

Text to SQL Generation

Jake Waffle

2023

Swagger UI

localhost:8081/swagger-ui/#/test/getMovies

test

GET /movies get movies

Parameters

Name	Description
query <small>★ required</small> string <i>(query)</i>	The input query to be used for movie retrieval. <input type="text" value="Movies starring Barbara Carrera"/>
limit integer <i>(query)</i>	The limit for paging. <input type="text" value="100"/>
offset integer <i>(query)</i>	The offset for paging. <input type="text" value="0"/>

Execute

Swagger UI

localhost:8081/movies?query=Movies%20starring%20Barbara%20Carrera&limit=100&offset=0

nse

Details

Response body

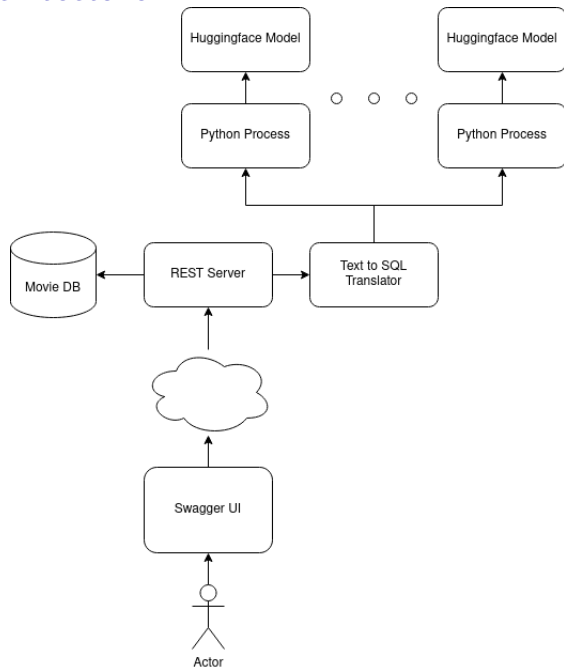
```
{
  "generatedSql": "select t1.movie_title, t1.year_released, t1.director_name, t1.producer_name, t1.studio_name, t1.color_process, t1.genre from movies as t1 join movie_casts as t2 on t1.movie_title = t2.movie_title join actors as t3 on t2.actor_name = t3.actor_name where t3.actor_name = 'Barbara Carrera'",
  "generatedSqlValid": true,
  "moviesWithActors": [
    {
      "movie": {
        "title": "Loverboy",
        "yearReleased": 1989,
        "director": "J.M.Silver",
        "producer": "Gary Foster, Willie Hunt",
        "studio": "TriStar",
        "colorProcess": "COLOR",
        "genre": "COMEDY"
      },
      "actors": [
        {
          "name": "Barbara Carrera",
          "startYearOfWork": 1978,
          "endYearOfWork": 1983,
          "gender": "female",
          "yearOfBirth": 1945
        }
      ]
    }
  ]
}
```

 Download

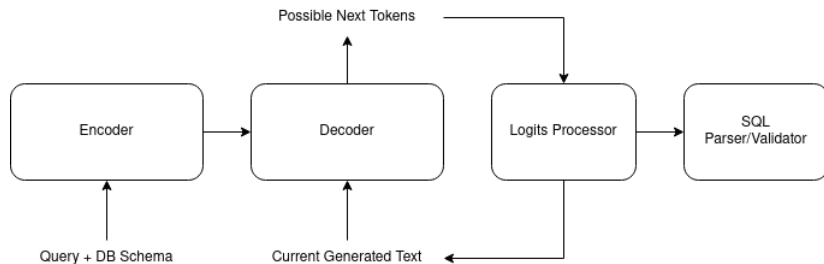
Response headers

```
content-length: 3588
content-type: application/json; charset=utf-8
x-total-count: 4
```

Architecture



Transformer



Why

- ▶ Manually supporting filters is maintenance heavy on the frontend/backend

Why

- ▶ Manually supporting filters is maintenance heavy on the frontend/backend
- ▶ Users don't like learning query languages like OData and reading documentation

Why

- ▶ Manually supporting filters is maintenance heavy on the frontend/backend
- ▶ Users don't like learning query languages like OData and reading documentation
- ▶ The model used isn't trained against a specific database schema

Pitfalls

- ▶ PEG parsers aren't well suited for incrementally validating generated text in the logits processor

Pitfalls

- ▶ PEG parsers aren't well suited for incrementally validating generated text in the logits processor
- ▶ Text generation support in libraries is generally immature

Pitfalls

- ▶ PEG parsers aren't well suited for incrementally validating generated text in the logits processor
- ▶ Text generation support in libraries is generally immature
- ▶ Python's GIL makes Huggingface's '`model.generate()`' unable to be used in multiple threads

Evaluation

- ▶ Spider text to SQL dataset

Evaluation

- ▶ Spider text to SQL dataset
- ▶ Content overlap