Final Report - Fine Tuning Methods for Low-resource Languages

Anton Johansson
Department of Computer Science,
Tsinghua University
wh24@mails.tsinghua.edu.cn

Tim Bakkenes
Department of Computer Science,
Tsinghua University
dim24@mails.tsinghua.edu.cn

Daniel Wang
Department of Electronic Engineering,
Tsinghua University
wyl24@mails.tsinghua.edu.cn

Abstract

The rise of Large Language Models has not been inclusive of all cultures. The models are mostly trained on English texts and culture which makes them underperform in other languages and cultural contexts. By developing a generalizable method for preparing culturally relevant datasets and post-training the Gemma 2 model, this project aimed to increase the performance of Gemma 2 for an underrepresented language and showcase how others can do the same to unlock the power of Generative AI in their country and preserve their cultural heritage.

1 Introduction

In this report, the final project of the Fall 2024 Machine Learning course is described and discussed. This project was inspired by the Kaggle competition *Google - Unlock Global Communication with Gemma*, which aims to improve the performance of the Gemma 2 LLM for underrepresented languages as an innovative solution for global communication [13]. This competition is in line with the UNESCO initiative *'Language Technologies for All'*, ensuring that AI models support a broad range of languages, which is vital to preserve global cultural heritage and prevent digital language extinction [1].

1.1 Background

The fast progress in Artificial Intelligence (AI) means that more and more people are using Large Language Models (LLMs). Since most LLMs are trained primarily on English data, other languages and cultures risk fading away, a concern highlighted by the World Economic Forum[23]. As these biased models become increasingly integrated into society, their potential to amplify existing biases and generate harmful content is intensified, as noted by Torrielli [22]. Also, it contributes to the "digital language death," where minority languages risk losing their online presence according to Soria et al. [20]. This emphasizes the importance of developing LLMs that support more than just the English language and culture. Despite some multilingual capabilities, models like Gemma often underperform in non-English languages, showing reduced accuracy and consistency [23].

1.2 Importance and Impact

It has been established that Google's Gemma 2 model was primarily trained on English data, leading to reduced performance in other languages and cultures. According to Yogarajan et al. [25], the ideal scenario would be to design LLMs with underrepresented groups in mind from the start. However, even though this might be possible in the future, the bias problem needs to be addressed immediately.

By showcasing methods and technologies whilst developing a fine-tuned version of Gemma 2, other developers will have a roadmap to adapt Gemma 2 for their purposes. Tailoring LLMs to underrepresented languages will empower people from all around the globe to utilize LLMs to solve diverse real-world problems. At the same time, it can be used to preserve the linguistic and cultural identities of local communities and enhance global communication.

Further highlighting the importance of their project, recent efforts such as BigScience's BLOOM [6] and Meta's "No Language Left Behind" [2] illustrate the growing focus on multilingual AI. These open-source projects highlight the urgency of addressing language coverage gaps. Also, Europe's eTranslation platform [7] and UNESCO's digital preservation efforts [1] emphasize the global demand for robust multilingual technologies.

However, the applications of tailoring LMMs is not limited to language and culture. The methods proposed could also be used to tailor models for specific business related tasks [10]. Furthermore, the model developed during this project could enable non-English speakers to utilize a LLM.

2 Formal Problem Definition

The goal is to fine tune googles' pre-trained Gemma 2 model for other languages than English and showcase and document the methods used to achieve this. In order to do this, a processed dataset used for post-training to fine tune the model, D', has to be crafted from various sources. The Gemma 2 model parameters, θ, and the context supplied via features like Retrieval-Augmented-Generation, C_{LLM}, should be optimized to minimize the loss function that can be expressed with the following equation where y is the expected result and $M_{\text{Gemma2}}(x, C_{\text{LLM}}; \theta)$ is the predicted result of the model given the prompt x, the context C_{LLM} and the model parameters θ.

$$\sum_{(x,y)\in D'} L(y, M_{\text{Gemma2}}(x, C_{\text{LLM}}; \theta))$$

In other words, the goal is to find the optimal model parameters, θ', and the best context generation to provide the optimal context, C'_{LLM}, that minimizes the loss function. The problem is expressed in the equation below.

$$(\theta', C'_{\text{LLM}}) = \arg\min_{\theta, C_{\text{LLM}}} \sum_{(x,y)\in D'} L(y, M_{\text{Gemma2}}(x, C_{\text{LLM}}; \theta))$$

The explanation above is a general description for the underlying optimization problem that we have worked on. However, in reality it is not clear what a good response is when it comes to language, historical facts and cultural relevance. The goal could therefore be augmented by explaining how a *good response* can be defined in terms of cultural authenticity, factual accuracy, and historical relevance. These criteria could for example be integrated into the training and evaluation by adding human feedback. Because of this, the language that was chosen for this project was Swedish since two thirds of the group members have native proficiency in the language.

3 Related Work For Inspiration

Recently Google [9] has fine-tuned Gemma 2 for Japan. They used reinforcement learning from human feedback and specialized post-training techniques to enhance performance in Japanese, without impacting the English performance. Furthermore, AI Sweden [3] has developed a Swedish large-scale generative language model to help Swedish organizations build language applications that previously were not possible. To do this, AI Sweden had to source Swedish data to train the model and documented it well which makes their work relevant to this project.

Besides Google's work on Gemma 2 and AI Sweden's Swedish model, projects previously mentioned such as BLOOM [6] and Meta's "No Language Left Behind" [2] have released extensive open-source multilingual datasets and training pipelines. These resources can inform best practices for those seeking to adapt Gemma 2 to underrepresented languages.

4 Methodology

This section will cover the methodology for completing this project. There were many parts to consider. For example, a dataset had to be crafted to fine tune the model for it to learn how to write better in Swedish. Another dataset for giving the model external data about Swedish culture and history also had to be curated. Then the model also had to be fine tuned and the mechanics for retrieving relevant context given a prompt also had to be understood.

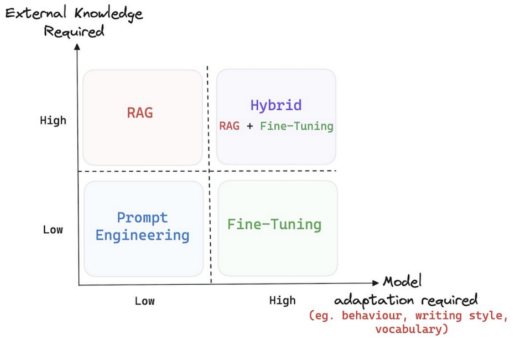


Figure 1: Method Matrix [17]

As can be seen in figure 1, a hybrid approach was used in this project where both external knowledge and model adaptation to a specific language was required. These topics are covered in more detail in the following subsections.

4.1 Datasets

There were two main parts to this project, fine tuning Gemma 2 for Swedish language and enriching its responses by incorporating external data as context. This meant that two datasets had to be created. One fine tuning dataset and one retrieval augmented generation dataset.

4.1.1 Fine Tuning Dataset

Since Gemma 2B already has a basic understanding of Swedish, and given our computational resources are limited while the model itself is relatively small (2 billion parameters), we decided to fine-tune it using a dataset of 500 prompts and related answers.

We began the fine-tuning process by curating a small dataset of 50 prompts, observing a slight performance boost. Encouraged by this, we expanded the dataset to 100 prompts, then to 200 prompts, and finally to 500 prompts. At this stage, we saw noticable improvement in performance. However even more prompts would likely improve performance even more but due to time and computational constraints we settled for 500 prompt/response pairs. The prompts were hand-crafted by us with assistance from ChatGPT. For generating these prompts, we used a pretrained Swedish language model available in the OpenAI store and prompted it to create prompts and answers in Swedish suitable for a large language model [18]. The prompts were designed to cover a diverse range of topics, including Swedish culture, the Swedish language, education in Sweden, translation tasks, and other commonly encountered subjects.

Once the prompts and answers were generated, each response was constrained to approximately 25 tokens in length. We then manually reviewed all the answers to ensure they were free from grammatical errors and factual inaccuracies. The model used to generate the prompt-answer pairs performed well, with an error rate of approximately one mistake per five answers.This systematic approach allowed us to fine-tune Gemma 2B effectively while ensuring high-quality data for the fine-tuning process.

In the table below, we present four examples of prompts and their corresponding answers in Swedish, along with their English translations for clarity and ease of understanding.

Table 1: Prompts and Answers in Swedish and English

Prompt (Svenska / English)	**Svar (Svenska)**	**Answer (English)**
Vad är typiskt för svensk midsommar? / What is typical of Swedish midsummer?	Svensk midsommar är en av de mest älskade högtiderna. Den firas med dans runt midsommarstången, traditionella sånger och mat som sill, potatis, gräddfil och jordgubbar. Kvinnor bär kransar av blommor i håret och firandet sker ofta utomhus när vädret tillåter det, dock regnar det ofta på midsommar. Det är en högtid fylld med gemenskap och glädje.	Swedish midsummer is one of the most loved holidays. It is celebrated with dancing around the midsummer pole, traditional songs and food such as herring, potatoes, sour cream and strawberries. The women wear flowers in their hair and the celebration often takes place outdoors when the weather permits, however it often rains during midsummer. It is a holiday filled with community and joy.
Hur fungerar Allemansrätten? / How does "Allemansrätten" work?	Allemansrätten ger alla i Sverige rätt att vistas i naturen, oberoende om det är privat eller allmän mark. Man får plocka bär och svamp och bada i sjöar, så länge man inte stör eller förstör. Det är en unik del av svensk kultur och naturvård.	"Allemansrätten" gives everyone in Sweden the right to be in nature, regardless of whether it is private or public land. You can pick berries and mushrooms and swim in lakes, as long as you don't disturb or destroy the environment. It is a unique part of Swedish culture and nature conservation.
Hur hanterar jag stress under tentaperioden? / How do I deal with stress during the exam period?	Planera studierna i små block med pauser, motionera för att släppa stress och sov regelbundet. Öva avslappningsövningar som djupandning vid behov.	Plan your studies in small blocks with breaks, exercise to release stress and sleep regularly. Practice relaxation exercises such as deep breathing when needed.
Vad är en synonym för ordet "fantastisk"? / What is a synonym for the word "awesome"?	En passande synonym är "enastående" eller "underbar", beroende på sammanhang.	A suitable synonym is "outstanding" or "wonderful", depending on the context.

4.1.2 Retrieval Augmented Generation Dataset

In order to be able to do Retrieval Augmented Generation, henceforth referenced as RAG, a dataset must first be created that contains the knowledge that is wished to be known by the model. The task of creating a good dataset is difficult and often requires a lot of people and time. Because of this, the size of the dataset was kept small because of the limited time and number of people involved in this project. The RAG still managed to give meaningful results. There are many things to think about when creating a dataset from scratch. Below follows information about how it can be done and how it was done for this project.

Identify sources and domain

First of all, feasible sources that can be used to collect data for the given domain has to be identified in order for the author of the dataset to find the relevant data. At the same time as identifying relevant sources, the domain that the system needs to have external data incorporated from also has to be identified. For this project the domain was Swedish culture and history and the main sources used was various museums in Sweden that have published information about the history and culture of Sweden throughout the years, Wikipedia[21] articles about the culture of Sweden and an initiative called LitteraturBanken[15]. LitteraturBanken[15] is an initiative that is aimed at preserving Swedish literary history and culture and provides it for free to the public to use in various formats.

Collect and Curate Data

Then raw data can be collected from these sources that could for example be in the form of epub files, scanned in images or readable pdfs. It is important to be careful when curating the dataset to make sure it is both reliable and comprehensive enough. In this project, the data was sourced in all of these formats and the dataset was curated to capture various aspects of swedish culture. From

more factual content from Wikipedia[21] and Museums to less factual from classic swedish literature sourced from LitteraturBanken[15].

Clean and Preprocess Data

Once raw data has been collected, it has to be cleaned and preprocessed. This includes:

- Removing duplicate information.
- Correcting spelling mistakes or OCR (Optical Character Recognition) errors if the data format is scanned in images.
- Taking care of newlines, punctuation and special characters.

Well cleaned data helps with the accuracy of the RAG. For this project, all of the mentioned data formats was experimented with but in the end scanned in images was not put in the final dataset due to bad quality of the processed data. It turned out to be much simpler to process the data when OCR errors did not have to be handled. The notebooks associated with this project still contained the code that was used to experiment with these kinds of data sources.

Chunk and Index the Data

To make retrieval efficient and relevant, it is common to break down the text into chunks. This can be done for different chunk sizes, in the case of this project the chunk size was 200 words. These chunks can then be indexed using a vector representation and searched for using similarity search with the prompt sent by the client.

Generating Embeddings

In order to do RAG, the goal is to generate vector embeddings for each chunk using an embeddings model. These embeddings are supposed to capture semantic relationships between chunks of text that makes it possible for a program to retrieve relevant context from the prompt sent by the client. There are a couple of ways to do this. There are embeddings model available such as Sentence-BERT or you could use a framework to train your own model based on the data. Generally, it requires a lot of data to train a good model for generating embeddings. But is was still attempted in this project and gave some promising results in some cases. However, it was not always reliable which will be shown more in section *4.4*. Therefore, a pre-trained embeddings model was also tested, `KBLab/sentence-bert-swedish-cased`. Since this project is about underrepresented languages and there might not be a model like this for all languages the goal of training one from the data was to demonstrate feasability even for other languages.

Deploy and Maintain

Finally, the indexed dataset could be hosted in for example a vector database so that it can be queried by the RAG system. In order to maintain it over time the dataset will require periodic updates and maintenance as the domain evolves and the capacity of the model is developed. Due to the financial limitations of this project the vector database was simulated by keeping a JSON-file of the embeddings.

4.2 Model Selection

The Kaggle competition does not specify a particular version of the Gemma 2 model to be used, so it was up to us to choose which one to use. First of all, the Gemma 2 model exists in different parameter sizes: 2B, 9B, and 27B. We opted for the 2B size because it balances performance and computational requirements, making it manageable to fine-tune on a single GPU while still being powerful. Furthermore, the specific model we chose was `gemma2-keras-gemma2_instruct_2b_en-v1`, which is already optimized for following prompts and producing coherent outputs, as it starts from an instruction-aware checkpoint. Additionally, many other participants in the competition used this model, indicating its fit for this task. We also experimented with a Gemma 2 base model that was not fine-tuned, but it proved to be much more difficult to work with and got worse results.

Moving on, we chose to work in TensorFlow for several reasons. First, it's a widely used industry standard, so learning it gives us valuable, practical experience. Second, the abundance of existing code and tutorials for fine-tuning in TensorFlow makes problem-solving easier. Finally, TensorFlow's

strong integration with TPU/JAX and other Google infrastructure offers better performance and streamlined deployment options. Also, we chose to write our code in Google Colab and run it on an A100 GPU for convenience and efficiency. Colab provides a simple environment and collaborative features enabling all of us to code in the same notebook. Also, the A100 GPU with 90 GB of VRAM significantly accelerates large-scale deep learning tasks compared to local hardware, enabling faster model training and experimentation.

4.3 Fine-Tuning

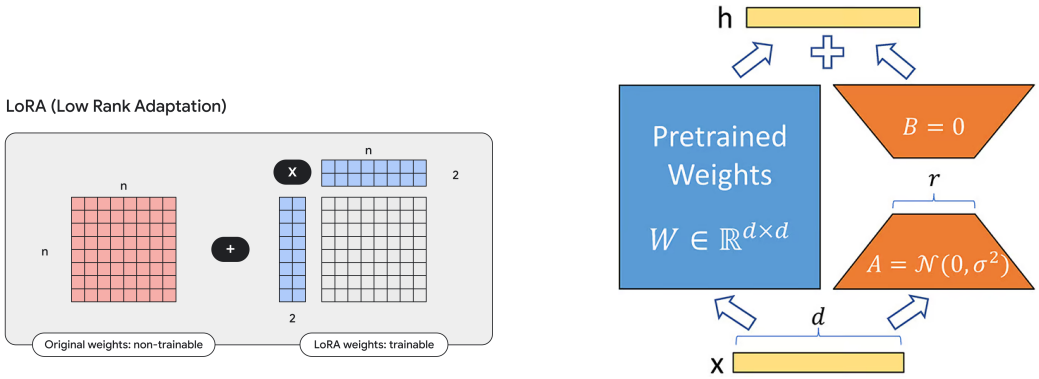
Fine-tuning refers to the process of adapting a pretrained language model to a specific task by modifying a subset of its parameters. In this project, we utilized the GemmaCausalLM model with Low-Rank Adaptation (LoRA) to efficiently fine-tune the model for our downstream task.

4.3.1 Model Architecture and Adaptation

We employed LoRA for parameter-efficient fine-tuning. This technique modifies a small subset of the pretrained weights using low-rank matrices as described in the formulas below:

$$W + \Delta W = W + BA, \quad \text{where } B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}, r \ll d$$

The advantage of LoRA is that it can efficiently adapt a pretrained model to a specific task while significantly reducing computational requirements. Unlike full fine-tuning, LoRA freezes most of our model parameters to efficiently fine tune Gemma 2 to Swedish. This can be seen in figure 2 below.



(a) LoRa [11] (b) LoRa [5]

Figure 2: LoRa configuration.

This approach minimizes the number of trainable parameters, enabling efficient training on hardware with limited resources. As our group had access to limited resources, it was a natural choice to use this method. Additionally, LoRA facilitates faster gradient computations, which contributes to shorter training times and smoother convergence.

Rank Selection and Considerations

The choice of the rank parameter $r = 8$ was chosen based on empirical evidence and our specific requirements. Initial experiments demonstrated that this rank provided sufficient expressiveness for the updates while maintaining the desired accuracy. Higher ranks, while potentially improving the model's performance further, were avoided due to memory constraints. The hardware used in this project imposed strict limitations on memory utilization, and a rank of 8 balanced the trade-off between task complexity and computational efficiency.

Preprocessor: "gemma_causal_lm_preprocessor"

Layer (type)	Config
gemma_tokenizer (GemmaTokenizer)	Vocab size: 256,000

Model: "gemma_causal_lm"

Layer (type)	Output Shape	Param #	Connected to
padding_mask (InputLayer)	(None, None)	0	-
token_ids (InputLayer)	(None, None)	0	-
gemma_backbone (GemmaBackbone)	(None, None, 2304)	2,614,341,888	padding_mask[0][0], token_ids[0][0]
token_embedding (ReversibleEmbedding)	(None, None, 256000)	589,824,000	gemma_backbone[0][0]

Total params: 2,614,341,888 (9.74 GB)
Trainable params: 2,614,341,888 (9.74 GB)
Non-trainable params: 0 (0.00 B)

Preprocessor: "gemma_causal_lm_preprocessor"

Layer (type)	Config
gemma_tokenizer (GemmaTokenizer)	Vocab size: 256,000

Model: "gemma_causal_lm"

Layer (type)	Output Shape	Param #	Connected to
padding_mask (InputLayer)	(None, None)	0	-
token_ids (InputLayer)	(None, None)	0	-
gemma_backbone (GemmaBackbone)	(None, None, 2304)	2,620,199,168	padding_mask[0][0], token_ids[0][0]
token_embedding (ReversibleEmbedding)	(None, None, 256000)	589,824,000	gemma_backbone[0][0]

Total params: 2,620,199,168 (9.76 GB)
Trainable params: 5,857,280 (22.34 MB)
Non-trainable params: 2,614,341,888 (9.74 GB)

Figure 3: Comparison of model summaries before and after fine-tuning.

The fine-tuning process with LoRA significantly reduces the number of trainable parameters from 2.61 billion (9.74 GB) to 5.86 million (22.34 MB), highlighting its efficiency in adapting large models with minimal computational overhead.

4.3.2 Training Setup

The training setup for fine-tuning the GemmaCausalLM model was carefully designed to optimize the model's performance while ensuring efficient utilization of computational resources. In this subsection we will explain the data preprocessing and model configuration.

Data Preprocessing

To ensure robust training, the dataset was prepared by converting the formatted data into a TensorFlow Dataset object, enabling efficient handling and batching of training and validation examples. The dataset was shuffled with a buffer size equal to the total size, ensuring unbiased training by randomizing the order of examples. Prefetching, implemented using `tf.data.AUTOTUNE`, improved input pipeline efficiency by preparing data while the model processed the previous batch. Also, a train-validation split of 90%-10% was applied to maintain a consistent evaluation process. Both subsets were batched with a size of 16 for efficient gradient computation.

Model Configuration

The GemmaCausalLM model was initialized with LoRA enabled at a rank of 8. Mixed precision training was employed to leverage both float16 and float32 data types, accelerating training and reducing memory usage. Additionally, the sequence length was adjusted to 128 tokens, balancing computational efficiency and contextual understanding.

The optimizer used was AdamW after experimenting with Adam and Stochastic Gradient Descent as well. Furthermore, gradient clipping is also utilized to make sure the gradients do not grow too large in size. As for the learning rate, a scheduler was used. There were a couple to chose from as can be seen in figure 5. In the end, the one that was chosen was to use Cosine Decay. To further optimize the training and fine-tuning we also added Warm-Up to the scheduler which makes the learning rate first gradually increase at the start of the training

To achieve efficient training and fine-tuning of the model, we employed a Cosine Decay with Warm-Up learning rate schedule. This approach ensures a gradual increase in the learning rate during the initial phase of training, followed by a smooth decay over the remaining training steps. This strategy helps stabilize the training process early on and facilitates convergence to an optimal solution.

Algorithm 1 AdamW Optimizer

given $\beta_1, \beta_2, \epsilon, \lambda, \eta, f$
initialize $\theta_0, m_0 \leftarrow 0, v_0 \leftarrow 0, t \leftarrow 0$
while θ_t not converged **do**
  $t \leftarrow t + 1$
  $g_t \leftarrow \nabla_\theta f(\theta_{t-1})$
  update EMA of g_t **and** g_t^2
  $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
  bias correction
  $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$
  $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$
  update model parameters
  $\theta_t \leftarrow \theta_{t-1} - \eta_t (\hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon) + \lambda \theta_{t-1})$
end while
return θ_t

Figure 4: AdamW Algorithm [12]

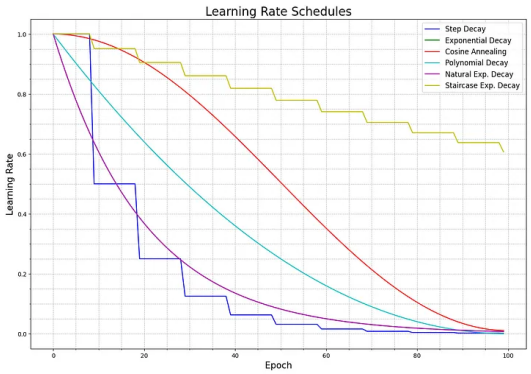


Figure 5: Learning Rate Schedulers Comparison [16]

4.3.3 Training Workflow

The training process was structured into a systematic workflow to ensure stability and performance improvements over epochs. Key components of the workflow included training loops, callbacks and monitoring, as seen below:

Training Loop

The training loop iteratively updated model parameters. The model was trained for 10 epochs with a batch size of 16, balancing computational requirements and model convergence. Sparse Categorical Accuracy was monitored on the validation set after each epoch to evaluate generalization. Mixed precision training was employed, as said before, enabling efficient scaling.

Callbacks and Monitoring

To ensure an adaptive and efficient training process, several callbacks were used. Early stopping was employed to halt training if the validation loss did not improve for 3 consecutive epochs, restoring the best model weights. Model checkpoints were saved after each epoch based on validation loss, ensuring that the best-performing model could be restored easily. Additionally, a `ReduceLROnPlateau` callback dynamically halved the learning rate if validation accuracy plateaued for 2 consecutive epochs. TensorBoard was integrated to log training and validation metrics, providing visual insights into model performance over epochs.

Overall, the final training process demonstrated consistent improvements over epochs, monitored through TensorBoard visualizations. Early stopping ensured optimal generalization without overfitting, while the saved model checkpoints allowed easy restoration and evaluation of the best-performing model.

4.3.4 Training Results

There was a lot of issues when training this model, the biggest one being overfitting. While the training loss and training accuracy steadily improves this is not the case for the validation metrics. Below in figure 6 it can be seen how the loss and accuracy quickly increases.

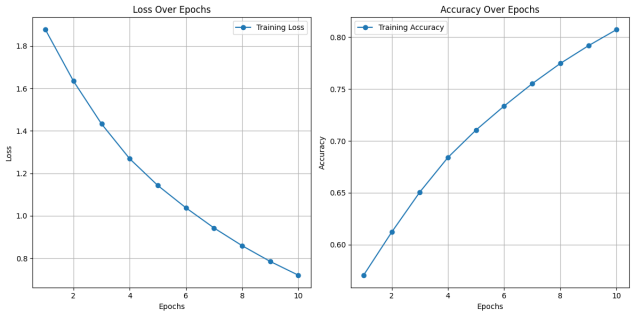


Figure 6: Training loss and accuracy

However, at the same time as the training metrics looks good the validation accuracy steadily drops and the validation loss keeps increasing which leads to worse results. However, after tweaking the model and implementing early stopping the following training trajectory was achieved as shown in figure 7 below.

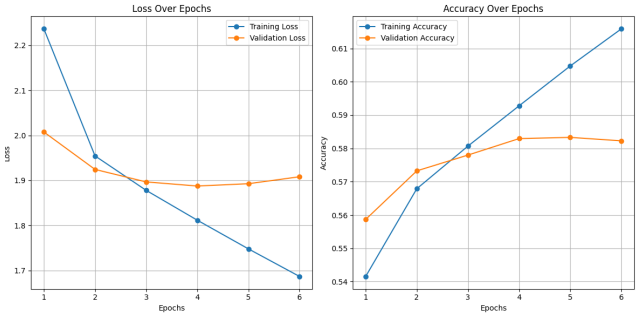


Figure 7: Training and Validation Loss and Accuracy

Below in figure 8 is another example of another configuration with a lower learning rate that also ended up overfitting. We address this result in details in here

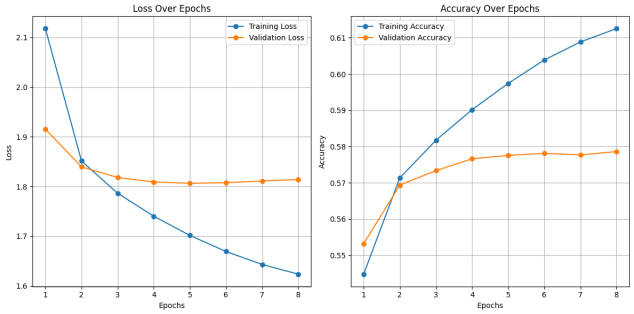


Figure 8: Training and Validation Loss and Accuracy

4.4 Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) is the process of optimizing the outputs of a LLM by utilizing a knowledge base outside of the models training data before generating the response to a prompt according to [4]. LLMs are already trained on large volumes of data and the goal of using RAG is to extend the capabilities of the model to specific domains without the need to retrain the model[4]. It can be expensive to retrain these large models so RAG offers an alternative. According to [4], it is a cost-effective approach to improve the responses of a LLM so that is is relevant and accurate in various contexts.

Because of this it was decided that it could potentially be a good fit for this project where Gemma 2 was to be optimized for Swedish language and culture. According to [4], RAG gives greater control over the generated text output and it becomes easier to understand why the model gave the response that it did since it is possible to look at the context retrieved.

One example from using the Gemma 2 model before fine-tuning and RAG shows how it hallucinates and generates false outputs which further motivates the use of RAG. It is shown below

```
Prompt: Vem är Sveriges statsminister?
Prediction:
Vem är Sveriges statsminister?

Sverige har en parlamentarisk demokrati med en statliga president som är
den högsta ledaren i landet. Presidenten utser också statsminister.

Presidenten utser statsminister från den största partiet i riksdagen.

Statsministerns roll är att leda regeringen och att utföra statliga
uppdrag.


Expected Response:
Vem är Sveriges statsminister?

Sveriges nuvarande statsminister är Ulf Kristersson, som tillträdde 2022.
```

In the example above the Gemma 2 model generates a response that says Sweden is a Republic, which is not the case since it is a Monarchy without a president. It clearly needs more information about Sweden. It also does not answer the question about who is the prime minister and is quite bad at Swedish grammar, which will be tackled with fine tuning.

The RAG process involves three primary steps that are explained in [14].

Context Retrieval

Given an input prompt q, the program should retrieve k relevant documents or chunks or sentences $\{d_1, d_2, \ldots, d_k\}$ from a base of knowledge D. This process can be expressed as:

$$\{d_1, d_2, \ldots, d_k\} = \arg\max_{d \in D} \text{Score}(q, d),$$

where $\text{Score}(q, d)$ is the measure of similarity used in the project. The similarity in this case is between the embeddings of the query q and a document/chunk/sentence d, depending on what type of data is used.

Combination with original prompt

The retrieved context is concatenated with the original input query or prompt q to form a new query with the context and the original prompt c. The combined context and original prompt is then sent to the LLM:

$$c = \text{concat}(q, d_1, d_2, \ldots, d_k).$$

Response Generation

The LLM then generates a response r that utilizes the combined context c:

$$r = \text{LLM}(c).$$

In this project, the similarity score Score(q, d) that was used was cosine similarity between the query and the embeddings.

The final response r is in other words influenced both by the quality of the context that was retrieved and by the LLM's ability to give good responses and interpret the combined query or prompt c.

In this project, RAG was implemented using the dataset that was sourced as described in section *4.1.2*. Then the method of generating the embeddings was not as straightforward. Two approaches were tested, using a pretrained model to generated the embeddings and training a new model using a framework like FastText.

4.4.1 FastText Model

According to [8], FastText is an open-source, free and lightweight library that can be used to learn text representations. It is included in gensim[19] and can easily be used to train a model that can generate embeddings as shown below. More information about the FastText implementation in Gensim can be found at [26].

```
model = FastText(flattened_sentences,
    vector_size=200,
    window=7,
    min_count=1,
    workers=4,
    sg=1,
    epochs=10)

model.save("fasttext_model.model")
```

Below is an example of this model after it had been trained on data from the RAG dataset. It is important to note that in order for it to work well. The data has to be properly processed an tokenized beforehand as explained in section *4.1.2*.

Prompt:
Hur ser Sveriges kultur ut? Svara med max 10 meningar.

Context as a small list of sentences:

['Sveriges rikes lag.', 'En studie av materiell kultur och ekonomi hos Sveriges första fångstfolk.', 'Ordet neutralitet försvann från Sveriges säkerhetspolitiska linje 2007.', 'Amerikansk kultur har sedan länge ett stort inflytande.', 'Svensk kultur är vidare starkt individualistisk och antinationalistisk, även självk ritisk.', 'Sveriges statsminister Ulf Kristersson.', 'Leve Sverige!', 'Till Sverige kom relikvariet 1632 som krigsbyte under trettioåriga kriget.', '[138] Svensk kultur är starkt egalitär och mycket öppen mot omvärlden.', 'Istället handlade den om Sverige idag: Sverige är ett demokratiskt land med fred, frihet och respekt för mänskliga rättigheter, oavsett religion eller ursprung.']

The prompt translates to "What does the culture of Sweden look like? Answer with at most 10 sentences." and it can be seen that many relevant sentences was retrieved using this FastText model. For example:

`'Amerikansk kultur har sedan länge ett stort inflytande.'` roughly translates to `'American culture has since long ago had a large influence.'`.

`'Svensk kultur är starkt egalitär och mycket öppen mot omvärlden.'` roughly translates to `'Swedish culture is strongly egalitarian and very open to the outside world.'`.

`'Svensk kultur är vidare starkt individualistisk och antinationalistisk, även självk ritisk.'` roughly translates to `'Swedish culture is furthermore strongly individualistic and anti-nationalistic, as well as self-critical.'`.

These are all good examples but there are also situations where the model does not generate good outputs at all. Therefore a pretrained model was also experimented with. Due to the nature of this project where the goal is to focus on underrepresented languages this could be considered a bad idea since not all languages have access to pretrained models. However, since it has been shown that FastText or similar models can be trained, it was decided that exploring the use of pretrained models

was a good idea to see what performance the RAG could generate with a model trained on more data. In order to make the FastText model more feasible more data would have to be used to train it.

4.4.2 Pretrained Sentence-BERT

The pretrained model used for this can be initialized in python using the following code:

```
model = SentenceTransformer("KBLab/sentence-bert-swedish-cased")
```

The pretrained model did, as expected, give more interesting results especially since it utilized chunking to not retrieve individual sentences but entire chunks. Below is an example when giving the following sentence to the embeddings model: "Hur ser Sveriges kultur ut?".

Chunk:

> Tätorter från 31 december 2020, kommuner från 30 september 2024) Alfred Nobel, instiftare av Nobelpriset. Muslimerna uppskattas 5 procent, eller 25 000, vara praktiserande (i betydelsen att de deltar i fredagsbönen och ber fem gånger om dagen). Det finns även buddhister, judar, hinduer och bahá'íer i Sverige. Bland de övriga finns enstaka som praktiserar modern asatro och traditionell samisk religion. Svensk kultur är en del av de nordiska, germanska och västerländska kulturområdena. Svenska kulturuttryck inom konst, musik och litteratur ansluter sig främst till dessa traditioner. Det offentliga stödet till kulturen är omfattande i Sverige. Folkligt deltagande är stort inom många kulturverksamheter, till exempel körsång, som engagerar tiotusentals svenskar. Värderingsmässigt skiljer sig svensk kultur starkt från världens genomsnittsvärderingar genom att vara mycket mer universalistisk, sekulär och inriktad mot postmaterialistiskt självförverkligande. Svensk kultur är starkt egalitär och mycket öppen mot omvärlden. Amerikansk kultur har sedan länge ett stort inflytande. Svensk kultur är vidare starkt individualistisk och antinationalistisk, även självkritisk. Maximal jämställdhet mellan kvinnor och män har ett centralt värde.

Similarity Score: 0.7748

Metadata:

- **Title:** Sverige – Wikipedia
- **Document ID:** 1_chunk54
- **File Path:** `data_preprocessing/data/wikipedia/Sverige.pdf`
- **Subdirectory:** `data_preprocessing/data/wikipedia`

It can clearly be seen that this response is more interesting. By using chunks of data more information will also be included and with this new approach such as which document the information came from.

4.4.3 RAG Architecture

This section will cover how the RAG mechanic was integrated with the Gemma 2 model in the notebooks used for this project. First of all when using the FastText model the problems with the model being trained on not enough data became painfully apparent. When giving the embeddings model a random question the embeddings model will not give back a good context, instead it will just give back other questions which is not helpful. Therefore the following method for utilizing it was implemented as shown below in figure 9.

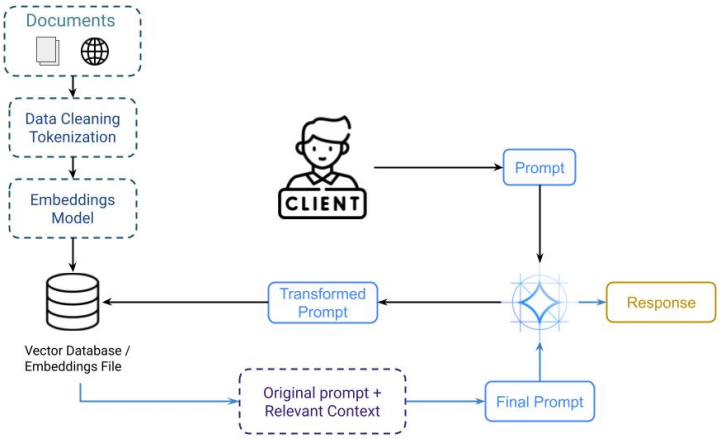


Figure 9: Rag Architecture using FastText

First of all, as can be seen in figure 9, the dataset is used to train an embeddings model and to generate embeddings stored in a Vector Database or an Embeddings File. In order to work around the problem with the FastText model trained on limited data, an alternative pipeline to the client sending the prompt directly to the model was implemented. How it works is that when the client sends a prompt the Gemma 2 model is first asked to transform the prompt into a simple statement that fits well with the embeddings and embeddings model. It is done with the following instructions:

```
context_prompt = f"Översätt inte detta.
                   Gör bara om det från en fråga till ett statement:
                   '{og_prompt}'. Answer with at most 4 words."
```

This basically tells the model to not translate the prompt from Swedish to english which was a big problem that was encountered. It also tells the model to rewrite the prompt from a question into a statement and to answer with at most 4 words. That transformed prompt, that is not a question, is then used to retrieve context from the embeddings file or vector database and the context is combined with the original prompt to get the final prompt. This final prompt is then given to the Gemma 2 model that generates the response. This method proved to be quite successful. One example is described below, without using this approach and passing the prompt directly to the embeddings and using the prompt: "Hur ser Sveriges kultur ut? Svara med max 10 meningar." ("What does the culture of Sweden look like? Answer with at most 10 sentences."). While some of the context is alright for this specific prompt, there are also questions in the retrieved context that have nothing to do with the desired answer such as "Hur firas dagen?" ("How is the day celebrated?"). By first passing it to the Gemma model to string sent to the embeddings for retrieval is "Sveriges kultur" which improves the context retrieved significantly.

In the case where a pretrained model is used for generating embeddings the process is simplified since there is no need to pass the prompt to the model twice and transform it to get good responses from the model. The method of using it in that case is shown below in figure 10.

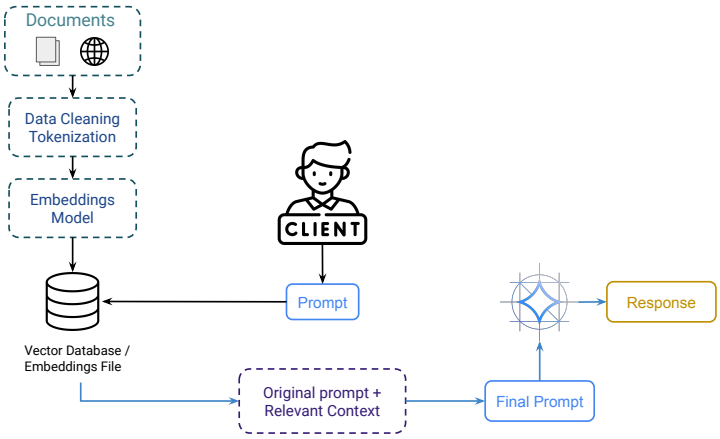


Figure 10: Rag Architecture using Sentence-BERT

The pretrained model is trained on more data and better att identifying similarity between text which is why the results are much better. It is therefore not necessary to send the prompt and have Gemma 2 transform it, instead it can be sent directly to retrieve context.

5 Results and Evaluation

5.1 Evaluation Metrics

The fine-tuned GemmaCausalLM model was evaluated using multiple metrics to assess key performance aspects: accuracy, linguistic quality, and semantic fidelity. These metrics were selected to align with common LLM tasks, including question answering, summarization, and machine translation.

5.1.1 Quality of Question Answering and Span-Based Tasks

For question answering and span-based tasks, we used **Exact Match (EM)** and **F1 score**. EM measured the percentage of predictions that perfectly matched the ground truth, ensuring strict correctness. F1 provided a balanced measure of precision and recall, allowing partial matches to be evaluated comprehensively. Together, these metrics assessed both the precision and completeness of the model's outputs.

5.1.2 Quality of Summarization

To evaluate summarization tasks, we employed **ROUGE** metrics, which measured how well the generated summaries preserved the content and structure of the reference texts. By comparing sequences of words and phrases, ROUGE provided insights into the model's ability to retain key information while maintaining coherence.

5.1.3 Quality of Translation and Text Generation

For translation and text generation, we used BLEU, METEOR, BERTScore, and COMET:

- **BLEU** measured the overlap of n-grams between the generated text and reference, emphasizing fluency and adequacy.
- **METEOR** extended BLEU by incorporating synonyms, stemming, and paraphrasing into its evaluation, providing a more nuanced measure of linguistic quality and alignment with reference texts.
- **BERTScore** used contextual embeddings to evaluate semantic similarity, ensuring the generated text preserved meaning, even when rephrased.
- **COMET** focused on cross-lingual translation, assessing how well the model maintained adequacy and fluency in multilingual contexts.

5.1.4 Comprehensive Evaluation Across Tasks

For question answering, EM and F1 scores captured accuracy and completeness. In summarization tasks, ROUGE measured content preservation and coherence. For translation and text generation, BLEU and METEOR assessed fluency and adequacy, while BERTScore and COMET ensured semantic and cross-lingual fidelity. This comprehensive evaluation framework ensured that the fine-tuned model was rigorously assessed across a diverse range of tasks, balancing strict accuracy with semantic and linguistic quality.

5.2 Results

5.2.1 Results for Question Answering and Span-Based Tasks

The model's performance on the question answering and span-based tasks is summarized in Table 2. The EM and F1 Score demonstrate the model's ability to generate accurate and complete answers.

Metric	Pretrained Model	Fine-Tuned Model	Fine-Tuned Model with RAG
EM	0%	0%	0%
F1	64.98%	47.72%	77.63%

Table 2: Question Answering Results

It is worth mentioning that the EM and F1 metrics, while widely used for evaluating extractive QA models, may not be the most suitable when prioritizing creativity and diversity. These metrics mainly focus on how closely the answers match the reference text and how accurate they are at a surface level, potentially undervaluing responses that are semantically correct but phrased differently or reflect nuanced reasoning. Additionally, achieving high scores with these metrics often depends significantly on the specificity of the prompt, as well-crafted, precise prompts guide the model toward extracting the exact spans required for evaluation. Alternative metrics that better capture semantic similarity and contextual relevance may be more appropriate for other QA tasks where flexibility is valued.

5.2.2 Results for Summarization

For summarization, ROUGE metrics were used to evaluate the quality of the generated summaries. Table 3 shows the scores for ROUGE metrics, indicating the model's effectiveness in retaining key information and producing coherent summaries.

Metric	Pretrained Model	Fine-Tuned Model
ROUGE-1	0.69	0.76
ROUGE-2	0.64	0.71
ROUGE-L	0.68	0.74

Table 3: Summarization Results

5.2.3 Results for Translation and Text Generation

The model's performance in translation and text generation tasks was evaluated using BLEU, METEOR, BERTScore, and COMET. The results in Table 4 highlight its fluency, semantic fidelity, and multilingual adequacy.

Metric	Pretrained Model	Fine-Tuned Model
BLEU	0.29	0.46
METEOR	0.82	0.51
BERTScore	0.77	0.76
COMET	0.60	0.72

Table 4: Translation Results

5.2.4 Interpretation and conclusion

The results demonstrate strong performance across a lot of task, particularly in translation, where the model achieved high BLEU and BERTScore values, indicating fluency and semantic fidelity. The lower METEOR score observed for the fine-tuned model may suggest that the model has overfitted to the fine-tuning dataset. This overfitting implies that while the model performs well on the specific data it was trained on, it struggles to generalize effectively to unseen examples.

The model also demonstrated good performance in question answering, as highlighted by a higher F1 score. However, the EM score is always 0, indicating no perfect alignment between predicted and ground-truth answers. While this result means that there is no precise textual alignment, it is less critical in certain applications compared to semantic accuracy. While summarization scores were satisfactory, there is potential for improvement in capturing nuanced information.

Quality of datasets The effectiveness of fine-tuning a model is directly influenced by the quality of the dataset used. In this project, the dataset of crafted prompt-response pairs played a crucial role in determining the performance of the fine-tuned model. However, the results, including lower METEOR and F1 scores, suggest that the dataset may lack the diversity and comprehensiveness needed for broader generalization.

The dataset appears to be too narrow, limiting the model's ability to handle prompts and responses that deviate from its training data. This lack of variability impacts the model's flexibility and hinders its potential to generate contextually relevant and semantically rich responses.

We address these limitations more in Section 6 and Section 7.

5.2.5 Subjective evaluation of the result

The quality of a response generated from a prompt can be a very subjective thing. Given two responses and two observers, the observers might prefer different responses. From this idea it could be argued that using subjective evaluation methods are relevant. In this project, the subjective evaluation was carried out by the group members with a Swedish background. From the observations made and with the knowledge the evaluators posses as native Swedish speakers, the methods proposed in this project do improve the responses.

As an example using the FastText model that was trained specifically for this project. To the question: "Hur ser Sveriges kultur ut? Svara med max 10 meningar.", the model gave a pretty uninteresting and not representative answer according to the group. After fine tuning and using RAG the answer was subjectively much more interesting and accurate. The before and after response is shown below:

Before:

> Sverige har en rik kulturhistoria med många olika traditioner och influenser.
>
> **Folkmusik:** Sverige har en stark tradition av folkmusik, med instrument som violin, accordion och fiol.
>
> **Konst:** Sverige har en lång tradition av konst, med många framstående konstnärer som Carl Larsson, Edvard Munch och Anders Zorn.
>
> **Litteratur:** Sverige har en rik litteraturhistoria, med många framstående författare som Astrid Lindgren, Selma Lagerlöf och Stieg Larsson.
>
> **Arkitektur:** Sverige har en unik arkitektur, med många vackra byggnader från olika epoker.
>
> **Mat:** Sverige har en rik matkultur, med många traditionella rätter och delikatesser.
>
> **Festivaler:** Sverige har många populära festivaler, som Melodifestivalen och Stockholm Pride.
>
> **Natur:** Sverige har en fantastisk natur, med många vackra skogar, sjöar och öar.
>
> **Idrott:** Sverige har en stark idrottskultur, med många framstående idrottsmän.

After:

> Sveriges kultur är en komplex och mångfacetterad konstruktion som präglas av en rad olika faktorer. Den är starkt influerad av svensk historia, med en lång tradition av folkmusik, konst och litteratur. Sverige har en stark tradition av egalitär och öppenhet, vilket syns i dess starka socialtjänst och välfärdssystem. Sveriges kultur är också starkt individualistisk och antinationalistisk, vilket gör det till en unik och dynamisk kultur. Sveriges kultur är också starkt präglad av den svenska naturen, med många vackra parker och skogar. Sveriges kultur är också starkt influerad av den svenska traditionen av folkmusik, konst och litteratur.

Not only is the format more what would be expected. The model is able to incorporate information from external sources using RAG which can be seen in how the new response captures that Sweden is a individualistic and anti-nationalistic country with a strong welfare system.

The context provided by the pretrained embeddings model was even better which shows the potential of this approach.

6 Discussion, Challenges, and Solutions

This section will cover some discussion about the implementation of the different aspects of this project as well as how things could have been done differently.

6.1 The impact of the constraints on this project

An interesting point during one lecture was that the relevance of the work increases when working with a larger model. This is because even with all the optimization that is done to a small model, the much larger model is probably going to be better still. Furthermore, it is possible that the techniques used to improve a tiny model does not scale when working with a larger model. Due to the financial and computational constraints of this project, the model used was a small version of Gemma 2 which is for these reasons not ideal. If a larger model would have been used, the chances are that it would perform better and that other methods for improving performance could have been used. It is also a possibility that the same methods could still be used.

6.2 Difficulty in comparing models

The responses generated by a model are inherently evaluated by the individual who wrote the prompt. Since human perspectives vary, this evaluation is subjective. As previously mentioned, Reinforcement Learning from Human Feedback can be implemented to improve results by incorporating subjective measures. This allows the model to generate responses tailored to user preferences, such as text length, structure, and language style.

The evaluation of the models in this project was also carried out on a set of test prompts and answers which means that it is possible that it would perform worse on a different set of tests. To get a fair comparison to other models, they would have to be tested on the same test cases.

6.3 The impact of future computing costs

One of the reasons for working with RAG, as pointed out in [4] and section *4.4*, was the cost of retraining large models. As RAG can be a cost effective way of improve the responses of a LLM it was seen as a good choice since the people working to fine tune a LLM for their domain might not have access to a lot of resources.

However, as Nvidia CEO Jensen Huang says in [24], the cost of the computation for AI will go down and if that is the future, which could likely be the case, the relevance of using RAG might go down. If the cost of compute goes down faster than the amount of data available then in the future it might not be necessary to utilize RAG. But, as the cost of compute decreases other bottlenecks might appear such as the quality of the data the models are trained on and how the data is curated. To use RAG to instead utilize external data in a more dynamic way might be valuable even if the cost of compute goes down. Another strength of RAG is the explainability of the results. It becomes easier to see why the model gave a response and where the data came from, which is crucial for building trust in AI systems and ensuring their outputs are reliable and transparent. A topic highly relevant right now.

6.4 RAG and Fine Tuning

As can be seen in figure 1, there seems to be logical reasons as to why doing both RAG and Fine Tuning can be a good idea. As shown in section *4.4*, the base pretrained model performs poorly in Swedish. The language is off and it does not know the facts about Sweden. Therefore doing both was a natural choice since there was a need for external knowledge and adaptation of the model to the language.

More work is needed both for fine-tuning and RAG, especially when it comes to the datasets. For fine-tuning, we ultimately decided to use 500 prompts and responses. While this may be slightly fewer than what others have used online, this approach combined with RAG still produced good results. However, if better fine-tuning performance is desired, a significantly larger dataset would be required. A larger dataset would allow the model to learn a more comprehensive representation of the target language and domain-specific knowledge, ultimately improving its ability to generate accurate and contextually appropriate responses. And while a lot of text was used for RAG, to keep a model that is up to date and accurate a more comprehensive dataset would have to be crafted.

7 Conclusion and Future Work

7.1 Conclusion

In summary, this project has demonstrated that by combining LoRA and RAG, it is feasible to adapt Gemma 2 for an underrepresented language like Swedish. The results show improvements in the tasks question answering, summarization, and translation tasks. RAG proved particularly effective in dynamically supplying the model with up to date information and reducing off-topic responses.

However, there is a lot of room for improvement. With more time and testing more configurations for the training perhaps the performance could have been greater. If the struggles with overfitting, resource limitations and small size of the datasets was fixed. The result could have been much better.

7.2 Future Work

The dataset used for fine tuning and RAG in this project are relatively small. In order to achieve better results larger datasets that are more carefully crafted could be utilized to better capture complex nuances in the language that is being worked on.

Furthermore, it was shown that a pretrained embeddings model was much better at capturing semantic relationships and it begs the question if there is room for further improvement. By using vast amounts of data in the target language to create a better embeddings model the performance could be increased.

Another interesting topic to explore would be to do reinforcement learning with human feedback. This is something that [9] did when fine-tuning Gemma 2 for Japanese. This would allow the model to become better at understanding what constitutes a good and relevant response which could increase the performance.

Moreover, the evaluation of the model was carried out on a relatively small number of prompt-response pairs and only two people who served as the judges for the relevance of the responses. When completing a scaled up version of this project, involving more judges and a more structured way to evaluate the responses in comparison to other models would be a good idea.

By improving on these suggestions, Gemma 2 could increasingly serve diverse cultural contexts and preserve global language heritage.

On a personal note, the group is satisfied with the results of this project and have learned a lot.

8 References

References

[1] Lt4all 2025 overview. https://www.lt4all2025.eu/overview/. Accessed: 2024-12-30.

[2] Meta AI. No language left behind (nllb). https://ai.facebook.com/research/no-language-left-behind/, 2022. Accessed: 2024-12-30.

[3] AI Sweden. Gpt-sw3, 2024. URL https://www.ai.se/en/project/gpt-sw3.

[4] Amazon Web Services. What is retrieval-augmented generation? https://aws.amazon.com/what-is/retrieval-augmented-generation/, 2024. Accessed: 2024-12-29.

[5] Anyscale. Fine-tuning llms: Lora or full-parameter? an in-depth analysis with llama 2. https://www.anyscale.com/blog/fine-tuning-llms-lora-or-full-parameter-an-in-depth-analysis-with-llama-2, 2024. Accessed: 2024-12-29.

[6] BigScience. Bloom: Bigscience large open-science open-access multilingual language model. https://bigscience.huggingface.co, 2022. Accessed: 2024-12-30.

[7] European Commission. etranslation platform. https://ec.europa.eu/info/resources-partners/machine-translation-public-administrations-etranslation, 2022. Accessed: 2024-12-30.

[8] Facebook AI Research. Fasttext. https://fasttext.cc, 2024. Accessed: 2024-12-29.

[9] Google. Gemma 2 japanese release, 2024. URL https://huggingface.co/collections/google/gemma-2-jpn-release-66f5d3337fdf061dff76a4f1.

[10] Google AI. Spoken language task-specific tuning with gemma, 2024. URL https://ai.google.dev/gemma/docs/spoken-language/task-specific-tuning.

[11] Google Developers. Fine-tuning gemma 2 with keras: Hugging face update. https://developers.googleblog.com/en/fine-tuning-gemma-2-with-keras-hugging-face-update/, 2024. Accessed: 2024-12-29.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2302.06675*, 2023. URL https://arxiv.org/pdf/2302.06675.pdf. Accessed: 2024-12-29.

[13] Kaggle. Gemma language tuning competition, 2024. URL https://www.kaggle.com/competitions/gemma-language-tuning.

[14] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020. URL https://arxiv.org/abs/2005.11401.

[15] Litteraturbanken. Litteraturbanken: Free swedish literature in digital format, 2024. URL https://litteraturbanken.se/epub?visa=pdf&sort=popularitet.

[16] Theo M. A very short visual introduction to learning rate schedulers (with code). https://medium.com/@theom/a-very-short-visual-introduction-to-learning-rate-schedulers-with-code-189eddffdb00, 2024. Accessed: 2024-12-29.

[17] ML Spring. Prompting vs. rags vs. finetuning. https://mlspring.beehiiv.com/p/prompting-vs-rags-vs-finetuning, 2024. Accessed: 2024-12-29.

[18] OpenAI. Chatgpt: Gpt-4 language model, 2024. URL https://chatgpt.com/g/g-sxq7w7XDi-gpt-chat-svenska. Accessed: 2024-12-30.

[19] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[20] Claudia Soria et al. The dominance of english-trained ai systems and the risk of digital language death: Preserving linguistic diversity for cultural identity and knowledge transmission. *Publications of the National Research Council of Italy*, 2020. URL https://publications.cnr.it/api/v1/documents/download/171362. Accessed: 2024-12-30.

[21] Swedish Wikipedia. Portal: Huvudsida. https://sv.wikipedia.org/wiki/Portal:Huvudsida, 2024. Accessed: 2024-12-29.

[22] Federico Torrielli. Stars, stripes, and silicon: Unravelling the chatgpt's all-american, monochrome, cis-centric bias. Technical report, University of Torino, 2024. URL https://arxiv.org/pdf/2410.13868.

[23] World Economic Forum. Generative ai and language bias: A need for inclusion, 2024. URL https://www.weforum.org/agenda/2024/05/generative-ai-languages-llm/.

[24] Yahoo Finance. Nvidia ceo says reasoning ai is the next step in artificial intelligence evolution. https://finance.yahoo.com/news/nvidia-ceo-says-reasoning-ai-100000432.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cu\protect\penalty-\@MZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAHisrHx3ayh5Kerhtl4Cu6IXAAPbvxmFC6bhxPbLqPD\protect\penalty-\@MXf8uiQCEgAnn0fy5zUq9Q1pZHso6bnYvet2jxJMxFGgkVqoQYx7m1VhZtBYsZ180E_B0KKUORwPbbbqczm4M3eqs-d1efARmqbYfyt404qbBVp-ZhORusfI3Ciz-Yobpa, 2024. Accessed: 2024-12-30.

[25] Vithya Yogarajan, Gillian Dobbie, Te Taka Keegan, and Rostam J. Neuwirth. Tackling bias in pre-trained language models: Current trends and under-represented societies, 2023. URL https://arxiv.org/abs/2312.01509.

[26] Radim Řehůřek. Gensim fasttext implementation. https://github.com/piskvorky/gensim/blob/develop/gensim/models/fasttext.py, 2024. Accessed: 2024-12-29.