**A Comprehensive Exploration on WikiSQL with Table-Aware Word Contextualization**

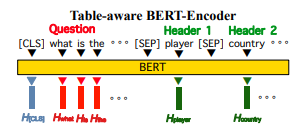
**Abstract:**

Given a table from Wikipedia article the task of WikiSQL is to map a natural language question to a SQL query. 3 BERT based architecture was explored and the best model outperformed in execution accuracy. The model also had better accuracy than human.

**Introduction:**

Semantic parsing is the task of translating natural language utterances to (often machine executable) formal meaning representations. There are three layer models used SHALLOW-LAYER, DECODERLAYER, and NL2SQL-LAYER. Word contextualization is crucial for language tasks with structured data. Models effectively achieve the upper bound of the accuracy on WikiSQL task.

**Table-aware BERT Encoder:**



[SEP] is used to separate between the query and the headers. Each query input Tn,1 . . . Tn,L (L is the number of query words) is encoded as following:

**[CLS], Tn,1, · · · Tn,L, [SEP], Th1,1, Th1,2, · · · , [SEP], · · · , [SEP], ThNh ,1, · · · , ThNh ,MNh ,[SEP]**

This input scheme is used in SHALLOWLAYER and NL2SQL-LAYER. For DECODER-LAYER, additional SQL vocabulary such as select, where, min, and > and start and end tokens are placed between [CLS] and question words separated by [SEP] for the sequence generation. By placing them in front of question- and header tokens, their positions remain invariant to questions and tables headers. The output from final Transformer block (Vaswani et al., 2017) are used in SHALLOW-LAYER and DECODER-LAYER whereas the output of final two Transformer blocks are concatenated and used in NL2SQL-LAYER.

**SHALLOW-LAYER:**

SHALLOW-LAYER uses syntax guided sketch, where the generation model consists of six modules, namely select column, select-aggregation, where-number, where-column, where-operator, and where-value.

**DECODER-LAYER:**

It contains LSTM decoders. It also generates first token of each header and interpret them as entire header tokens during inference stage using Point-to-SQL module. The model generates only the pointers to start and end- where-value tokens omitting intermediate points.

**NL2SQL-LAYER (SQLOVA):**

It contains both encoders and decoders treating the output of table-aware BERT encoder as word-embedding vectors. It also generates SQL query using six separate modules.

**Experiment:**

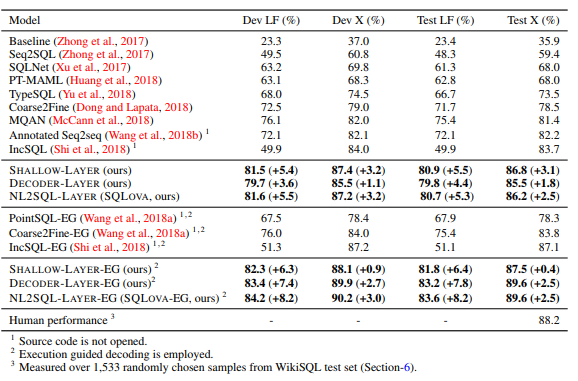
The configuration is as follows:

* Pre trained BERT model
* Fine tuned with ADAM optimizer with learning rate 10−5
* Sequence-to-SQL module of NL2SQL-LAYER(SQLOVA)
* the decoder in DECODER-LAYER are trained with 10−3 learning rate with β1 = 0.9, β2 = 0.999
* Batch size is 32

natural language utterance is first tokenized by using Standford CoreNLP. Each token is further tokenized (into subword level) by WordPiece tokenizer. The PyTorch version of BERT code is used for word embedding and part of the code in NL2SQL-LAYER is influenced by the original SQLNet source code. All computations were done on NAVER Smart Machine Learning (NSML) platform.

**Accuracy:**

The execution accuracy is measured by evaluating on the answer returned by ‘executing’ the query on the SQL database. Below table illustrates:



**Summary:**

The effectiveness of table-aware word contextualization on a popular semantic parsing task, WikiSQL was demonstrated. BERT-based table-aware encoder and three task-specific modules with different model complexity on top of the encoder, namely SHALLOW-LAYER, DECODERLAYER, and NL2SQL-LAYER was proposed. The simplest module, SHALLOW-LAYER, can outperform the previous best model, but a sufficiently dense module, NL2SQL-LAYER, gives the best result across several different settings.