ISSN 2395-1621

Natural language to SQL query conversion using Deep Learning

Kshitij Narvekar, Krishna Gardharia, Siddharth Bahekar, Prof. Manjusha Amritkar

knarvekar31@gmail.com, gardhariakrishna@gmail.com,
siddharthbahekar13@gmail.com, manjusha.amritkar@gmail.com

dept. of Information Technology

International Institute of Information Technology Pune, India

ABSTRACT

Almost each and every application people used today is leveraging the power of abundant data in its own ways to provide exceptional functionalities. The main objective remains the efficient storage and fast retrieval of this data. Databases especially relational databases are the first and foremost choice for fulfilling these objectives. As industries have become more data driven than ever before the ability for anyone in the industry to quickly access and use the data for their benefit is very crucial. While it might be great if every worker just learned how to write perfect SQL queries, it'snot going to happen anytime soon. Thus, organizations are looking for means and ways to reduce the burden on data management teams and allow non-developers to query their databases. Our solution is a part of NLIDB which focuses on converting the Natural language question into SQL query by using deep Neural Networks.

ARTICLE INFO

Article History

Received: 8th March 2020

Received in revised form :

8th March 2020

Accepted: 10th March 2020

Published online :

11th March 2020

I. INTRODUCTION

Building a SQL query can be difficult task for non-technical users. Thus, we need a way of converting natural language questions into its equivalent SQL query which in itself is a challenging problem. We are considering the WikiSQL (Zhong, 2017) which is a hand -annotated dataset for SQL queries and corresponding natural language English questions., and Spider (Tao Yu, 2019), which is a large- scale complex and cross-domain semantic parsing and text- to-SQL dataset annotated by 11 Yale students. Given a natural language question, the system needs to produce a SQL query with the help of two types of neural networks.

Our system combines the powers of two neural networks. The first neural network is used to categorize the type of query and the second network is used to understand the structure of the different queries. It is not necessary that the user always use the terms which exactly resemble the column names in the tables. Thus, our further scope is to implement reinforcement learning to better understand what the user is querying.

II. RELATED WORK

Semantic parsing has been in the forefront of natural language processing tasks. There are many existing implementations that make use of semantic parsing for the generation of queries. Nowadays sequence to sequence models have proven to be achieving over 80% exact Some earlier work includes semantic parsing using small number of programs but the same dataset is used for training as well as testing the programs which has shortcomings when dealing with modern data intensive models (John M. Zelle , 1996), (Fei Li, 2014),(NavidYaghmazadeh, 2017).

Semantic parsing has been used extensively in earlier implementations of this task, but the query structures generated were not accurate enough as compared to the target query. To overcome this and for the ability to solve nested queries on unseen databases we use specific syntax trees for complex and cross-domaintext-to-SQL task. (TaoYu, 2018). It exploits syntax information for code generation tasks. (Yin and Neubig, 2017) (Rabinovich2017). Introduced a neural model that transduces a natural language statement into an abstract syntax tree (AST).

The best way to approach this problem then seems to be a sequence to sequence model which learns the structure of the queries along with its equivalent natural language question. This has been employed with Attention mechanisms (Li Dong, 2016) which treats semantic parsing as a vanilla sequence transduction task and uses attention mechanism to handle rare mentions of entities and numbers. This has also been modified with bidirectional attention (Tong Guo, 2018).

There has been work done to improve the results of sequence models by using reinforcement learning (Zhong, 2017) but on state-of-the-work WikiSQL, it reports an improvement of only 2%.

Our model thus aims at improving the sequence to sequence accuracy by not only using reinforcement learning after the sequence has been formed (Zhong, 2017) but also using a convolution network prior to the sequence to sequence model. Convolution neural networks have been proved to be effective at text classification tasks (Xiang Zhang, 2015). CNN can be used to classify among the type of queries like INSERT, SELECT, UPDATE, DELETE. Although there is a lack of datasets which include queries other than SELECT. Thus, the CNN will be trained to classify SELECT queries into ones with WHERE clause, ones with JOIN and ones with Sub queries. This will in turn be trained on separate sequence to sequence models which will achieve higher accuracy as compared to a single sequence to sequence model trained on the entire dataset.

This is done with keeping in mind an upcoming dataset which will include all types of queries, which will be easy to incorporate with our model.

III. MODEL

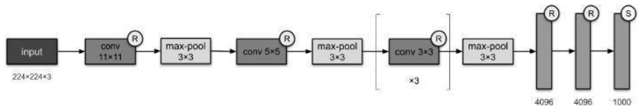
Our project will include two neural networks; including a Convolutional Neural Network and a Recurrent LSTM Neural Network which will help map the Natural Language statement into an ordered SQL query.

The system can be split into distinct three steps:

- A Convolutional Neural Network implementing AlexNet architecture for classifying the type of SQL query. i.e. (INSERT, SELECT, DELETE)
- A Recurrent/LSTM Neural Network used as a Sequence to Sequence model for predicting the sequence of SQL query based on the Natural Language statement.

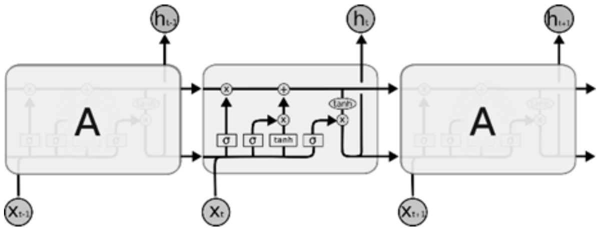
The CNN model responsible for classification of query type will be using ngram Word2Vec (Mikolov, 2013) fixed embeddings. The embedding will result in a character level model which is proved to be better than word level models

The CNN architecture used is AlexNet (Alex Krizhevsky, 2012) which is a 8 layer neural network with 5 convolution and 3 fully connected layers. It was the first architecture to make use of the ReLU activation function



A. LSTM

Long Short-Term Memory, are a special kind of RNN introduce by (Hochreiter&Schmidhuber, 1997). While learning entire sequences of strings like any SQL query, a later part of the query say the WHERE clause maybe corresponding to a part of the question that came much before in the sequence and in order to understand this relation RNN lack the capability to hold memory for a long time. Thus, we need something like LSTM in order to understand these long- term dependencies,



The horizontal line running through the top of the diagram is responsible for maintaining the cell state.

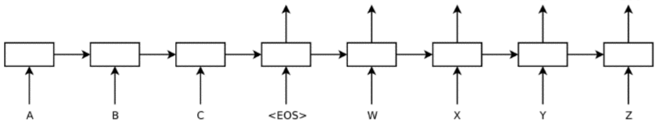
LSTM provide a structure which allows holding the state of data for a long period and keeps it unchanged. It achieves this by using a simple linear architecture which makes information to flow easily.

They use different types of gates which allow the removal of selective information and allow to add more relevant information which is made of a sigmoid layer and uses pointwise multiplication operation.

B. Sequence to Sequence networks –

A typical sequence to sequence model as given by (Ilya Sutskever, 2014) has two parts–an encoder and a decoder. Both the parts are practically two different neural network models combined into one giant network. What an encoder network basically does is understand the structure of the input sequence in our case the natural language question or statement and create a smaller dimensional representation of it.This smaller dimension version is then passed on to a decoder network which generates a sequence that represents the output.

The main reason for the use of this type of model is that conventional neural nets like convolutional networks use a specific size of input and cannot handle varying input sizes and as well as not correlated output sizes. Thus, using such architecture allows us to solve a range of new problems



IV. RESULT

We shall compare our results with –

SEQ2SQL model created by (Zhong, 2017) as they have implemented an augmented pointer network and generated the final SQL query using reinforcement learning. Attentional sequence to sequence neural semantic parser by (Dong&Lapata, 2016) performs exceptionally on semantic parsing datasets which was not possible by non-neural semantic parsers while not even using hand-engineered grammar.

We are using PyTorch for implementing out neural networks. We have used a fraction of Spider dataset to train out models. The models were run for maximum of 200 epochs using Adam optimizer introduce by (Diederik, 2015).

The CNN model implemented using Alex architecture achieves more than 95% accuracy in categorizing the type of SELECT query.

This is where our approach separates itself from other as none of the prior has used classification of queries. This is necessary in order to choose which Sequence model to use ahead. Our proposal is that the sequence model will perform well when trained on specific sequences as a query with JOIN clause and one with WHERE clause have separate sequences. Also, the ones with WHERE and JOIN clauses together have a higher complexity which may restrict the model form performing well on simple queries.

The Sequence to Sequence model achieves 53% accuracy for categorizing the SELECT queries with WHERE clause and 45% accuracy for queries with JOIN clause.

V. CONCLUSION

Almost all the implementations focusing on the natural language to SQL query conversion task were heavily relying on semantic parsing. Also, there was later use of Sequenceto Sequence models which understood the order much better. None of the solutions up until now have classified the types of queries.

We proposed a combination of deep neural networks for the translation of natural language questions to SQL query. Our model leverages the structure of the SQL query by using sequence to sequence model.

Prior to sequence to sequence model the CNN will help in classifying the type of query, i.e. whether it has a WHERE or JOIN clause.

The CNN could serve a higher purpose with enough types of queries to train on including INSERT, DELETE and UPDATE etc. Also, the implemented sequence model would perform much better with the use of Attention mechanism.

The overall results could also see a change with the use of Reinforcement learning.

REFERENCES

[1] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. arXiv preprint arXiv:1709.00103, 2017.

[2] John M Zelle and Raymond J Mooney. Learning to parse database queries using inductive logic programming. In Proceedings of the national conference on artificial intelligence, pp. 1050– 1055,1996.

[3] Pengcheng Yin, Zhengdong Lu, Hang Li, and Ben Kao. Neural enquirer: Learning to query tables with natural language. arXiv preprint arXiv:1512.00965, 2015.K. Elissa, “Title of paper if known,” unpublished.

[4] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. Sqlizer: Query synthesis from natural language. In OOPSLA,2017.

[5] Fei Li and HV Jagadish. 2014. Constructing an interactive natural language interface for relational databases. VLDB.

[6] Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. 2017. Sqlizer: Query synthesis from natural language. Proc. ACM Program. Lang., 1(OOPSLA):63:1–63:26.

[7] Tao Yu Michihiro Yasunaga Kai Yang Rui Zhang Dongxu Wang Zifan Li Dragomir R. Radev. 2018. SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task. arXiv:1810.05237v2 [cs.CL] 25 Oct 2018

[8] Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In ACL (1), pages 440–450.
Association for Computational Linguistics

[9] Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7 -12, 2016, Berlin, Germany, Volume 1: Long Papers.

[10] Tong Guo, Huilin Gao. 2018. Bidirectional Attention for SQL Generation.arXiv:1801.00076v6 [cs.CL] 21 Jun 2018.

[11] Xiang Zhang Junbo Zhao Yann LeCun. 2016. Character-level Convolutional Networks for Text Classification. arXiv:1509.01626v3 [cs.LG] 4 Apr 2016.

[12] Mikolov, T,Sutskever, I, Chen, K, Corrado, G. S. and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, 3111–3119.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097– 1105, Lake Tahoe, California, USA.

[14] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural computation, 1997.

[15] Ilya Sutskever, Oriol Vinyals and Quoc V. Le. 2014. Sequence to Sequence learning with neural networks. arXiv:1409.3215v3 [cs.CL] 14 Dec 2014.