Pattern Recognition

Joris ten Tusscher, Mats Veldhuizen, Jordy van Vliet, Jan Vrieler

December 2018

1 Introduction

Many organizations make use of relational databases. These databases are very useful, but require skill and knowledge to effectively interact with. Knowledge of SQL is not a common skill in employees, making the use of these relational databases more difficult for many companies. The field of Natural Language Interfacing, or NLI in short, provides ways for humans to interact with computers using natural language (Androutsopoulos et al.[1]).

Recent advancements in this field focus on converting natural language to useable SQL queries using neural networks. Zhong [2] created a neural network in 2017 which produced queries with an accuracy of 35,9%. One year later, in 2018, Shi and Tatwawadi[3] produced a network with an accuracy of 87,1%. This research will focus on the differences between the neural nets created in these papers, and on which advancements have been made that led to the improvements in accuracy. Both neural networks will be replicated, and their outputs will be analyzed.

2 Data and related work

The data used in this research is known as the WikiSQL dataset [2], which is an open-source dataset. It is a corpus of 80.654 instances of questions posed in natural language, SQL Queries, and SQL tables extracted from 24241 HTML tables from Wikipedia.
Multiple researchers have designed neural networks for the WikiSQL database. Links to their papers and their accuracy can be found on the WikiSQL Github repository[1] under the section Leaderboard. Here one can also find links to the corresponding papers.

[1] https://github.com/salesforce/wikisql

3 Network architecture

Our research will focus on two papers from the WikiSQL Github repository. The papers are written by Zhong [2] and Wang [4]. Zhong makes use of LSTMs, multi-layer perceptrons and pointer networks [5]. Wang uses LSTMs, pointer networks and bi-directional recurrent neural networks. Wangs paper performs better due to the addition of execution-guided decoding, which boils down to preferring valid partial SQL queries over queries that have a higher probability according to the neural network, but are not valid SQL.

4 Result evaluation

The leader board presented in the Github repository makes use of two different forms of accuracy: logical form accuracy, and execution accuracy. Therefore, we use those performance measures as well. Logical form accuracy checks whether the query produced by the network matches the query used as training data. The execution accuracy simply checks whether the produced query produces the right results. Additionally, the length and complexity of the produced query could be taken into account when the models perform differently in that regard.

5 Planning

We are not exactly sure yet who will do what, but the overall planning can be found in the following table.

Week	Activity
2	Implementing Seq2Sql
3	Extending Seq2Sql with Execution-Guided Decoding Hyperparameter tuning Data analysis
4	Writing paper Preparing and giving presentation

References

[1] G.D. Ritchie Androutsopoulos and P. Thanisch. Natural language interfaces to databases - an introduction. 1995.

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

[3] T. Shi and K. Tatwawadi. Incsql: Training incremental text-to-sql parsers with non-deterministic oracles. *arXiv:1809.05054v2 [cs.CL] 1 Oct 2018*, 2018.

[4] Chenglong Wang et al. Robust text-to-sql generation with execution-guided decoding. *arXiv:1807.03100v3 [cs.CL]*, 2018.

[5] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.