TED-Learn: Towards Technology-Enabled Database Education

Sourav S Bhowmick
School of Computer Science & Engg
Nanyang Technological Univ
Singapore
Query Optimization in RDBMS

```
select o_orderpriority, count(*) as order_count from orders where o_orderdate >= date '1996-03-01' and o_orderdate < date '1996-03-01' + interval '3' month and exists (select * from lineitem where l_orderkey = o_orderkey and l_commitdate < l_receiptdate) group by o_orderpriority order by o_orderpriority limit 1;
```
Circa 2017

**Student:** I’ve been trying to understand the QEPs in X but it’s really hard to follow….the descriptions are very different from what we learn from textbooks and lecture slides 😞

**Me:** Yeah! DBMS vendors use vendor-specific implementation and language. You may refer to their manuals for details...

**Student:** It’s boring to read manuals! In any case, for different DBMS I have to peruse different manuals…it’s such a waste of time!

**Me:** Well, existing RDBMS are designed primarily for enterprises and not for students.... 😊

**Student:** Can you please make things easy for us to learn? I am really struggling to understand query plans....
Hit Like A Brick Through the Window
Database Research for Enterprises

Gartner DBMS Market Share Ranks: 2011-2020

Source: Market Share: All Software Markets, Worldwide, 2020, 14 April 2021 (G00743607)
© 2021 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates.
Key Question

Can we describe a QEP using natural language to enhance DB education?
Juxtapose Ideas That Often Don’t Go Together

https://towardsdatascience.com/text-to-sql-learning-to-query-tables-with-natural-language-7d714e60a70d
Juxtapose Ideas That Often Don’t Go Together

```
select o_orderpriority, count(*) as order_count from orders where o_orderdate >= date '1996-03-01' and o_orderdate < date '1996-03-01' + interval '3' month and exists (select * from lineitem where l_orderkey = o_orderkey and l_commitdate < l_receiptdate) group by o_orderpriority order by o_orderpriority limit 1;
```

The query is executed as follow.
Step 1, perform sequential scan on table orders and filtering on (o_orderdate >= '1996-03-01':date) AND (o_orderdate < '1996-06-01 00:00:00':timestamp without time zone) to get intermediate table T1.
Step 2, perform sequential scan on table lineitem and filtering on l_commitdate < l_receiptdate to get intermediate table T2.
Step 3, hash table T2 and perform hash join on table T1 and table T2 under condition orders.o_orderkey = lineitem.l_orderkey to get intermediate table T3.
Step 4, sort T3 and perform aggregate on table T3 with grouping on attribute orders.o_orderpriority to get intermediate table T4.
Step 5, limit the result from table T4 to 1 record(s) to get the final result.
Overview of NEURON [SIGMOD 2019]

NEURON

- Rule-based natural language description
- Build on top of PostgreSQL

```
select o_orderpriority, count(*) as order_count
from orders
where
  o_totalprice > 100
and exists (
    select *
    from lineitem
    where
      l_orderkey = o_orderkey
      and l_extendedprice > 100
  )
group by o_orderpriority
order by o_orderpriority;
```

**Rule Template**

hash table <T> and perform hash semi join on table tablenametable <T> under condition (<C>) to get intermediate table <TN>.
NEURON v1.0

https://neuron.scse.ntu.edu.sg/#/
User Feedback: How easy is it to understand the query plan presented in various formats?

- 62 students volunteered for the survey (Oct 2019)
- NEURON is the easiest format (74.2%) to understand.
- Visualized plan (71%) is comparable with NEURON (74.2%).
- Majority (62.9%) of respondents found JSON format difficult to understand.
User Feedback: Which query plan format is most preferred?

- The survey participants preferred NEURON the most (53.2%)
- Very few participants (3%) chose JSON as the most preferred choice.
User Feedback: How easy is it to understand the query plan presented in various formats?

- 56 students volunteered for the survey (Oct 2020)
- NEURON is the easiest format (91.1%) to understand.
- Majority (66.1%) of respondents found JSON format difficult to understand.
- In term of ease of understanding (ranking of 4 or 5):
  NEURON (91.1%) > Visualized plan (66.1%) > JSON (17.9%)
User Feedback: Which query plan format is most preferred?

- The survey participants preferred NEURON the most (55.4%)
- Very few participants (5.4%) chose JSON as the most preferred choice.
User Feedback: How likely are you to recommend NEURON to a course mate?

- 89.3% of respondents are quite likely to recommend NEURON to a course mate.
User Feedback

“I use SQL Server. I can’t use your tool! It only works on PostgreSQL!”

“After running few queries and reading the descriptions, I feel bored and skip sentences as the language is repetitive!”

“The natural language translation can be improved to summarize complex conditions of the QEP.”
## Issues

<table>
<thead>
<tr>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>• How do we generalize NEURON across different RDBMS?</td>
</tr>
<tr>
<td>• How can we alleviate boredom?</td>
</tr>
</tbody>
</table>

## Challenges

<table>
<thead>
<tr>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Different RDBMS have physical operators with different names</td>
</tr>
<tr>
<td>• Rule-based algorithms naturally generates similar text descriptions.</td>
</tr>
</tbody>
</table>
## Key Idea

- Instead of mapping an entire QEP to its NL description, map the set of physical operators in a RDBMS to corresponding NL descriptions.
- Stitch them together to generate the description of a specific QEP.
- More manageable to label physical operators.
- Enables generalization to handle any application-specific database.
- Orthogonal to the complexities of SQL queries.

## Two Variants

- RULE-LANTERN
- NEURAL-LANTERN
Labeling Physical Operators

A declarative framework where a subject matter expert (SME) can create and manipulate the labels using a declarative language called POOL (Physical Operator Object Language).

POEM Data Model

• A simple and flexible graph model where all entities are objects.
• Each object -> a physical operator of a relational query engine.
• Objects are either atomic or complex having attribute-value pairs.
• source, name, alias, defn, desc, type, cond, and target.
• Values of all attributes are from the atomic type string.
CREATE POPERATOR hashjoin FOR pg (ALIAS = null, TYPE = 'binary', DEFN = null, DESC = 'perform hash join', COND = 'true', TARGET = null)

ComposeOperator

- Specify generation of an NL description template of an operator.
- Uses the desc, type, and cond attributes of operators to generate.

POperate$(oid, source, name, alias, type, defn, cond, targetid)$
PDef$(oid, desc)$

COMPOSE hash FROM pg

“hash $R1$”

COMPOSE hash, hashjoin FROM pg USING hashjoin.desc = 'perform hash join'

“hash $R1$ and perform hash join on $R2$ and $R1$ on condition $cond$”.
```
select o_orderpriority, count(*) as order_count
from orders
where
  o_totalprice > 100
and exists (  
    select *
    from lineitem
    where
      l_orderkey = o_orderkey
      and l_extendedprice > 100
  )
group by o_orderpriority
order by o_orderpriority;
```
User Feedback (Revisited)

“After running few queries and reading the descriptions, I feel bored and skip sentences as the language is repetitive!”
### NEURAL-LANTERN

#### Leverage DL
- Regard a QEP as an input language while the NL description as the output.
- Interpreting QEP into NL can be viewed as a machine translation task.

#### Challenges
- DL-based techniques need massive training sets of labeled examples to learn from.
- Prohibitively expensive as they demand database experts to translate thousands of QEPs.
- The platform needs to be *generalizable* and application domain-independent for ease of deployment and usage.
## NEURAL-LANTERN: Training Data

### Training Data Generation

- We adopt Kipf et al. [CIDR 2019] to generate a set of SQL queries given a particular schema and database instance.
- A collection of QEPs corresponding to these queries.
- Each QEP is decomposed into a set of acts, each of which corresponds to a set of operators in an operator tree (subtree).
- For each act-> RULE-LANTERN to generate NL description.

### Diversifying Text

- For each RULE-LANTERN result, we apply three state-of-the-art synonymous sentence generation tools and acquire a set of synonymous sentences.
- Remove duplicates and manually eliminate invalid sentences.
Diversifying Translation:
1. hash table \(<T>\) and hash semi enter under condition \(<C>\) on table tablename and table \(<T>\) to obtain intermediate table \(<TN>\).
2. hash table \(<T>\) and do a half hash join on table tablename and table \(<T>\) under condition \(<C>\) to get intermediate table \(<TN>\).
3. hash table \(<T>\) and perform hash semi join on table tablename and table \(<T>\) under condition \(<C>\) to get transitional table \(<TN>\).
QEP2Seq Model

- Follows the Seq2Seq structure.
- The acts collection $actCol$ is composed of a series of acts $L_1, L_2, \ldots, L_n$, each of which is derived from the QEP.
- The Encoder RNN encodes each word in $L_i$ into the corresponding hidden state $h_t$ using an LSTM layer.
- We use an LSTM decoder with an attention mechanism to let the decoder focus on the relevant portion of the encoder while generating a token.
- We adopt both static (Word2Vec and GloVe) and contextual word embeddings (ELMo and BERT) in decoder.
- Training data: workloads in TPC-H (22 queries) and SDSS (71 queries)
- Apply trained model on IMDB (1000 SQL queries)
- Pre-trained word embeddings can reduce the validation set loss while improving validation set accuracy and alleviate overfitting problem.
User Feedback: Which query plan format is most preferred?

- JSON: 30.23%
- Visual tree: 27.91%
- RULE-LANTERN: 11.63%
- NEURAL-LANTERN: 30.23%

- 43 students volunteered for the survey (2020)
User Feedback: How easy is it to understand the query plan presented in various formats?

![Bar Chart]

- Not easy at all (1)
  - JSON: 14
  - Visual tree: 10
  - RULE-LANTERN: 10
  - NEURAL-LANTERN: 10
- Somewhat easy (2)
  - JSON: 6
  - Visual tree: 6
  - RULE-LANTERN: 6
  - NEURAL-LANTERN: 6
- Easy (3)
  - JSON: 2
  - Visual tree: 2
  - RULE-LANTERN: 2
  - NEURAL-LANTERN: 2
- Extremely easy (4)
  - JSON: 0
  - Visual tree: 0
  - RULE-LANTERN: 0
  - NEURAL-LANTERN: 0

Total responses: 42
Impact of Boredom

Do learners feel bored?

- We presented a set of output generated by each approach in random order.
- Asked the subjects to rate the degree of boredom (boredom index) they felt perusing these plans to understand QEPs using the Likert scale of 1-5.

<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE-LANTERN</td>
<td>2</td>
<td>7</td>
<td>19</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>NEURAL-LANTERN</td>
<td>6</td>
<td>11</td>
<td>22</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>NEURON</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>LANTERN</td>
<td>6</td>
<td>12</td>
<td>21</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
**NEURON vs LANTERN**

**Integrate RULE-LANTERN and NEURAL-LANTERN**

- Track \((QEP, \text{NL description})\) pairs viewed by each participant.
- By default, the NL description of each physical operator is generated using RULE-LANTERN.
- Whenever an operator appeared more than a pre-defined frequency threshold (i.e., 5) in total in different QEPs associated with a participant, NEURAL-LANTERN is invoked to generate the description for the operator.
User Feedback: LANTERN vs NEURON

![Bar chart showing user feedback comparison between LANTERN and NEURON. The x-axis represents the number of responses, with categories ranging from 'Not good at all (1)' to 'Extremely well (5)'. The y-axis indicates the number of responses. The chart illustrates a higher number of positive responses for NEURON compared to LANTERN.](chart.png)
Query Optimization + Natural Language Generation = DB Education
### Summary

<table>
<thead>
<tr>
<th>Simple and Communicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Generating natural language descriptions of QEPs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timely and Fundamental</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Lifelong learning is increasingly becoming a reality</td>
</tr>
<tr>
<td>• By fundamental we mean that the idea touches on something basic to humans in collective settings</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Personally Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Education is personal!</td>
</tr>
</tbody>
</table>
Interesting Issues based on Feedbacks

- How do we summarize the descriptions?
- How can learners explore the impact of physical operators on a QEP?
- How can learners converse with a query optimizer to aide learning?
DBMS Needs to Break Out from the Enterprise Jar

How do we facilitate understanding of SQL?

How do we let learners converse with the query optimizer?

Not limited to relational DBMS
Acknowledgements

Hui Li, Xidian Univ, China
Shafiq Joty, NTU
Patricia Chen, Dept of Psychology, NUS

- Siyuan Liu, NTU
- Weiguo Wang, Xidian
- Peng Chen, Xidian
- Zheng Li, Xidian

- Student volunteers in NTU and Xidian
Thank You!