English to Egyptian Arabic Translation Using Pretrained Models and QLoRA

Project Overview

Goal: Translate English text into Egyptian Arabic (dialect) using AI.

Key Features:

- Started with a pretrained model (NLLB-200) for Modern Standard Arabic.
- Fine-tuned it to adapt to the Egyptian dialect.
- Used parameter-efficient fine-tuning (QLoRA) to save resources.

Dataset Information

Source: Preprocessed dataset from Egyptian TV series dialogues.

Details:

- Contains parallel English and Egyptian Arabic (spoken) sentence pairs.
- Focuses on conversational and informal language.
- Small-scale dataset, requiring efficient fine-tuning techniques.

Step 1: Using a Pretrained Translation Model

Technology: Pretrained Language Model (NLLB-200, by Meta)

What it does:

- Already knows how to translate between many languages.
- We load it and test it out-of-the-box to translate English to Modern Standard Arabic.

How it works:

- The model converts English sentences into Arabic by learning patterns from huge text data.

Model Details

Base Model: NLLB-200 Distilled (600M parameters)

Key Features:

- Supports over 200 languages (including Modern Standard Arabic).
- Designed for high-quality translations with minimal errors.
- Distilled version for faster performance and lower memory usage.

Fine-Tuning Technique: QLoRA (Quantized LoRA)

- Reduces memory usage by using 4-bit quantization.
- Adds lightweight adapters (LoRA) to train new tasks efficiently.

Step 2: Preparing Data for Fine-Tuning

Technology: Tokenization (HuggingFace Tokenizer)

What it does:

- Converts sentences into numbers that the model can understand.
- Prepares both English inputs and Egyptian Arabic targets.

How it works:

- Words and sentences are broken down into tokens (pieces), then changed into IDs for the model to learn from.

Step 3: Model Quantization and Adapting for Fine-Tuning

Technology: BitsAndBytes (quantization), LoRA (parameter-efficient fine-tuning)

What it does:

- Shrinks the model to use less memory and work faster (quantization).
- Adds a small adapter to the model so it can learn new things (LoRA).

How it works:

- Quantization stores numbers in a smaller format.
- LoRA lets us teach the model new tricks without changing its main knowledge.

Step 4: Fine-Tuning the Model

Technology: HuggingFace Trainer, TensorBoard

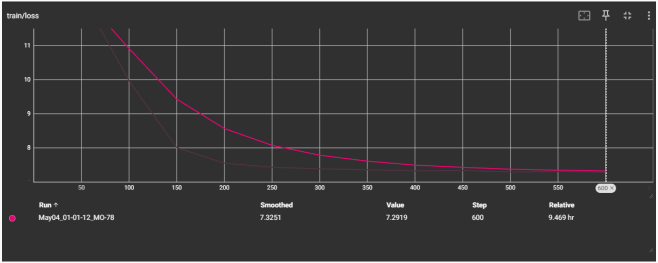
What it does:

- Trains the adapter on our sentences, so the model starts to speak more like an Egyptian.
- Tracks the model's progress during training.

How it works:

- The trainer updates only the adapter's parameters to learn the new dialect quickly.
- We can see the training curve in TensorBoard.

Training Progress



	Run ↑	Smoothed	Value	Step	Relative
●	May04_01-01-12_MO-78	7.3251	7.2919	600	9.469 hr

Model Training Progress (TensorBoard Curve)

Step 5: Evaluating the Fine-Tuned Model

What we did:

- Compared translations from the base model (Modern Standard Arabic) and our fine-tuned model (Egyptian Arabic).

Sample outputs:

English	**Model Output**
I want to eat	*Base:* أريد أن أكل *Fine-tuned:* عايز أكل.
Where is the market?	*Base:* أين السوق؟ *Fine-tuned:* فين السوق؟
This is delicious	*Base:* هذا رائع *Fine-tuned:* ده حلو جدا
It's very hot today	*Base:* الحرارة اليوم حارة جداً *Fine-tuned:* اليوم حار جدا.

Model Limitations

Challenges:

- Limited dataset size: May not cover all variations of Egyptian Arabic.
- Dialect-specific nuances: Some slang or regional expressions might still be mistranslated.
- Requires more computational time to fine-tune on larger datasets.

Future Improvements:

- Expand dataset with more diverse dialogues.
- Test with other dialects or informal settings.
- Optimize training further for better slang handling.

Thank You!