Fine-Tuning a Large Language Model (LLM) for Italian-to-Neapolitan Dialect Translation

January 15, 2025

Aur Marina Iuliana

Abstract

The project aims to fine-tune a Large Language Model (LLM) to enhance its understanding and generation of text in the Neapolitan dialect. To address training time and GPU memory constraints, Low-Rank Adaptation (LoRA) is used for efficient fine-tuning. Evaluation metrics, including Perplexity, BLEU score, ROUGE-L score, and BERT score, are used to compare the fine-tuned model with the vanilla version, highlighting the results and suggesting directions for future work.

1. Introduction and Motivation

In recent years, the latest versions of LLMs have introduced innovative methodologies for fine-tuning, significantly enhancing the adaptability of these models for a wide range of applications. (Santilli & Rodolà, 2023). Although most LLMs already support the Italian language, evaluating their performance in generating and understanding Italian dialects could provide valuable insight. Specifically, this project focuses on the **Neapolitan dialect**, as it represents one of the most distinctive Italian dialects.

2. Dataset

The dataset used is sourced from HuggingFace. It contains a comprehensive collection of traditional Neapolitan songs from napoligrafia translated into the Italian language.

Specifically, the dataset includes three main fields: `url`, which contains the source URL from which the data was collected, `napoletano`, which contains the original text in the Neapolitan dialect, and `italiano`, which includes the corresponding translation into Italian. The dataset consists of a total of **14.2k examples**: 80% of the data are used for training, 10% for validation, and 10% for testing.

2.1. Data Format

For effective fine-tuning, it is crucial to format the data according to a specific recipe. An example of a **chat template for fine-tuning** is as follows:

```
<bos><start_of_turn>user
 Translate the provided text from Italian language to
 Neapolitan dialect. Return only the text translated in
 Neapolitan, without any additional details.
 Italian Text: Ma al tramonto giu’ a Posillipo<end_of_turn>
<start_of_turn>model
Ma ’int’o tramonto ’nterra Pusilleco<end_of_turn>
```

For **inference**, the chat template includes a **generation-special token** at the end, as shown below:

```
<bos><start_of_turn>user
 Translate the provided text from Italian language to
 Neapolitan dialect. Return only the text translated in
 Neapolitan, without any additional details.
 Italian Text: E’ quasi ottobre, mi sembra inverno<end_of_turn>
<start_of_turn>model
```

3. Gemma-2-2B-it Model

The selected model for this project, **`Gemma-2-2B-it`**, is an instruction-tuned LLM specifically designed for dialogue scenarios. This makes it an ideal candidate for our project, where we aim to fine-tune the model to translate from Italian to the Neapolitan dialect in a chat-based format.

4. Data Preparation and Analysis

4.1. Exploratory Data Analsys (EDA)

An exploratory data analysis (EDA) is performed to obtain information about the data. First, we examine the most common words in the Neapolitan text, their percentage distribution, and the lexical diversity in the dataset. We found that the dataset is **well-distributed**, as no single word frequency significantly dominates the others, with a **lexical diversity** of 0.16. An analysis of the **sentence length distribution** in the dataset reveals an average sentence length of approximately 40 characters.

Email: Aur Marina Iuliana <aur.1809715@studenti.uniroma1.it>.

Deep Learning and Applied AI 2024, Sapienza University of Rome, 2nd semester a.y. 2023/2024.

Subsequently, we investigate the **linguistic divergence** of the Neapolitan dialect from the Italian language. We found an average divergence of 0.155, suggesting that the Neapolitan dialect and Italian share many similarities in their textual representation (embeddings). Despite this, as shown in the graphical representation of the **textual representations** in Figure 1, the model is capable of distinguishing between the two languages, resulting in two separated clusters.

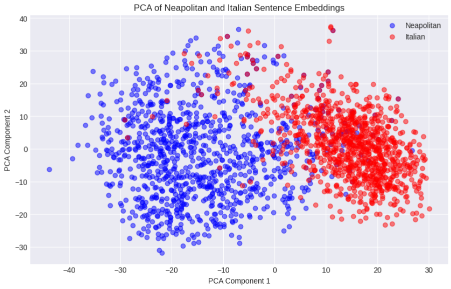

Figure 1. PCA of Neapolitan and Italian Sentence Embeddings

5. Fine-Tuning

Training a new tokenizer from scratch is computationally demanding and data-intensive. Instead, **augmenting an existing tokenizer** is a more efficient approach, preserving the model's original language capabilities while incorporating the new language.

5.1. Low-Rank Adaptation (LoRA)

LLMs are often high-dimensional and over-parametrized, while the information encoded in them tends to be well-approximated in a much lower dimension (Frankle & Carbin, 2019). **`Low-Rank Adaptation (LoRA)`** is a Parameter-Efficient Fine-Tuning (PEFT) technique that decreases the number of parameters for fine-tuning by decomposing large matrices. This reduction in parameters results in a decrease in training time and GPU memory usage, while preserving the quality of the results. After applying LoRA with a rank of 16 and a scaling factor of 32, only **0.79% of the parameters** are trained during the fine-tuning process.

5.2. Fine-Tuning Results

The **`SFTTrainer`** is used for supervised fine-tuning of the Gemma model. The training arguments and graphical representations of the fine-tuning process are available for review on Weights & Biases Project. The fine-tuning process is configured to run for two epochs, with two evaluation steps performed per epoch. As illustrated in Figure 2, both the training and evaluation losses decrease over time and the gradients stabilize. This trend indicates that the model is effectively learning to perform the task and improving its understanding of the Neapolitan dialect.

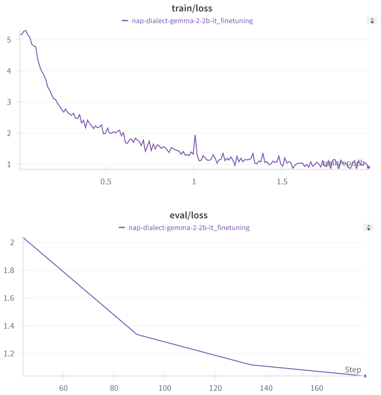

Figure 2. Training and Evaluation Losses over Time

6. Performance Evaluation

6.1. Comparison of the Models

Table 1 provides a comprehensive summary of the performance of the fine-tuned and vanilla models. The results clearly indicate that the fine-tuned model exhibits better performance across all metrics. This suggests that the fine-tuned model is more confident in predicting each token based on the prior tokens and that the generated text by the fine-tuned model closely approximates the expected output.

Table 1. Performance comparison

Models	BLEU score	rougeL score	BERTf1 score	ppl
Vanilla	0.015	0.284	0.747	180.223
FineTuned	0.230	**0.657**	**0.877**	**68.377**

7. Qualitative Assessment and Future Work

Examples of **generated sentences** by the fine-tuned and vanilla models can be found in the two tables on Weights & Biases Project. The results are highly promising, as the text produced by the fine-tuned model is accurately written in the Neapolitan dialect and closely aligns to the specific requirements of the translation task. The dataset used in this project is highly specific and relatively small, focusing on the domain of Neapolitan songs. For future work, it would be interesting to expand the dataset by incorporating a wider variety of text types, such as conversations, proverbs, and descriptions, to make the learning process more robust and comprehensive (Jang et al., 2024).

References

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019. URL https://arxiv.org/abs/1803.03635.

Jang, D., Byun, S., Jo, H., and Shin, H. Kit-19: A comprehensive korean instruction toolkit on 19 tasks for fine-tuning korean large language models, 2024. URL https://arxiv.org/abs/2403.16444.

Santilli, A. and Rodolà, E. Camoscio: an italian instruction-tuned llama, 2023. URL https://arxiv.org/abs/2307.16456.