Text-to-SQL Translation with Various Neural Networks

Yipeng He, Di Bai, Wantong Jiang
{yipenghe, dibai, wantongj}@stanford.edu

Overview

Generating SQL queries from natural language questions to help people easily retrieve data from databases has long been an interesting but challenging problem. In this project we explore and evaluate different deep neural networks for this task.

- Baseline: LSTM based seq-to-seq model.
- SQLNet with GloVe and BERT embeddings.
- Transformer and CNN based aggregator prediction

Dataset

- Wiki SQL Dataset
- Contains 80654 SQL queries extracted from 24241 HTML tables from Wikipedia

Text query:

```
Which school did Herb Williams go to?
```

Corresponding SQL query:

```
SELECT school_name
WHERE  student_name = 'Herb Williams'
```

Slot filling strategy

- Instead of predicting the whole SQL query at once, break the task into 3 subtasks. Predict aggregator, select clause column and conditions with subnetworks.
- We focused on trying to use Transformer for the SELECT clause (predicting $AGG and $SELECT_COL)

```
SELECT $AGG $SELECT_COL
WHERE $COND_COL $OP $COND_VAL
(AND $COND_COL $OP $COND_VAL)*
```

References

1. Xiaojun Xu, Chang Liu, and Dawn Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436*, 2017
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems* , pages 5998–6008, 2017.

Approaches

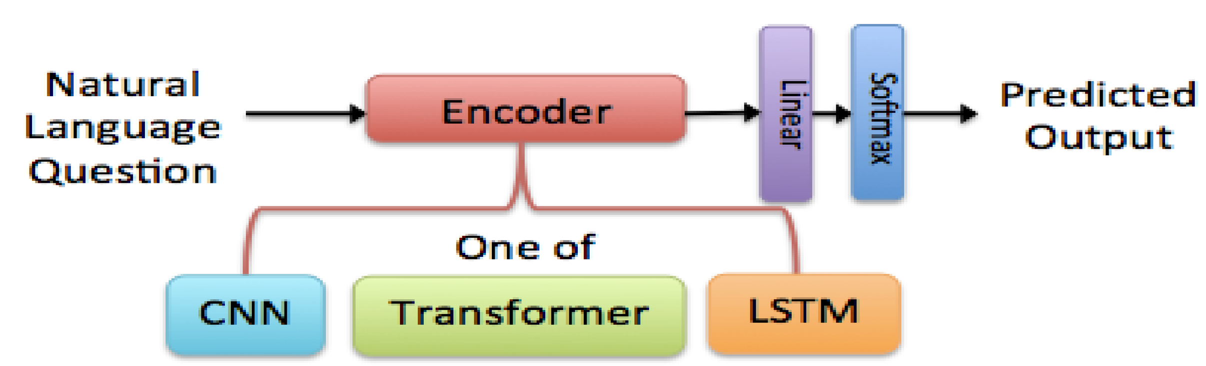
Baseline and different word embeddings:

- At first we view this task as a language translation problem, so we implemented a bidirectional-LSTM based neural machine translation model as our baseline. It can be seen that NMT performs poor than others which modeled syntactical structure information.
- Previous commonly used embedding is GloVe, but in this project we implemented BERT embeddings and achieved better performance.

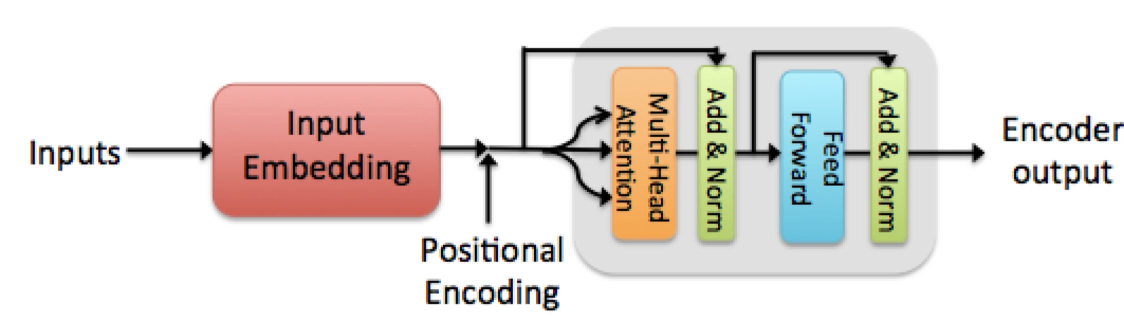
Model	Acc_{qm}	Acc_{ex}
NMT (GloVe)	29.9%	–
Seq2SQL (GloVe)	49.8%	57.9%
SQLNet (GloVe)	58.4%	65.3%
SQLNet (BERT)	61.4%	67.8%
TypeSQL(GloVe)	68%	74.5 %

Different encoders for aggregator:

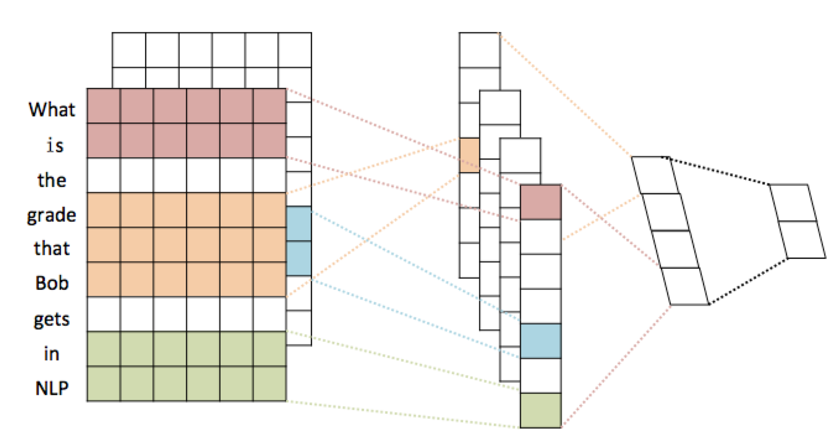
We break the translation task into 3 subnetworks as explained in Strategy session. For encoder we explored 3 deep neural networks as follows.



- Transformer based encoder architecture:



- CNN based encoder architecture:



Results

Our results show that transformer and CNN performed similarly on this task on dev set and test set, and give comparable results with the original LSTM encoder. However, transformer performs really poor on selection clause prediction. It might be due to limited time of training.

Model	train accuracy	dev accuracy	test accuracy
LSTM encoder	99.2%	89.2 %	90.0%
Transformer encoder	86.9%	86.6%	86.6%
CNN encoder	99.1%	86.4%	85.3%

Model	dev accuracy	test accuracy
LSTM encoder	89.6%	88.5%
Transformer encoder	19.6%	18.7%

Error Analysis

Columns in table: "Year", "Álbum", "Charts", "Sales", "Certification"
Question: "How many tries against were there with 17 losses?"
Ground truth aggregator: No aggregator
Transformer encoder prediction: COUNT

Question: "What were the number of sales before 1991?"
Ground truth aggregator: No aggregator
Transformer encoder prediction: COUNT

The Aggregator predictor may lack understanding of column names that already have the aggregated results.

Conclusion

- SQL syntax info helps neural network understand "text-to-SQL" task
- BERT embedding performs better than GloVe on existing models
- Transformer and CNN perform comparably with LSTM on $AGG prediction
- Transformer fails to generate reasonable $SELECT_COL prediction temporarily