1 Abstract

The recently-released Spider task was the first large-scale, complex, cross-domain dataset for semantic parsing in the form of text-to-SQL question answering. I experiment with a baseline SQLNet model applied to Spider. While the column attention mechanism was a crucial feature of SQLNet when it achieved state-of-the-art performance on the WikiSQL task, boosting it by an additional 3 points, no similar improvement is gained on the Spider task. The attention mechanism is helpful at the meta-level insofar as intuitive attention is learned and can help visualize the model. However, the failure of the attention mechanism on bottom line results suggests future areas of improvement for the model.

2 Introduction

The problem of translating natural language to a computer-understandable encoding has been studied for a long time. This need has increased recently due to the exponential explosion of recorded data in structured databases query-able by SQL. At the same time, few people have adequate understanding of SQL to
leverage this data; even the ones who do find it tedious to express their query in such a rigid syntax.

There has been much work towards automating the “sequence to SQL” translation process, including the generation of datasets such as ATIS [1] [2], GeoQuery [3], and WikiSQL [4]. Each of these provided valuable datasets and models; however, the queries were not nearly complex enough to extract powerful information from databases. Further problems include that the datasets have evaluation unfairness, such as randomly splitting highly-redundant examples into train and test.

3 Problem Description

The Spider dataset [5] strove towards complex queries. Its complexity is represented in transcending simple “SELECT ... WHERE,” queries; instead using “GROUP BY,” nested, “HAVING” and “JOIN”-ing over many tables. See Figure 1 for the syntax-level contributions of Spider. An example complex question is shown in Figure 2.

Additionally, unlike WikiSQL, Spider is cross-domain in the sense that different databases are about a wide range of different topics: singers’ perfor-
### Complex question
What are the name and budget of the departments with average instructor salary greater than the overall average?

### Complex SQL
```
SELECT T2.name, T2.budget
FROM instructor as T1 JOIN department as T2
ON T1.department_id = T2.id
GROUP BY T1.department_id
HAVING avg(T1.salary) >
(SELECT avg(salary) FROM instructor)
```

Figure 2: Example complex query in Spider dataset

These databases are split between train and test (rather than simply splitting examples on the same database), forcing the model to generalize in a much more profound way: the models must truly understand the semantic content of the question as it relates to the given dataset, rather than simply memorizing highly-redundant patterns between train and test.

## 4 Approach

Several models have been proposed for addressing this new semantic parsing task. One of the most promising baselines is adapted from SQLNet, a state-of-the-art model for the WikiSQL task. This model was initially used for WikiSQL’s more simply-structured queries, but it has been extended to capture the greater complexity in the Spider task.

My contribution is to experiment with the adapted SQLNet model and expose its effectiveness on Spider, while also suggesting improvements.

SQLNet employs a sketch approach that corresponds to the SQL grammar and treats SQL generation as a slot-filling exercise within this sketch. This improves over other models in that it fundamentally avoids sequence-to-sequence structure when order does not matter, such as each column to be conditioned...
Figure 3: SQLNet Sketch-Based Model

over in a multiple “WHERE” clause (see Figure 3).

One of the major insights of the SQLNet model was the use of column attention. Rather than directly proceeding with the question and column encoding, the question is re-encoded conditional on each column. Intuitively, if the model must decide whether to fill a certain column into a slot, it is difficult to directly use the question encoding, which might not remember important word-level information related to the correct column. Instead, with column attention, the model can upweight words in the question most relevant to classification tasks specific to that column.

Formally, without the attention, the model for predicting a column in the WHERE component is:

\[ P_{\text{where}col}(\text{col}|Q) = \sigma(u_c^T E_{col} + u_q^T E_Q), \]

where \( E_{col} \) are the last layer of the Bi-LSTM column encodings, and \( E_Q \) are the same for the question tokens. The \( u \) are learnable matrices, and the sigmoid function is used to scale the real-number output to the zero-to-one interval. With attention, a matrix \( w \) is learned based on the column and question encodings to represent how to up-weight certain of the token encodings. With
attention, we have

\[ P_{\text{where} \mid \text{col}}(Q) = \sigma(u_c^T E_{\text{col}} + u_q^T E_{Q \mid \text{col}}), \]

where now,

\[ E_{Q \mid \text{col}} = H_Q w \]

is the attention-upweighted transformation of the question Bi-LSTM highest layer.

5 Data Sets Used

The Spider task, presented by [5] and described in Section 3, is publicly available at https://yale-lily.github.io/spider.

6 Evaluation Method

The original Spider task allows for evaluation of the model by metrics such as Component Matching, Exact Matching, Execution Accuracy, and accuracy as a function of query difficulty. We focus on the first two of these. For more details, refer to [5].

6.1 Component Matching

In keeping with the Spider task, component matching is tested over the following SQL components:

1. SELECT
2. WHERE
3. GROUP BY
4. ORDER BY

Exact match within each component is tested against the gold SQL. To represent the “order doesn’t matter” nature of a select component such as “SELECT avg(col1), min(col1), max(col2),” the F1 score to incentivize precision and recall over elements in a set is used. For instance, the mutated component “SELECT min(col1), max(col2), avg(col1),” would achieve the full score. An average F1 score per component over all examples is taken.

6.2 Exact Matching

Exact matching is equivalent to all components fully matching in the order-doesn’t-matter sense described in the last section. That is, the predicted query is correct if and only if all components are correct.

7 Results

My experiments centered on the problem of understanding the column attention mechanism, specifically when applied to the Spider task. First, The SQLNet paper reports a 3 point improvement when they use column attention; I find no similar improvement in the adapted SQLNet model applied to Spider (see Table 1). Second, although the SQLNet paper emphasizes “column attention is a special instance of the generic attention mechanism,” they make no attempt to visualize this attention as has become popular in other settings where attention has been used. Thus, I sought to visualize the attention mechanism, shown in Figure 4.

Although somewhat intuitable, we notice that the attention still seems to be slightly biased towards the start and end of the sentence (see Figure 5). This implies that the encoding of the bi-LSTM at the start and end might be-
Figure 4: Column Attention for Question about Singers

Figure 5: Column Attention for Question about Continents
Table 1: Model Results. We contrast the model with column attention against one with attention removed.

<table>
<thead>
<tr>
<th>Attention?</th>
<th>Select</th>
<th>Where</th>
<th>Group By</th>
<th>Order By</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>30.2</td>
<td>54.6</td>
<td>73.2</td>
<td>85.5</td>
<td>10.4</td>
</tr>
<tr>
<td>No</td>
<td>30.3</td>
<td>51.5</td>
<td>72.0</td>
<td>86.0</td>
<td>10.2</td>
</tr>
</tbody>
</table>

and-large be sufficient for the task. If this is true, the parameter space can be decreased without the attention mechanism, as can the run time (46 minutes down from 53 in my experiment). The model can be enriched elsewhere without overfitting.

8 Conclusion

Although the SQLNet model has been reapplied with some success to the Spider task, there is clearly room for improvement. Towards this end, I sought to experiment with what gain the column attention is giving the new model on the Spider task. On the Spider task there is hardly an improvement for using attention. Attention lends interpretability to the model through the attention plot, but suggests enriching the model elsewhere could yield better results. I propose a future direction of self-attention when learning the column embeddings, since fundamentally order of columns in the table does not matter. Furthermore, in [6], the authors find that attention can improve over bi-LSTM, which is currently used to represent column meaning.

9 Appendix: Code and Data

References


