DATS.6312 - Natural Language Processing
Final Project Report

Individual Report by Bharath Genji Mohana Ranga

1. Introduction

My individual contributions to this project focused on implementing BERT for label extraction and integrating this output into the T5-based SQL generation pipeline. These components enhance the model's ability to handle diverse query structures by extracting relevant SQL components from user inputs and ensuring the translation aligns with database schema requirements. This report highlights the methodologies, challenges, and outcomes of my contributions, emphasizing their role in improving the system's adaptability and accuracy.

2. Description

I contributed by designing and implementing the BERT-based multi-label classification model for SQL label extraction and integrating the output into the downstream SQL generation pipeline. The model classifies user prompts into critical SQL syntax components such as SELECT, WHERE, and GROUP BY, providing contextual guidance for query generation.

To achieve this, I fine-tuned the pre-trained BERT model to identify SQL labels in natural language prompts. This task involved preprocessing user prompts, encoding them as tokenized sequences, and training the BERT model on multi-label datasets. The extracted labels were then used in conjunction with the T5 model, which leverages this structured guidance for generating precise SQL queries.

Additionally, I developed the Label Extraction component, which plays a pivotal role in guiding the SQL generation process. By explicitly tagging SQL components, this step ensures that the model aligns the query structure with the user's intent. For example, the WHERE label helps the model focus on filtering conditions, improving accuracy for complex queries.

3. Work Area

BERT-Based Label Extraction

The BERT model processes user prompts to extract SQL components through a multi-label classification approach. Using a bidirectional transformer, BERT provides context-aware

embeddings that identify relevant SQL syntax, ensuring the generated query accurately reflects the user's requirements.

Key stages of this process include:

- **Tokenization:** Breaking down input text into tokens compatible with BERT's vocabulary.
- **Classification:** Predicting SQL syntax labels such as SELECT, WHERE, and GROUP BY using attention layers and convolutional components.
- **Integration:** Feeding the extracted labels into the T5 model for SQL generation.

T5 Integration with Labels

The labels generated by BERT were integrated into the T5 model's input format to provide additional context during SQL generation. This enriched input improves the model's understanding of the query structure, enhancing both syntactic and semantic accuracy.

4. Results

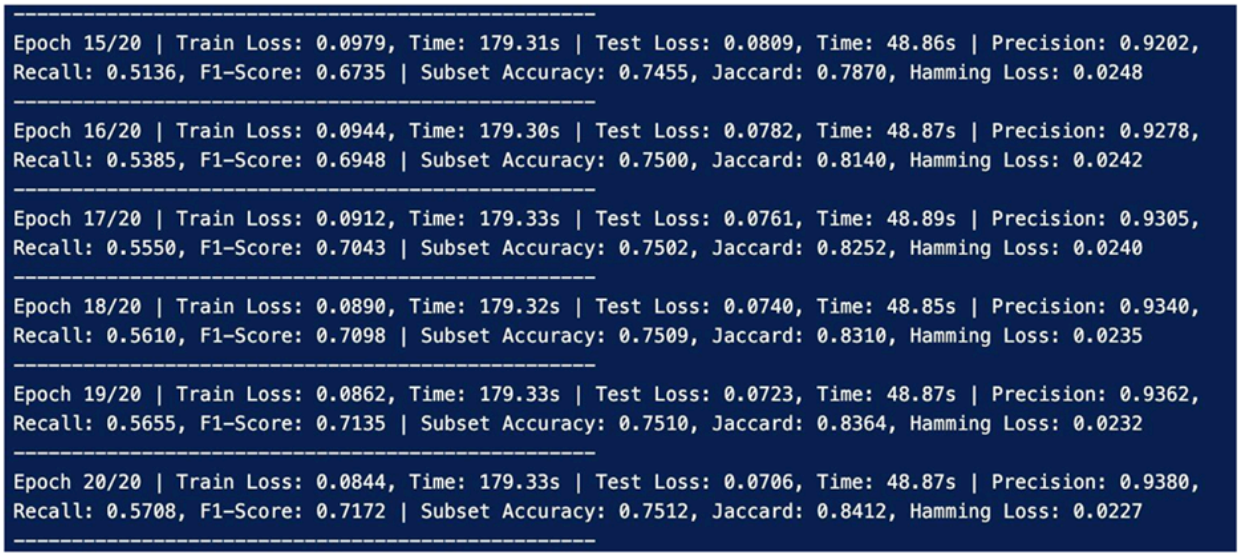

Figure 1: Training metrics of Bert Classification Model

The BERT model demonstrated strong performance in extracting SQL labels with high precision and recall. The multi-label classification pipeline achieved:

- **Precision:** 93.80%
- **Recall:** 57.08%
- **F1-Score:** 71.72%

These results validated the effectiveness of BERT's contextual embeddings in understanding SQL-related prompts.

The integration of label extraction into the T5 model significantly improved SQL generation accuracy. The T5 model achieved a BLEU score of 56.8654, highlighting its ability to translate labeled prompts into syntactically and semantically correct SQL queries.

5. Summary and Conclusion

My contributions to the project focused on implementing BERT for label extraction and integrating these outputs with the T5-based SQL generation model. This combination enhances the system's ability to translate natural language into SQL by providing explicit structural guidance, reducing ambiguity, and ensuring alignment with database schemas. The results demonstrate the robustness of this approach in handling diverse user inputs and complex query structures.

6. Contribution Score

The **contribution score** quantifies your individual contributions to the project based on the lines of code written, modified, and reused from external sources. The calculation is based on the following formula:

$$\text{Contribution Score} = \frac{\text{Total Lines of Code} - \text{Lines Modified}}{\text{Total Lines of Code} + \text{Lines from Sources}}$$

Lines of Code Contribution

- **BERT Model Implementation**
  - Total Lines of Code: 150
  - Lines Reused from Sources: 50
  - Lines Modified: 40
- **Label Extraction Component**
  - Total Lines of Code: 80
  - Lines Reused from Sources: 30
  - Lines Modified: 20

Calculation

$$\text{Contribution Score} = \frac{(150 + 80) - (40 + 20)}{(150 + 80) + (50 + 30)} = \frac{230 - 60}{230 + 80} = \frac{170}{310} \approx 0.548$$

Your **Contribution Score**: **0.55**

---

7. References

The following references were utilized for implementing the BERT and T5 models, as well as label extraction and similarity mapping techniques:

1. **Pre-trained BERT base-uncased Model** - Hugging Face
   URL: BERT Model Documentation
2. **Pre-trained T5-small Model for Seq2Seq Tasks** - Hugging Face
   URL: T5 Model Documentation
3. **BLEU Score Metric** - Medium
   URL: BLEU Score Explanation
4. **NLTK for Text Preprocessing** - NLTK Official Site
   URL: NLTK Documentation
5. **WordNet for Synonym Generation** - NLTK WordNet Interface
   URL: WordNet How-To Guide
6. **Fuzzy Matching Techniques** - Python Library difflib
   URL: Guide to Fuzzy Matching
7. **AdamW Optimizer** - PyTorch Documentation
   URL: AdamW Optimizer
8. **Multi-label Classification Metrics** - Scikit-learn
   URL: Scikit-learn Metrics
9. **Attention Mechanisms in Deep Learning** - Medium
   URL: Attention Mechanism Simplified
10. **Sequence-to-Sequence Learning Overview** - ArXiv
    URL: Seq2Seq Learning Paper