MapQA: Geo-spatial Question Answering from OpenStreetMap

Mihir Kulkarni, Supervised by Dr. Muhao Chen and Ehsan Qasemi

If interested contact Mihir Kulkarni, msk5743@psu.edu

Work performed under REU Site program supported by NSF grant #2051101

USC Viterbi School of Engineering
Information Sciences Institute

Problem Statement

- Harnessing geo-spatial data has been a challenging task
- The demand for question-answer systems using geo-spatial data, like OpenStreetMap, is rapidly increasing
- These systems have immense potential in diverse arenas, including urban planning and emergency aid

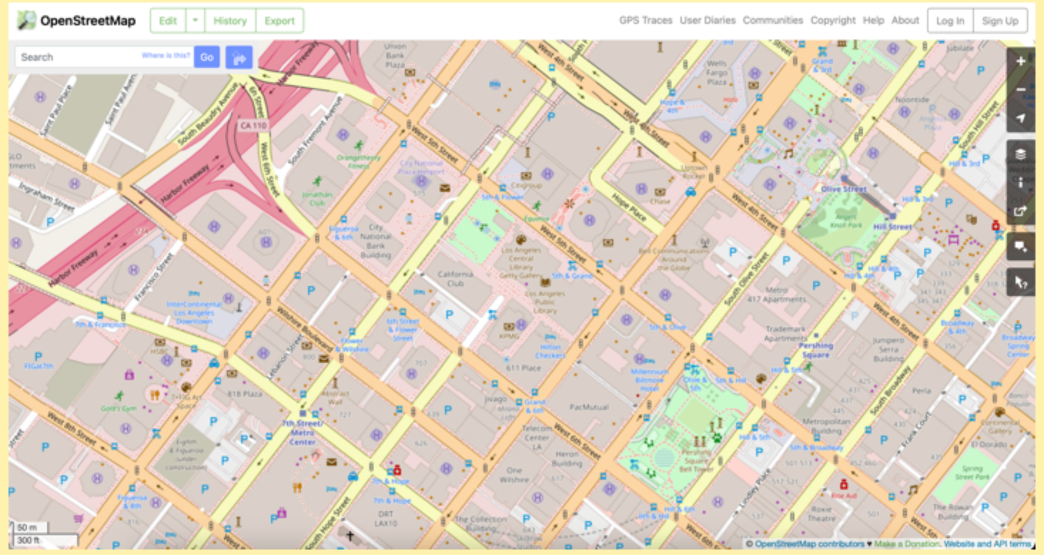

Fig 1: OpenStreetMap interface

Approach

- **Aim:** Using a comprehensive dataset developed through advanced SQL queries on a PostgreSQL server, we aim to gain a better understanding of geo-spatial question-answer systems
- For instance, a user might ask, "How many museums are there in Los Angeles?"
- This query can be retrieved using an SQL query:

```
SELECT COUNT( DISTINCT museums.*)
FROM point AS museums
JOIN multipolygons AS city
WHERE
 ST_CONTAINS(city.wkb_geometry,
museums.wkb_geometry)
 AND city.name = "Los Angeles"
 AND museums.ameneties = "museums";
```

- **Dataset Generation:**
  - Begin with a robust question categorization, segregating them into 8 clear categories
  - Sampled and annotated 30 questions
  - Based on comprehensive literature review
  - Utilize expert designed SQL queries to extract knowledge from OpenStreetMap
  - Use Python to generate question-answer pairs using weak supervision

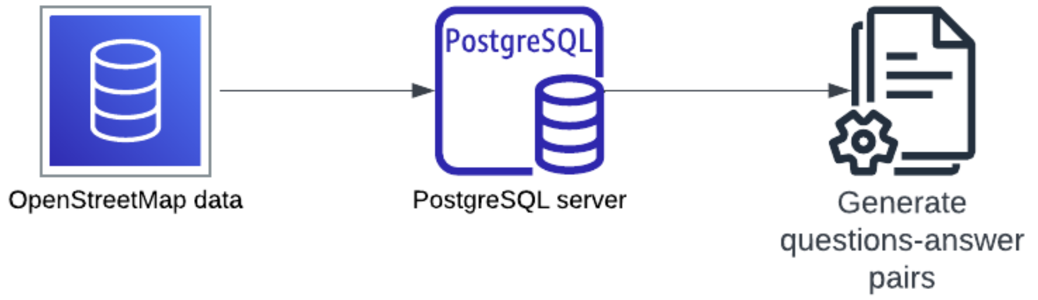

Fig 2: Generate Dataset

- **Benchmarking:** We focused on two classes of QA models:
  - Text2SQL: GPT-4 as our Text2SQL baseline, gauging its strengths and limitations in handling spatial queries
  - Retrieval models: Test DPR, GeoLM and SpaBERT, our spatially-aware models, against this baseline

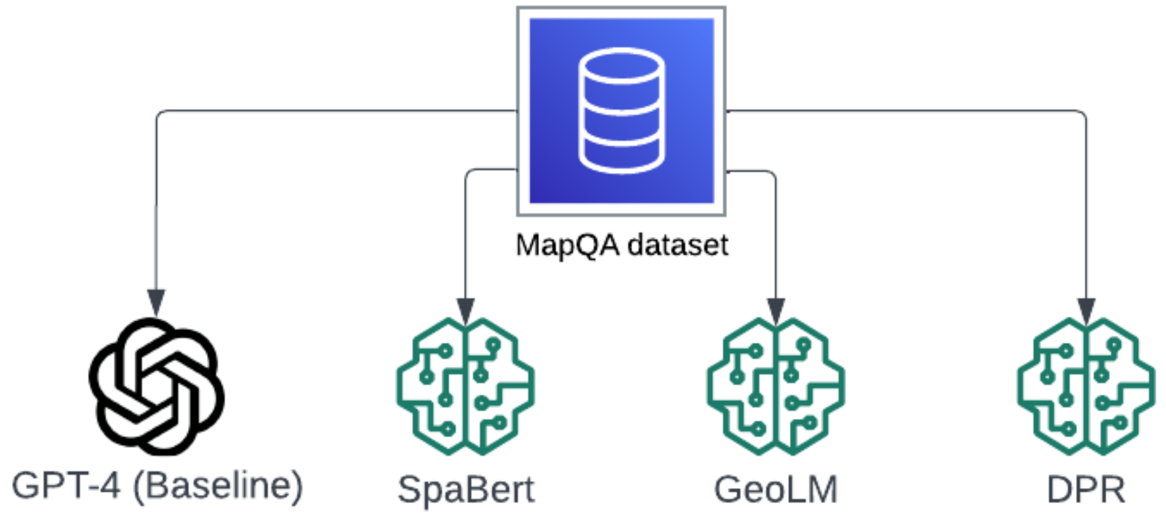

Fig 3: Model benchmarking

- **Quality Control:**
  - Analyzed generated datasets for consistency
  - Sampled 40 questions and answers with 100% accuracy
- **Evaluation:**
  - Assess errors with baseline model's performance
  - Classes of errors of Text2SQL QA outputs: failure to answer, incorrect answer and correct answer
  - Utilize accuracy scores as the primary metric, keeping the three distinct outcomes in consideration.
- **Optimization:**
  - Tune baseline LM prompts to provide complete context for the dataset
  - Based on initial test results, engage in systematic hyper-parameter tuning to refine model accuracy
  - Conduct thorough error analysis to understand model shortcomings and iterate improvements

Future Work

- Exploring other models and architecture tweaks to improve performance of the baseline
- Evaluating the accuracy of the retrieval-based models
- Investigating the feasibility of real-time deployment in practical applications