prediction tasks such as constituency parsing is going to be used as future work in this paper.

Siddhant Srivastava et.al. [16] aims to build a machine translation system. Here, machine translation with statistical approach is used. Deep learning models in improving different components of Statistical Machine Translation (SMT) is done first and then shifted to end-to-end neural machine translation (NMT). Singh, Shashi Pal, et al. [16] uses Deep Neural Network (DNN) and concept of deep learning in the field of natural language processing for machine translation. Recursive recurrent neural network (R2NN) is a best machine learning technique. It is the combination of recurrent neural network and recursive neural network. It is known as the combination of *Recursive AutoEncoder.* The paper proposes how to train the recurrent neural network by using semi-supervised learning methods to reorder for source to target language. Word2vec tool is needed to generate source language word vectors and Auto Encoder helps us rebuild target language vectors in tree structure.

Ms. Neeta Verma et.al. [17] proposes a survey on neural machine translation. Aims to build a single neural network that can be aligned together to maximize performance, translation and efficiency. It belongs to a group of encoders and decoders.It encode a source text or phrase in a fixed-length vector. It describes different approaches for language translation. The survey proposes Hidden Markov Model(HMM) based encoder decoder models.

There are two python packages ln2SQL and SQLNet which are both open sources but the main issue with these two is that they are too narrow. There are various other technologies such as Artificial Intelligence combined with rule-based NLP. But it provides the difficulty of creating the parsers. All the deep learning techniques discussed above provides good results and performance but accuracy is low and it is because of the dataset. The main issue lies with the dataset and that's why here uses Spider Dataset which is better than the previous one.

III. TEXT-TO-SQL

Text-to-SQL is a task of converting natural language query to structured query language. The system proposes a deep neural network. Encoder-Decoder model also known as sequence to sequence model can be used for this task. Also applied attention mechanism to the encoder-decoder model. Figure 3 shows the basic architecture of text-to-SQL task. The main stages are data-set collection, preprocessing, word-embedding and neural network.

A. Dataset Collection

Spider is a big, complicated, cross-domain semantic parsing and text-to-SQL dataset that benefits directly both NLP and DB communities published in November 2018. Eleven computer science learners who were indigenous English speakers wrote and evaluated all questions and SQL queries. Develop the dataset as outlined by Tao Yu, Rui Zhang et al. in five steps. They spend a total of about 1,000 hours of human labor.

The five steps are:

1) Database Collection and Creation
2) Question and SQL Annotation
3) SQL Review
4) Review and paraphrase of the questions
5) Final question and review of the SQL

For table and column names, abbreviations in databases are commonly used. For example, student id might be represented by stu id. And then converted to excel format.

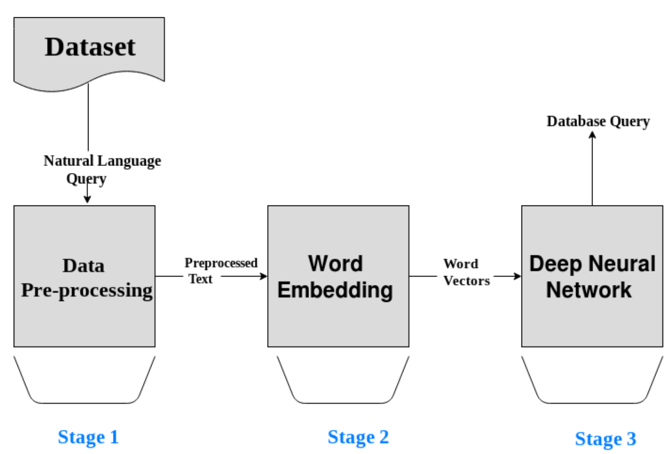

Fig. 3. Basic Architecture of Text-to-SQL Task

B. Preprocessing

The paper proposes word level processing. It is same as the character level processing but here instead of making a dictionary of characters, we make a dictionary of the words used in the text we want to process or sometimes we use the most frequent 10,000 words of the texts. Convert text to lowercase and uses the Spacy tool for tokenization and make it into lower case. Append START and END to the target data. START to the Start of Sentence and END to the End of Sentence. Make dictionaries to convert words to indexed numbers.

C. Word Embedding

A Word Embedding format usually attempts to map a word to a vector using a dictionary. Almost all deep learning algorithms are not capable of processing inputs in the form of strings or text. Word embedding helps to map the text input to vector representations so that they can be understood by deep learning algorithms.

In the proposed system, the word embeddings are generated using GloVe model. GloVe, for Global Vectors, is a model for creating word embeddings based on the global corpus statis- tics. It is trained on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. GloVe is a global log-bilinear regression model for unsupervised learning of word representations that outperforms other models on word analogy, word similarity, and named entity recognition tasks.

The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. That is, the context of the word has an influence on the position of that word in the vector space. Given a large corpus, GloVe initially constructs a co-occurrence for corpus and then factorize to yield a lower dimensional matrix, where each row is the vector representation of a word.

D. Deep Learning Model

Sequence to sequence model is used for building the model. The glove embeddings is passed through the encoder decoder model. The encoder model consists of the Input layer, Embed- ding layer and LSTM layer. The result of the encoder module will be a state vector or context vector. This state vector will be the input for the decoder module. The decoder module consists sof Input layer, Embedding layer, LSTM layer and a Dense layer. The encoder and decoder model is learning the method of *Teacher Forcing.*

Teacher Forcing [18]is a strategy to train recurrent neural networks using model output as an input from a previous time step. The approach was originally described and developed for the training of recurrent neural network as an alternative technique for backpropagation through time. Forcing teachers work by using the actual or expected output from the training dataset at the current step y(t) as input in the next step X(t+1) instead of the output generated by the network.

For long sentences, basic encoder decoder model will not work. So attention mechanism is used. Here context vector will be weighted sum of all the past encoder states. Attention Keras takes a more modular approach, implementing attention at a more atomic level (i.e. for each decoder step of a given RNN / LSTM / GRU decoder). The attention layer is built using Bahdanau Attention as described in [19]. It supports attention visualization.

IV. RESULT AND DISCUSSION

A. Dataset

The statistics of Spider and other text-to-SQL datasets is summarized in Table II. A new complex and cross-domain semantic parsing and text-to-SQL task where different complex SQL queries and databases appear in train and test sets known as *spider*. Compared to other datasets, Spider contains multi-table databases and SQL queries including many complex SQL components. Spider contains about twice as many nested queries such as ORDER BY and GROUP BY than the previous text-to-SQL datasets. The proposed system uses different databases for training and testing, evaluating the cross-domain performance.

TABLE II
COMPARISONS OF TEXT-TO-SQL DATASETS.

Dataset	**Q**	**SQL**	**DB**	**Complex Query**
ATIS	5280	947	1	320
GeoQuery	877	247	1	233
WikiSQL	80,654	77,840	26,521	0
Spider	10,181	5,693	200	4058

Spider is the only text-to-SQL dataset that includes multi-domain databases and com-plex SQL queries with various tables. It was intended to evaluate the capacity of a system to generalize not just fresh SQL queries and database schemes, but new domains as well. Spider has 200 distinct databases with 138 distinct fields, including college, business, restaurants, student, academic, etc. Most domains have one database, containing 20-50 questions, and various domains such as flight data have various databases with a total of over 100 questions. The Spider has an average of 27.6 rows and 8.8 foreign keys. The average length of the question and length of the SQL are about 13 and 21 respectively. Details of spider dataset is shown in Table III.

A. Experimental Setup and Results

Our proposed two deep learning model is implemented with Keras with Tensorflow [20] as the backend. The performance of the proposed text-to-SQL system is compared with the existing seq2seq model with Geoquery as Dataset as in [21]. The basic sequence to sequence model are trained using a batch size of 64 for 150 epochs. The sequence to sequence model plus attention are trained using a batch size of 32 for 400 epochs.

TABLE III
DETAILS OF SPIDER DATASET

Sl.No:	Feature	Count
1	Questions	10,181
2	SQL Queries	5,693
3	Databases	200

The evaluation metrics include Exact matching and SQL Hardness Criteria. We measure whether the totality of the predicted query is equivalent to the question of ground truth. We divide SQL queries into three levels of order to better understand the performance of the model on different queries: easy, medium, hard. The difficulty based on the number of SQL components, selections and conditions, so that queries containing more SQL keywords (GROUP BY, ORDER BY, INTERSECT, nested sub queries, column selections and aggregators etc.) are considered to be more difficult. For example, a query is considered hard if it includes more than two SELECT columns, more than two WHERE conditions, and GROUP BY two columns, or contains EXCEPT or nested queries.

TABLE IV
ACCURACY OF EXACT MATCHING ON SQL QUERIES WITH DIFFERENT HARDNESS LEVELS

Dataset	Model	Easy	Medium	Hard
Spider	Seq2Seq	75%	64.9%	53%
	Seq2Seq + Attention	55.3%	60%	80.49%
GeoQuery [21]	Seq2seq	53.4%	52.8%	51.6%
	Seq2seq + SQLTokenize	58.2%	57.4%	57.2%

The performance metrics used to evaluate the model is accuracy, and Table IV shows the accuracy performance of all models. The accuracy values were calculated using the formula (TP + TN)/(TP + TN + FP + FN). TP is the true positive rate, the number of elements correctly identified by the framework. FP is the false positive rate. TN is the true negative rate while FN is the false negative rate.

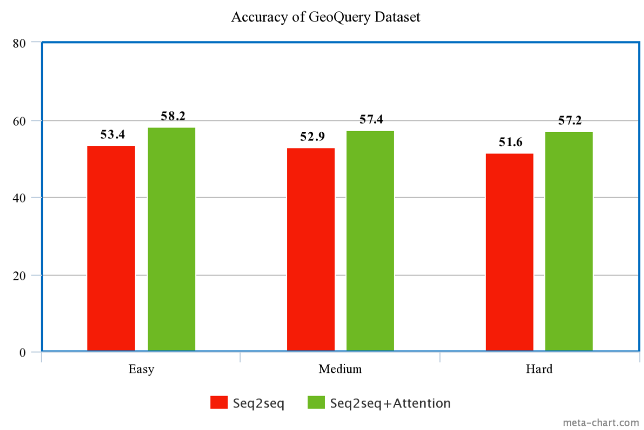

Fig. 4. Accuracy of models in GeoQuery dataset

Table IV is based on two datasets. One is our proposed model using Spider dataset and another one using GeoQuery dataset. The basic sequential model improves accuracy for easy and medium questions whereas for hard queries, atten- tion model improves the performance. Since GeoQuery have limited amount of data and less number of complex queries, it doesn't provide good results compared to Spider. Figure 4 and 5 plots the accuracy of models in spider and GeoQuery dataset.

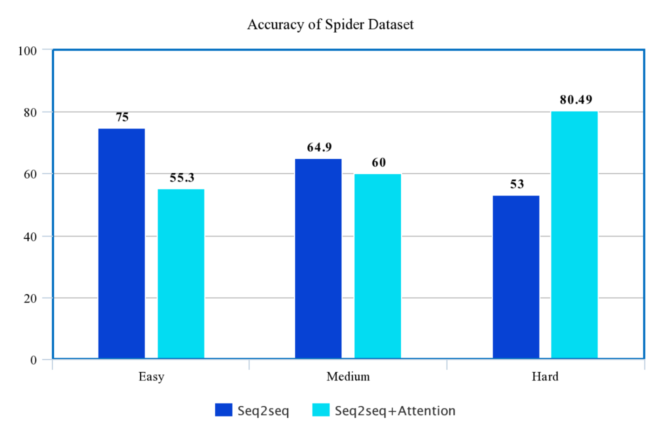

Fig. 5. Accuracy of models in spider dataset

BLEU score [22] can be also used for the evaluation of the generated SQL queries. A metric for assessing a produced phrase to a reference phrase is the Bilingual Evaluation Understudy Score, or BLEU. A perfect match result in a score of 1.0, while a perfect mismatch results in a score of 0.0. To assess the efficiency of automatic machine translation systems, the score was created. The BLEU score for medium questions is given in Figure 6.

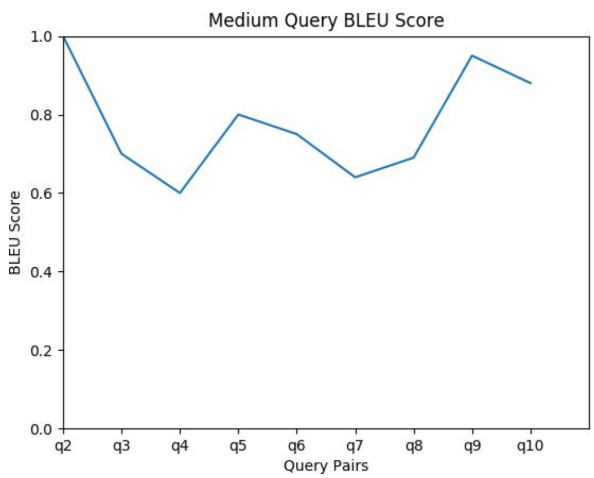

Fig. 6. BLEU Score of Medium Queries

V. CONCLUSION AND FUTURE SCOPE

The proposed system is a text-to-SQL system which converts natural language to SQL query with deep neural net- work. The deep neural network used here is sequence to sequence network. Basic encoder-decoder model and encoder- decoder model with attention is used. The proposed system handles three types of queries: easy, medium and hard. It is assessed on Spider dataset, a big, complicated, cross-domain semantic parsing and text-to-SQL dataset that benefits both NLP and DB communities directly. The system is evaluated based on exact matching and SQL Hardness Criteria. To understand the performance of the model, queries are divided into three levels such as easy, medium, hard. The attention model gives better results for hard queries. The proposed system is also compared with existing sequence to sequence modelling with GeoQuery as dataset. The overall performance of all models are not satisfactory, indicating that our task is challenging and there is still a large room for improvement.

Further improvements can be made by handling extra hard queries. Sequence to sequence plus copying in which certain segments in the input sequence are selectively replicated in the output sequence can be applied. Spider dataset presents a challenge for the model to generalize to new databases. Experimental findings on these state-of - the-art models indicate plenty of room for improvement on this assignment.

REFERENCES

[1] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[2] K. Cho, B. Van Merrie¨nboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[3] A.-M. Popescu, O. Etzioni, and H. Kautz, "Towards a theory of natural language interfaces to databases," in *Proceedings of the 8th international conference on Intelligent user interfaces*, pp. 149–157, ACM, 2003.

[4] S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, and L. Zettlemoyer, "Learning a neural semantic parser from user feedback," *arXiv preprint arXiv:1704.08760*, 2017.

[5] L. R. Tang and R. J. Mooney, "Using multiple clause constructors in inductive logic programming for semantic parsing," in *European Conference on Machine Learning*, pp. 466–477, Springer, 2001.

[6] F. Li and H. Jagadish, "Constructing an interactive natural language interface for relational databases," *Proceedings of the VLDB Endowment*, vol. 8, no. 1, pp. 73–84, 2014.

[7] N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig, "Sqlizer: query synthesis from natural language," *Proceedings of the ACM on Programming Languages*, vol. 1, no. OOPSLA, p. 63, 2017.

[8] T. Yu, M. Yasunaga, K. Yang, R. Zhang, D. Wang, Z. Li, and D. Radev, "Syntaxsqlnet: Syntax tree networks for complex and cross-domaintext-to-sql task," *arXiv preprint arXiv:1810.05237*, 2018.

[9] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," *arXiv preprint arXiv:1709.00103*, 2017.

[10] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," *arXiv preprint arXiv:1809.08887*, 2018.

[11] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle, "Ontology-based extraction and structuring of information from data-rich unstructured documents," in *CIKM*, vol. 98, pp. 52–59, 1998.

[12] T. Mahmud, K. A. Hasan, M. Ahmed, and T. H. C. Chak, "A rule based approach for nlp based query processing," in *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, pp. 78–82, IEEE, 2015.

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

[14] P.-S. Huang, C. Wang, R. Singh, W.-t. Yih, and X. He, "Natural lan guage to structured query generation via meta-learning," arXiv preprint arXiv:1803.02400, 2018.

[15] L. Dong and M. Lapata, "Language to logical form with neural attention," arXiv preprint arXiv:1601.01280, 2016

[16] S. Srivastava, A. Shukla, and R. Tiwari, "Machine translation: From statistical to modern deep-learning practices," arXiv preprint arXiv:1812.04238, 2018.

[17] H. H. Nasereddin and A. A. R. Omari, "Classification techniques for automatic speech recognition (asr) algorithms used with real time speech translation," in 2017 Computing Conference, pp. 200–207, IEEE, 2017.

[18] N. Toomarian and J. Barhen, "Fast temporal neural learning using teacher forcing," June 27, 1995. US Patent 5,428,710.

[19] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition,"in 2016 IEEE International Conference on Acoustics, Speech and SignalProcessing (ICASSP), pp. 4945–4949, IEEE, 2016

[20] A. Gulli and S. Pal, Deep Learning with Keras. Packt Publishing Ltd, 2017.

[21] E. Varsha and P. Rafeeque, "Translating natural language sentences into database query," in 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), pp. 1–6, IEEE, 2018.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in Proceedings of the 40th annual meeting on association for computational linguistics, pp. 311–318, Association for Computational Linguistics, 2002.