1. 找一个靠谱模型 跑分 一样的话 作为后端算法 （问下老师）
2. Case study 展示模型效果 展示几个例子 做个可视化
3. 前端 web
4. 模型部署后端 前后端交互
5. 租gpu 智星云 主机 端口 用户名 初始密码
6. vscode 远程资源管理器 ssh 配置

7.T5或FlanT5 先把数据集处理了 没有现成的，全自己写

C:\Users\19259\.ssh\config

Text-to-SQL任务是将自然语言文本转换为SQL查询，其主要分为自然语言理解（NLU）和SQL生成两个部分。 1. 自然语言理解（NLU）部分 在NLU部分，目标是将自然语言中提取的实体信息和查询意图映射到数据库中的表和列。通常包括以下步骤： - 分词和词性标注：将自然语言文本分解为单词序列，并标注每个单词的语法/句法角色。 - 命名实体识别（NER）：识别自然语言文本中的实体，例如地点、人名、组织、时间等，并将其标注为特定的命名实体。 - 意图识别：识别自然语言文本中的查询意图，例如选择、投影、过滤、聚合等，以及查询关键词和表。 - 语义解析：将自然语言转换为语义表示形式，如逻辑形式。 2. SQL生成部分 在SQL生成部分，生成SQL查询，输入为表示查询意图和实体的语义表达式，通常包括以下步骤： - SQL模版匹配：将SQL模版与语义表达式进行匹配，生成合法的SQL查询模版。 - 条件翻译：将WHERE子句中的查询条件翻译为SQL语句中的查询条件，包括操作符、比较符和运算符等。 - 投影列和表的翻译：将SELECT子句和FROM子句的列翻译为SQL语句中查询的列和表。 通过这两个部分的处理，最终可以将自然语言文本转换为SQL查询语句。不同的text-to-SQL模型可能有不同的实现方法和细节，但这是通常的一般性过程

Spider是一个针对SQL数据库的自然语言接口理解任务的数据集。Spider数据集分别提供了问题、SQL语句和数据库架构，旨在通过自然语言问题生成SQL查询，从自然语言接口到数据库操作之间进行关键对齐。如果要对Spider数据集进行处理，可以参考以下步骤： 1. 数据集探索：了解数据集的基本信息、特点、样本分布等内容。最好下载数据集的说明文档或参考相关论文，更好地认识数据集。 2. 数据清洗：包括去重、划分训练集、测试集和验证集以及处理缺失值等。这些步骤可以使用Python等编程语言的相关库实现。 3. 数据可视化和分析：对数据集整体和样本数据进行可视化分析，可以更好地了解各种情况下的样本分布和数据情况数据质量等。 4. 数据预处理：对问题文本和SQL语句进行分词、去停用词、繁简转换等预处理，以应对文章中的不规则问题和文本。 5. 特征处理：抽取文本的词向量、实体、关键词等特征，可以通过向量化、TF-IDF等方法处理，作为模型的输入。 6. 模型训练和评估：基于预处理后的特征、模型等设计模型，完成模型的训练和评估，可以使用常见的深度学习模型，如Transformer网络等。 7. 模型调优：根据评估结果，调整模型参数、更换模型架构，进一步提高模型性能。 希望这些步骤可以帮助你处理Spider数据集，提高自然语言接口理解的性能。`

为了解决该任务对大型高质量数据集的需求，我们创建了数据集 Spider，它包含 200 个具备多个表的数据库、10181 个问题、5693 个对应的复杂 SQL 查询。所有这些由 11 名耶鲁大学学生标注完成，共耗时 1000 小时！