NLP Project
Speech-to-text(STT) with Named-Entity-Recognition(NER)

Syllabus:

1. Introduction and Project Overview
2. Datasets and Preprocessing
3. Model Selection and Fine-Tuning
4. Challenges, Solutions, and Conclusion

Introduction and Project Overview

1.1 Project Goal

This project developed an end-to-end pipeline for processing Uzbek speech data, transcribing it into text, and extracting named entities. The pipeline integrates a fine-tuned Speech-to-Text (STT) model and a fine-tuned Named Entity Recognition (NER) model, both adapted for the Uzbek language. The aim was to create a robust system capable of accurately handling Uzbek audio and extracting key information.

1.2 Pipeline Components

The pipeline consists of two core modules:

- **Speech-to-Text (STT):** A pre-trained model is fine-tuned on Uzbek speech data to convert audio input into text.
- **Named Entity Recognition (NER):** A transformer-based model is fine-tuned to identify key entities within the transcribed text, including person names, dates, locations, organizations, and a custom entity type

1.3 Document Structure

This document details the project, encompassing the following sections:

- Datasets and Preprocessing
- Model Selection and Fine-Tuning
- Evaluation Metrics and Results
- Pipeline Integration and Usage
- Conclusion and Future Work

Datasets and Preprocessing

2.1 Speech-to-Text Dataset

- **Dataset:** The Mozilla Common Voice dataset (Uzbek subset) was used for fine-tuning the STT model. This dataset contains recorded speech segments and their corresponding transcriptions.
- **Preprocessing:**
    - **Audio Normalization:** Audio data was normalized to a standard loudness level to avoid inconsistencies.
    - **Splitting:** Data was split into train+validation, and test sets to prevent overfitting and for proper evaluation.

2.2 Named Entity Recognition Dataset

- **Dataset:** The Uzbek NER dataset provided by Risqaliyevds on Hugging Face was used. It includes text and annotations for the following entity types:
    - PERSON
    - DATE
    - LOC
    - ORG
    - EMAIL, and others
- **Preprocessing:**
    - **Tokenization:** Text data was tokenized using the tokenizer specific to our chosen model.
    - **NER Tag Alignment:** NER tags from the dataset are aligned with tokens by using "B-" and "I-" prefixes
    - **Data Splitting**: The dataset was split into train and test sets to prevent overfitting.
-

3: Model Selection and Fine-Tuning

3.1 Speech-to-Text Model

- **Base Model:** The Whisper-small model from OpenAI was used as the pre-trained base for the STT module. Whisper is a multi-lingual model trained on a large variety of speech data.
- **Fine-Tuning:**

```
per_device_train_batch_size=4,
gradient_accumulation_steps=2,
learning_rate=1e-1,
warmup_steps=500,
max_steps=3000,
```

```
num_train_epochs=5,
fp16=True,
evaluation_strategy="epoch",
per_device_eval_batch_size=4,
generation_max_length=225,
logging_steps=25,
remove_unused_columns=False,
label_names=["labels"],
gradient_checkpointing=True,
```

3.2 Named Entity Recognition Model

- **Base Model:** The dbmdz/bert-base-turkish-cased model was selected, as it had a strong performance on related tasks.
- **Fine-Tuning:**

```
evaluation_strategy="epoch",
learning_rate=5e-5,
per_device_train_batch_size=16,
per_device_eval_batch_size=16,
num_train_epochs=3,
weight_decay=0.01,
logging_dir="./logs",
save_strategy="epoch",
load_best_model_at_end=True,
metric_for_best_model="overall_f1",
report_to="none"
```

- **Low-Rank Adaptation (LoRA):** Applied LoRA to reduce the number of trainable parameters during fine-tuning and to achieve better results.

3.3 Hyperparameter Tuning

The learning rate, batch size, and number of epochs were selected based on best practices. I tried several times to select the best option.

4: Challenges, Solutions, and Conclusion

The biggest challenge I faced was the GPU limitations. I only had access to free GPU options, such as those on Kaggle and Google Colab. Unfortunately, none of them were sufficient to fully train the STT (Speech-to-Text) model. Despite trying multiple times(with different parameters), the training was never completed successfully. As a result, I was unable to finalize my STT model.

For the NER (Named Entity Recognition) task, this limitation was less of an issue because the dataset and model parameters were much smaller, making it easy to train on free GPUs.

Another significant challenge was time management. Due to university exams, I started this task quite late, and now I urgently need to travel to Namangan. Unfortunately, this means I cannot make further improvements before the deadline.

However, I plan to address these challenges in the future. Once I upgrade my GPU resources and resolve other issues, I will improve both models and integrate them into a complete pipeline.

For the pipeline, I used an existing Uzbek STT model (oyqiz-stt) instead of mine own STT model. It performed well and met the requirements for this task.

I plan to address some issues with the NER model by experimenting with different approaches and fine-tuning its parameters. For the STT model, I will retry training later with optimized parameters. Once I have access to a more powerful GPU, I also intend to incorporate additional algorithms to improve the STT model's performance.