Research Paper Link: https://arxiv.org/pdf/2402.08349

Dataset Creation and Setup: FootballDB

- Data Collection: FootballDB contains comprehensive information about FIFA World Cup matches, teams, players, and clubs from 1930 to 2022. This data was enriched by scraping and integrating multiple sources like DBpedia and Wikidata.
- Schema Design: The researchers structured FootballDB with three different data models (v1, v2, v3) to see how design changes impact SQL generation.
- For example:
  1. v1 was a standard relational schema.
  2. v2 introduced additional tables to address issues with foreign key constraints and improve query generation.
  3. v3 simplified joins and relationships by adding intuitive columns, reducing query complexity.

Deployment and Real-World User Interaction

- Live Deployment: The system was deployed publicly during the FIFA World Cup 2022, attracting around 6,000 natural language queries from real users. Users could ask football-related questions, which were then translated into SQL by the system.
- User Feedback: To refine the system, user feedback was integrated through mechanisms like thumbs-up/down on query results, SQL query correction, and monitoring common user questions to update the dataset accordingly.

Experimental Design: System and Model Evaluation

- **Design Dimensions:** The study evaluated each Text-to-SQL system by analyzing four main design dimensions:
- **Data Model:** Testing the performance across v1, v2, and v3 to see which schema design facilitated better SQL generation.
- **Language Model:** Evaluating different models, including small models (ValueNet with BART), medium models (T5), and large language models (GPT-3.5 and LLaMA2). Training Data Size: Testing the effect of different training set sizes (100, 200, and 300 samples) to assess how much labeled data is needed for meaningful improvements.

- **Pre- and Post-Processing:** Analyzing various pre-processing (e.g., schema linking) and post-processing (e.g., SQL generation constraints) techniques to see how they impacted the system's performance.

Evaluation Metrics and Testing

- **Execution Matching (EX):** Since exact SQL matching was not feasible due to parser limitations, they used *execution matching* — testing if the predicted SQL query returned the correct data.
- **Query Complexity Analysis:** Queries were analyzed based on characteristics like the number of joins, filters, set operations, and projections. The complexity level was categorized using a modified Spider benchmark metric (easy, medium, hard, extra hard).
- **Performance Across Query Characteristics:** Execution accuracy was evaluated against various characteristics, allowing the team to identify which types of queries (e.g., those with many joins or filters) posed the most challenges.

Results and Observations

- Comparison by Model Size: GPT-3.5 (175 billion parameters) performed better than other models, showing the advantage of larger models, but with significant inference costs.
- Impact of Data Model Optimization: The v3 data model, with fewer joins and more intuitive structures, enabled higher accuracy across all systems, highlighting that simpler schemas can help reduce system errors.
- Inference Time and Hardware Costs: The study documented that medium-sized models like T5-Picard had long inference times (over 5 minutes per query), indicating that practical deployments require improvements or optimizations.