**Natural Language Querying of Complex Business Intelligence Queries**

Jaydeep Sen*, Fatma Özcan*, Abdul Quamar†, Greg Stager†, Ashish Mittal*,
Manasa Jammī*, Chuan Lei†, Diptikalan Saha*, Karthik Sankaranaranayanan*

*IBM Research AI, †IBM Canada

**Motivation**
- Business Intelligence (BI) queries provide invaluable insights in the enterprise
- NL interfaces enable BI querying for business users, who are not SQL experts, beyond fixed reports
- Existing NLDB systems fail to handle complex nested SQL queries needed by BI in the enterprise

**Overview**
- **Extension of our earlier system, ATHENA**: A state-of-the-art Ontology Based NLDB system
- **Ontology** is used to capture the deep domain semantics needed to model the target domain
- Heuristics to detect and guide subquery formations by combining the use of intelligent lexicon analyzers together with deep domain reasoning over the ontology
- Generic and domain agnostic system and algorithms, capable of generating complex SQL queries involving selections, aggregations, as well as nesting
- Rule-based interpretation, no need for training data
- High accuracy in preliminary results, proving the effectiveness of using a combination of lexical analyzer and deep domain reasoning

**FIBEN: Finance Domain Benchmark Dataset**
- Emulates real world data mart for a financial application
- Combines SEC data with transactional TPoX data

**SEC Data**
- Provides information about public companies, their officers and financial metrics
- Dataset extracted from the public SEC filings submitted as XBRL documents
- Data curated by running named entity extraction, and entity resolution by IBM Research

**TPoX Data**
- Transaction Processing benchmark for financial applications.
- Data generator allows scaling

Data transformed to conform to standard finance ontologies:
- FIBO (Finance Industry Business Ontology)
- FRO (Finance Report Ontology)

**System Architecture**
- NLQ Engine
- Translational Index
- Domain Ontology
- Ontology to Database Mapping
- Query Translator
- Nested Query Handler
- Evidence Annotators
- Nested Query detector
- Subquery Formation
- Subquery Join Condition
- Query Building
- Results

**Example Walkthrough**
- Show me everyone who bought stocks in 2016 that has gone up in value
- Evidence Set: #1(ES1)
- Joining Condition = (((value.value > 0) and (current.value > 0)) and (value.current.value > 0))
- Shared Evidences = ([stock.value], [value.value])

**Preliminary Results**

<table>
<thead>
<tr>
<th></th>
<th>Overall Accuracy</th>
<th>FIBEN</th>
<th>ATHENA</th>
<th>NALIR</th>
<th>DBPal</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLNest</td>
<td></td>
<td>92.78</td>
<td>65.35</td>
<td>28.86</td>
<td>41.75</td>
</tr>
<tr>
<td>FIBEN</td>
<td></td>
<td>79.71</td>
<td>0.0</td>
<td>10.14</td>
<td>21.73</td>
</tr>
</tbody>
</table>

**References**
2. Shreyas Bharadwaj, et al., "Creation and Interaction with Large-scale Domain-Specific Knowledge Bases", in PVLDB 10(12)
3. FIBO. https://spec.edmcouncil.org/fibo/