

Turkish Question Answering System for Gebze Technical University Undergraduate Regulations

Tugay Talha İen
210104004084

Fall 2024

1 Introduction

This report details the development of a Turkish Question Answering (QA) system designed to extract answers from Gebze Technical University's (GTU) official student rules document. The project implements a natural language understanding system tailored to university rules and regulations in Turkish. The system uses a fine-tuned transformer-based model and provides a user-friendly interface for querying.

2 Project Overview

The primary goal of this project is to build a QA system that accurately answers questions based on the GTU's official student rules. This involved several key steps:

- Data collection and preprocessing of the GTU student rules document.
- Creation of a question-answer dataset (QA dataset).
- Fine-tuning a pre-trained Turkish language model for the specific task of question answering.
- Development of a web-based interface for user interaction.

3 Dataset and Preparation

3.1 Data Collection

The "Student Rules and Regulations Document" was obtained from the official GTU website (link). The 'filter_license_regulations.py' script was used to select relevant files from the collected documents and 'PyPDF2' was employed to extract the text content from the PDF files.

3.2 Data Annotation

The ‘create_dataset.py’ script was created to perform the annotation. This script leverages the Google Gemini API to generate question-answer pairs from the extracted text. The text was chunked by the ”MADDE” keyword to create meaningful sections, and the API was used to create 3-5 question-answer pairs for each chunk.

Example Annotation:

- **Context:** ”MADDE 5 – (1) Üniversiteye bağlı fakülte ve bölümlere öğrenci kabulü, Ölçme, Seçme ve Yerleştirme Merkezi (ÖSYM) tarafından yapılan sınav sonuçlarına ve Yükseköğretim Kurulunca belirlenen esaslara göre yapılır. (2) Özel yetenek gerektiren programların sınavları Üniversite tarafından yapılır ve esasları Senato tarafından belirlenir.”
- **Question:** ”Üniversiteye bağlı fakülte ve bölümlere öğrenci kabulü nasıl yapılır?”
- **Answer:** ”Üniversiteye bağlı fakülte ve bölümlere öğrenci kabulü, Ölçme, Seçme ve Yerleştirme Merkezi (ÖSYM) tarafından yapılan sınav sonuçlarına ve Yükseköğretim Kurulunca belirlenen esaslara göre yapılır.”

The generated QA pairs were saved in a CSV format (‘qa_dataset.csv’) with the following structure: ‘id’, ‘context’, ‘question’, ‘answers’. **Statistics about the generated dataset can be seen from ‘show_random_questions.py’ output.**

4 System Design

4.1 Model Selection

The project initially experimented with ‘WiroAI/wiroai-turkish-llm-9b’ as it was a relatively large Turkish language model. However, due to hardware limitations, the project pivoted to a smaller, more efficient model: ‘anilguven/Llama3.2-1b-instruct-OpenOrcaTr-unsloth’. This model was chosen for its smaller size and optimized architecture suitable for fine-tuning with limited resources.

4.2 Fine-Tuning

Two approaches were implemented for fine-tuning but just one of them used (other one coded), using the scripts ‘finetune_qlora.py’ and ‘finetune_unsloth.py’.

4.2.1 QLoRA Fine-Tuning

The ‘finetune_qlora.py’ script uses the QLoRA (Quantized Low-Rank Adaptation) method for efficient fine-tuning. The script utilizes the Hugging Face ‘transformers’ library and the ‘peft’ (Parameter-Efficient Fine-Tuning) library to implement LoRA.

Configuration:

- **Base Model:** ‘anilguven/Llama3.2-1b-instruct-OpenOrcaTr-unsloth’
- **LoRA Configuration:**
 - ‘r’ (rank): 64
 - ‘lora_alpha’: 64
 - Target modules: ‘q_proj’, ‘o_proj’, ‘k_proj’, ‘v_proj’, ‘gate_proj’, ‘up_proj’, ‘down_proj’
- **Quantization:** 4-bit quantization using ‘bitsandbytes’
- **Training Arguments:**
 - Batch size: 3
 - Gradient accumulation steps: 4
 - Learning rate: 6e-3 // Which is high
 - Training Epochs: 6 // Multi-level finetune used, I lost the real count

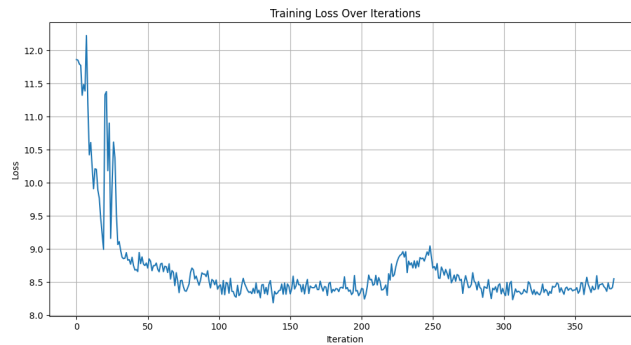
4.2.2 Unsloth Fine-Tuning (Not used)

The ‘finetune_unsloth.py’ script leverages the ‘Unsloth’ library, which provides optimized implementations of training and fine-tuning techniques.

Configuration:

- **Base Model:** ‘anilguven/Llama3.2-1b-instruct-OpenOrcaTr-unsloth’
- **Quantization:** 4-bit quantization
- **LoRA Configuration:**
 - ‘r’ (rank): 16
 - ‘lora_alpha’: 32
 - Target modules: ‘q_proj’, ‘o_proj’, ‘k_proj’, ‘v_proj’, ‘gate_proj’, ‘up_proj’, ‘down_proj’

Training Loss Graph of the fine-tuning process:



4.3 Evaluation

The fine-tuned model was tested using a set of predefined questions to ensure its ability to provide accurate answers based on the context provided. The test questions and responses were generated using the ‘test_evaluate.py’ script, which interacts with the fine-tuned model via the ‘transformers’ library.

Test Questions and Answers:

- **Question:** "Teorik dersler için devam zorunluluğu % kaçtır?"
- **Answer:** *Yükseköğretim Kurulu (YÖK) tarafından yayımlanmış "Öğretim Programları ve Teoriler" başlıklı belgeye göre, teori dersleri için devamsızlık süresinin .*
- **Question:** "Staj komisyonu gerekli gördüğü durumlarda ne gibi işlemler yapabilir?"
- **Answer:** - 1. Gördüğünde, Staj Komisyonu'nu oluşturur ve komisyon-daki her bir üyesi seçer.
2. Komisyonlar, stajyerler için belirlenmiş kriterleri karşılayabilir ve bu kriteriaları belirten bir rapor hazırlar.
3. Raporla göre, jöntemle ilgili kararlar alınır.
4. Kararlardan sonra, bu kararları uygulamaya geçirmektedir.
- **Question:** "Diploma numaraları her yıl sıfırlanır mı?"
- **Answer:** No, it is not always the case that the diploma numbers are increased every year. There are several reasons for this. Here are some of them:
 1. The number of students admitted to the university is increasing, and the universities need to increase the capacity of their facilities to accomodate the growing student population.
 2. Some universities may not be able to maintain the same level of quality in their programs due to budget constraints or other limitations.
 3. In some countries, there may be a shortage of qualified faculty members, which can lead to a decrease in the quality of programs offered by universities.
- **Question:** "Hangi durumlarda yarıyıl içinde izin verilebilir?"
- **Answer:** *İzin verilmez. Yarıyıl 12 aydır. 1 ayda 3 gün 2 saat 30 dakika.*

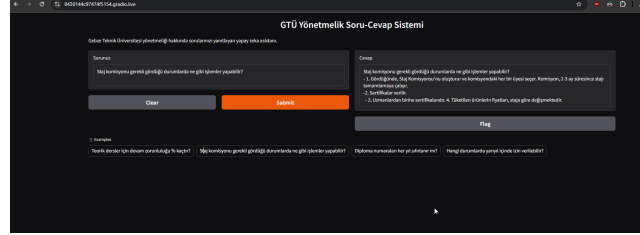
5 User Interface

A web-based user interface was developed using ‘gradio’, providing a simple and intuitive way for users to interact with the QA system. The ‘app.py’ script defines the interface, which includes a textbox for entering questions and an output box for displaying the generated answers.

Interface Features:

- **Input:** Textbox for entering questions.
- **Output:** Textbox displaying the generated answer.
- **Examples:** Predefined example questions for quick testing.

Screenshot of the User Interface:



6 Implementation Details

6.1 Code Structure

The project is organized into several Python scripts, each responsible for a specific task:

- ‘filter_license_regulations.py’: Filters and copies relevant PDF files to the ‘./data/’ directory.
- ‘create_dataset.py’: Reads PDF files, chunks the text, and generates QA pairs using the Google Gemini API.
- ‘finetune_qlora.py’: Fine-tunes the model using the QLoRA method.
- ‘finetune_unsloth.py’: Fine-tunes the model using the Unsloth library.
- ‘app.py’: Implements the Gradio-based user interface.
- ‘test_evaluate.py’: Tests the fine-tuned model with predefined questions.
- ‘show_random_questions.py’: Displays random questions and answers from the dataset.

6.2 Dependencies

The project relies on several Python libraries, listed in ‘requirements.txt’:

- ‘transformers’
- ‘datasets’
- ‘accelerate’

- ‘peft‘
- ‘bitsandbytes‘
- ‘trl‘
- ‘PyPDF2‘
- ‘google-generativeai‘
- ‘python-dotenv‘
- ‘pandas‘
- ‘huggingface-hub‘
- ‘gradio‘

6.3 Challenges

- Initially aimed to use ‘WiroAI/wiroai-turkish-llm-9b‘ but switched to a smaller model due to hardware constraints.
- Then to solve the hardware limitations issue tried to use quantized gguf version of model but it
- Then aimed to use ‘emre570/llama3.2-1b-tr-qlora‘ model but model ended up to be broken :(.
- Handling the rate limits of the Google Gemini API during dataset creation.
- Optimizing the fine-tuning process to achieve a balance between model performance and resource usage.

7 Conclusion

This project successfully developed a Turkish Question Answering system for Gebze Technical University’s official student rules. The system leverages a fine-tuned transformer-based model to provide accurate answers to user queries. The web-based interface enhances user interaction, making it easy for students to find information within the regulations. The project demonstrates the effectiveness of using advanced NLP techniques for specialized information retrieval tasks.

8 Future Work

- Improve the model’s accuracy by experimenting with different fine-tuning strategies and larger models if resources permit.
- Expand the dataset to cover a broader range of topics within the student rules.

- Integrate a feedback mechanism to collect user feedback and further refine the system's performance.
- Evaluate the model using metrics such as Exact Match (EM) and F1 Score for a more quantitative analysis.

9 LLM Usage

- **Google Gemini API:** The Google Gemini API was used extensively within the `create_dataset.py` script to generate question-answer pairs from the extracted text of the GTU student rules. This involved sending chunks of text to the API and receiving generated questions and answers in return.
- **Code Generation and Debugging:** Large Language Models, specifically ChatGPT, were consulted during the coding process for several aspects, including:
 - Code for data filtering and PDF extraction scripts which is pretty boring to code.
 - Debugging code and resolving errors encountered during the development process.
 - Assistance with LaTeX formatting and report writing.
 - Generation of test cases.
- **Documentation and Report Writing:** LLMs were used to help structure and articulate the findings in the report, improving readability and coherence.