UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

Usos y aplicaciones de modelos de lenguaje masivos en español

Sebastián David Mena Guitarra

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en Ciencias de la Computación

Quito, 03 de octubre de 2023

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Usos y aplicaciones de modelos de lenguaje masivos en español

Sebastián David Mena Guitarra

Nombre del profesor, Título académico **Daniel Riofrío, Coordinador**

Quito, 03 de octubre de 2023

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Sebastián David Mena Guitarra

Código: 00212779

Cédula de identidad: 1724373046

Lugar y fecha: Quito, 03 de octubre de 2023

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

La evolución de la inteligencia artificial ha abierta nuevas posibilidades dentro del Deep Learning como, por ejemplo, procesamiento de lenguaje natural (NLP). Una aplicación de NLP son los modelos masivos de lenguaje (LLM), los cuales han surgido como la última tecnología en tareas como generación de texto, traducción y búsqueda de respuestas. Sin embargo, los LLM están en una etapa muy temprana de su existencia. En este artículo, se usarán tres de las arquitecturas más recientes y potentes de LLM para estudiar y entender el funcionamiento de los modelos, las aplicaciones y las limitaciones que tienen al momento de trabajar con los modelos de manera local en una computadora local de gama de consumidor.

Palabras clave: modelo de grande lenguaje, fine-tuning, deep learning, transformer architecture, procesamiento de lenguaje natural, preentrenamiento, generación de texto.

ABSTRACT

The evolution of artificial intelligence has opened new possibilities within Deep Learning, such as Natural Language Processing (NLP). One application of NLP is large language models (LLM), which have emerged as the latest technology in tasks such as text generation, translation, and question answering. However, LLMs are in a very early stage of their existence. In this article, three of the most recent and powerful LLM models will be used to study and understand the functioning of the models, their applications, and the limitations they have when working with them locally on a high-spec consumer product line computer.

Palabras clave: large language model, fine-tuning, deep learning, transformer architecture, natural language processing, pretraining, text generation.

Table of Content

Table Index

Figure Index

INTRODUCTION

Artificial Intelligence has revolutionized the world in the last few years and will continue to do so for a long time (Kapoor, 2021). This statement comes alongside two main factors: enormous amounts of data that can be accessed via the Internet, and GPU development (Mahapatra, 2018). The concept of deep learning has existed since 1950, but it could not be fully implemented in real life because of the lack of computing power needed to run the algorithms (Dettmers, 2015). However, that has changed in recent years with a boom in Graphics Processing Units (GPU) development and now the world is experiencing new breakthrough AI technologies every few months.

One of these revitalized possibilities is Natural Language Processing (NLP). It shared the same obstacles as a rising new concept in 1950 because of the absence of computing power necessary to bring it to the real world. In the 2010s, NLP was studied with models such as Support Machine Vectors and Hidden Markov Chain (Nadkarni, Ohno-Machado, & Chapman, 2011). Nowadays, the most prominent model architectures to tackle NLP tasks are neural networks, especially ones that use encoder-decoder architecture and transformers architecture, both of which are deep learning models (DeepLearning.AI, 2023). Large Language Models (LLM) are a result of the latter and have expanded the horizons of what was considered possible in NLP (Wei, et al., 2022).

Large language models predict the next word given a sequence, but the output is not guaranteed to be correct. These models do not ‘think’ per se, they generate text based on patterns and information learned during training (Riedl, 2023). This is possible because of the aforementioned transformers architecture. Transformers models use encoders and decoders to manage big amounts of data, processing all words at once and decreasing training speeds and computational power needed relative to other types of neural networks (DeepLearning.AI,

2023). This also allows LLMs to manage more information and as such, produce better results when analyzing input text and generating output text. Nowadays, LLMs are trained with trillions of tokens (a token is a word in NLP) of information and operate with tens of billions of parameters. A parameter is a variable that the model uses to process, learn, and generate data. In neural network terms, a parameter is a weight. Usually, more parameters result in better results at the cost of more computational power (Microsoft, 2023). For example, Google's PaLM was trained on 3.5 trillion tokens and operates with 340 billion parameters.

These new massive LLMs have opened possibilities in NLP tasks. One of the most prominent is to create a chatbot, especially tailored to provide a customized customer experiences and answer complex questions (Lee, 2023). If trained correctly, LLMs can be adapted to be state-of-the-art chatbots such as ChatGPT-4, made with OpenAI's GPT model, and Bard, made with Google's Gemini model. These chatbots have transformed the way people interact with the internet and with knowledge in general and represent a shift in human-computer interactions (Alabbas, 2023).

As a final note, this investigation will follow the ISO/IEC 23053 standard. This standard defines various concepts and terminology in Artificial Intelligence related topics. This standard was selected because this project will cover a variety of AI-related topics such as neural networks and their architecture. As such, in order to be coherent with the majority of referenced papers and projects, in is important to define a standard such as this to ensure the comprehension of the project.

Neural Networks

All large language models are neural networks (Muehmel, 2023), so it is important to explore this topic first. “Artificial neural networks are quantitative models linking input and outputs adaptively learning in a learning process analogous to that used by a human brain” (Abdi, Valentin, & Edelman, 1999). To link input and output, a neural network is composed of neurons, each one being a mathematical function that calculates an output given an input (Muehmel, 2023). Neurons are connected to one another through weights. Weights feed neurons the input and then help the neuron propagate the computed values to the output neurons. Learning happens when the weights joining the neurons change and improve their output (Aggarwal, 2018).

In general, the goal of a neural network is to map n-dimensional real input to an m-dimensional real output (Rojas, 1996). To learn how to map which inputs to which outputs, a neural network needs to be fed input-output pairs of the function to be learned, this is called training data. After being fed the training data, the neural network tries to predict the output based on the input received. It then checks if the prediction was the same as the given output. The neural network adjusts the weights in response to prediction errors, with the goal of making future predictions more accurate. It can be said that the weights are being changed in a mathematically justified manner to reduce error (Aggarwal, 2018). This happens with every neuron in the neural network, effectively changing and improving the function in each one of them until the result is accurate enough. This is called backpropagation, an iterative process which informs previous neurons in the network what needs to change in order to change to achieve the best training results.

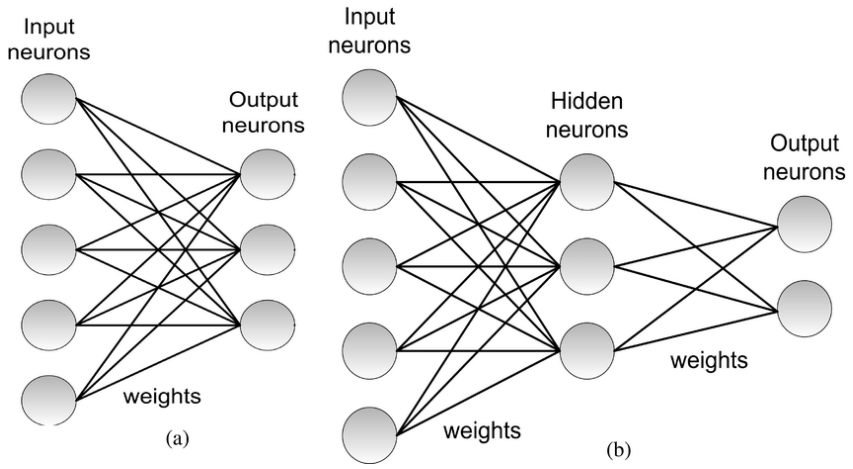

Figure 1: a) single layer neural network vs. b) multi-layer neural network

However, neurons are not just independent entities organized in a random fashion. Neurons are organized in layers and the number of layers in a network determine its complexity (Varma, 2020). A model with lesser complexity, single layer, may not be able to understand a complex training dataset, but a model with excessive complexity, let us say 100 layers, may overfit the training dataset and produce equally inaccurate results as the single layer model. The first layer of a neural network is called 'input layer' and the last layer is called 'output network'. All layers in between are called 'hidden layers' because its calculations are not visible to the user (Aggarwal, 2018). Layers can have different structures and different numbers of neurons in each one.

Transformers architecture

Neural networks are not really the end goal of deep learning, they are the founding block of many of its technologies instead. One such technology is the transformers architecture, which is a specific type of neural network designed to handle very large inputs and outputs with less computational power than recurrent neural networks. This architecture specializes in text generation, translation, and analysis because of its particular attention mechanism (Vaswani, et al., 2017).

Classic transformers use an encoder-decoder architecture. Simply put, the encoder receives an input sequence and then turns it into a vector. Then, it passes this vector to the decoder that will convert it into an output sequence. Both mechanisms work together to turn any given input into a coherent output, entirely created based on what the input asked and pre-trained data. However, large language models do not really use the 'classic' version of transformers, but a modification that has erases the encoders entirely. This new transformer is called 'decoder-only' and was first introduced in 2018 by Google Brain's researchers. This group of researchers concluded that decoder-only models were not only faster than encoder-decoder models but could also analyze long sequences of text with ease, even writing Wikipedia articles with state-of-the-art quality (Lui, et al., 2018). There are also encoder-only transformers, such as Google's BERT, which is better at classification tasks (Devlin, Chang, Kenton, & Toutanova, 2018). Nevertheless, this dichotomy between encoder-only and decoder-only transformers is misleading because, structurally, both architectures are the same, the only change is that decoder-only transformers are capable of recursion, meaning that they can access their own outputs from all previous steps (Roberts, 2023). Following Google's discoveries in transformer models, the decoder-only transformer stablished itself as the standard for generative NLP tasks, such as chatbots. Let us study how this architecture manages to generate text so effectively.

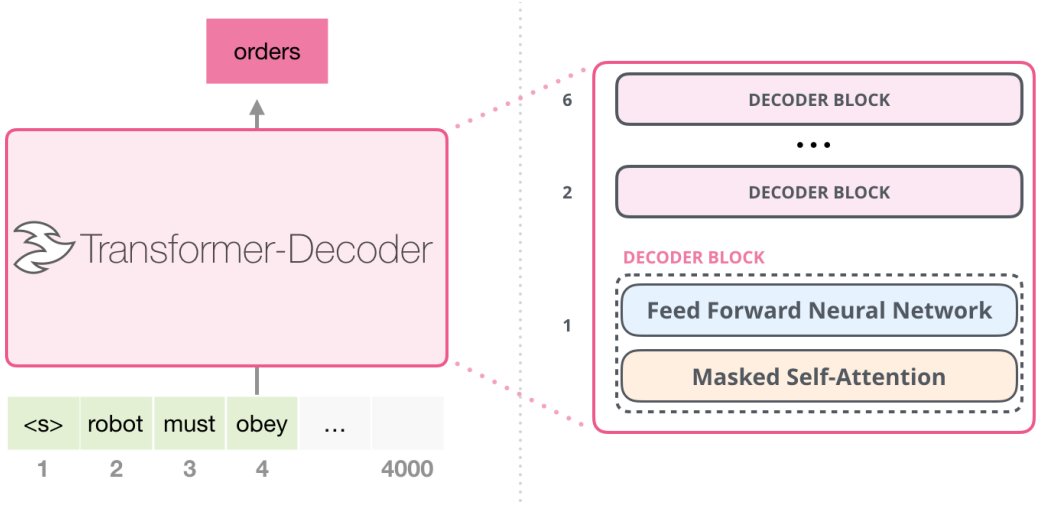

Figure 2: architecture of a decoder-only transformer model

The first important concept is 'self-attention'. This mechanism allows a model to capture relations between different positions in a sequence by 'attending' to all positions at the same time (Abideen, 2023). For example, let us look at this sentence: 'The dog will follow the command given to **it'**. People have no trouble understanding what the word 'it' is referring to. However, pre-transformers neural networks had trouble recognizing the relationship between 'dog' and 'it', even if both meant the same, just at different positions in the sequence. This is exactly what the attention mechanism does, it allows the model to understand relevant and associated words that explain context before feeding the word to the neural network (Alammar, 2019).

This is done in the 'Masked Self-Attention' component on the decoder block by a process called masked self-attention. This is a special type of self-attention in which the model processes one word at a time and not all the words at the same time, as is done in a normal self-attention mechanism. Masked self-attention is used alongside multi-head attention, which is a mechanism that performs multiple masked self-attention processes simultaneously, so that more relevant relations and context can be found in sequences (Sarkar, 2022). For example, in the sentence presented before, the word 'command' also refers to the dog,

because it is a word currently associated with them. Maybe a single masked self-attention process will not recognize this relation, but with multi-head attention, there is more probability that the model notices it.

Fine-tuning Large Language models

Pre-training a large language model is an extremely time consuming and expensive task. So as to train the first version of LLaMA, Meta used 2048 NVIDIA A100 GPUs, each one with 80GB of VRAM for about 21 days (Touvron, et al., 2023). This is simply not possible for a normal person or even a medium-sized technology company. Right now, only the biggest corporations in the world can develop and train a state-of-the-art large language model from the ground up. However, a paper in 2021 by Microsoft researchers presented a new technique called Low Rank Adaptation (LoRA) to finetune large language models and specialize them in specific tasks.

LoRA was inspired a concept discovered by Meta in 2020 that proved that, for LLMs, there exists a low dimension reparameterization that is as effective for finetuning as the full parameter space (Aghajanyan, Zettlemoyer, & Gupta). This means that it is possible for a smaller-size matrix to influence the bigger original matrices if done correctly. This research inspired the original LoRA paper by Hu et al. in 2021 that introduced a new way of finetuning LLMs without additional inference latency or added model size, as with other alternatives. The way LoRA achieves this is by freezing the original pre-trained weights to the model and injecting smaller low-rank trainable matrices into some of the layers of the Transformers architecture, reducing the number of trainable parameters. The smaller matrices are adaptations of the original much bigger matrices that were original to the model. These matrices are the ones that are trained with the new data and then injected back into the original model (Hu, et al., 2021).

In theory, LoRA could be applied to any of the layers of the LLM, but in the original paper only the attention layers are modified and not any of the MLP (multi-layer perceptron) layers (Hu, et al., 2021). Figure 3 shows a LoRA configuration that is targeting 3 of the 4 attention layers, those being ‘q’, ‘k’ and ‘v’.

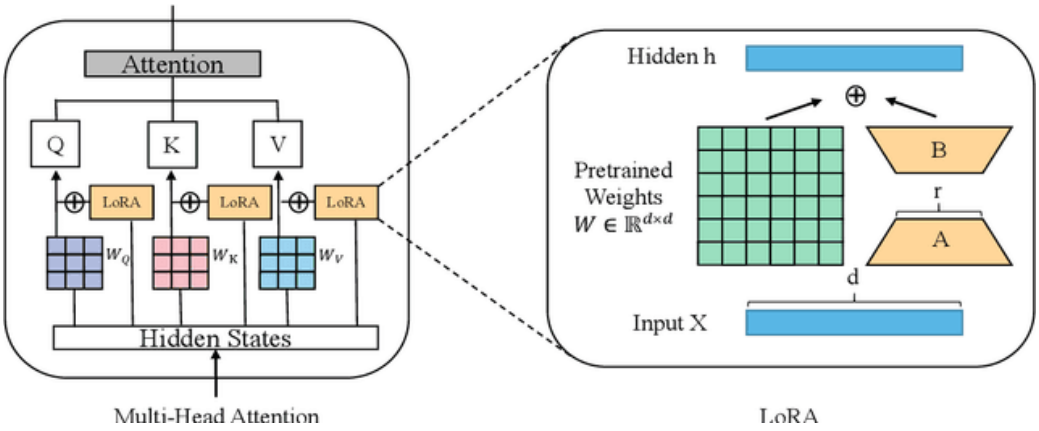

Figure 3: Low Rank Adaptation visualization

The main advantages of LoRA are the reduced hardware requirements needed to finetune a model. For example, it is possible to finetune ChatGPT-3, which has 175B parameters, with just 365GB of VRAM instead of the original 1.2TB that were needed to pre-train the model. (Hu, et al., 2021). LoRA is best suited to finetune a model to perform downstream tasks and is not a replacement for building new models. For example, changes in architecture such as Grouped Query Attention (GQA) available in Mistral cannot be implemented via LoRA.

Model size and quantization

As stated previously, large language models require a lot of computing power to work appropriately. For example, to run an LLM with 40 billion parameters, such as FALCON 40B, the computer should have enough GPU VRAM to load all 40 billion parameters, alongside a powerful enough CPU so that there are no bottlenecks. Combine this with the fact that, generally, the more parameters a model has, the better its performance and quality will

be (Johnson, 2023), and that means that there is a serious economic barrier regarding using large language models. Powerful GPUs are becoming increasingly costly, and as such, to choose a model to use, the hardware limitations must be considered. Before talking about which models were chosen, let us first talk about why large language models require so much VRAM.

Almost all parameters in a large language model share the same size. By default, parameters in the 'transformers' library have a size of 16 bits. Multiplying 2 bytes times the number of parameters a model has results in the VRAM needed to run that LLM locally. For the mentioned FALCON 40B, the calculation is done such as:

$$VRAM = \#\ of\ parameters\ \times memory\ per\ parameter$$

$$VRAM = 40\ \times 10^9 \times 2\ bytes$$

$$VRAM = 80GB$$

To have this much VRAM, multiple GPUs must be used. For example, 1 NVIDIAS A100 80GB, each one costing 15 000 USD. This creates a serious economic barrier to use large language models.

Quantization is a method developed for the sake of making large language model use and deployment easier. Quantization reduces the size of the parameters so that less VRAM is needed to load a model, so that instead of each parameter having a size of 4 bytes, each one has a size of 2 bytes or even 1 byte. The most common form of quantization is to transform floating point numbers with a size of 32 bits to integers of 8 bits, effectively reducing by approximately 4 times the amount of VRAM needed to load the model (Labonne, 2023).

There are multiple ways of quantizing a model, such as converting all the model to the specific size wanted, such as 4-bits or 6-bits per parameter, but this means transforming the

whole model which requires the use of external libraries to load the model, further complicating the use of the model. Instead, PyTorch, alongside the bits and bytes library, offer a function to load a model in 8-bits or 4-bits, without so much as changing the model but instead loading all weights with the specified size (Hugging Face, 2023).

The combination of Quantization and Low Rank Adaptation result in a new concept called Quantized Low Rank Adaptation (QLoRA), which is the combination of both technologies, resulting in the possibility to finetune and deploy LLMs on consumer level hardware (Dettmers, Pagnoni, & Holtzman, 2023).

Hardware setup

Before choosing models, it is first necessary to describe the specifications of the computer that is going to be used to develop this project.

- Computer model: Dell Precision Tower 7820
- CPU: Intel Xeon Silver 4216 @ 2.10GHz x 32
- GPU: NVIDIA A4000 16GB VRAM
- RAM: 45.7GB
- Disk: 512GB SSD
- Operating System: Ubuntu 20.04 LTS 64-bit
- CUDA Version: 12.1

Proposal Description

Given the hardware limitations and the new opportunities that Large Language Models open, the proposal of this project is to test the adaptability of small LLMs to follow instructions in another language that they were not directly pretrained on. This language is going to be Spanish, and the models are going to be tested on 2 different tasks. The first task refers to

Spanish instructions that contain general instructions such as creating a story or defining a certain word. The second task refers to domain-specific instructions in Spanish. These instructions will be specific, so much so that the base models might have never seen the content they refer to. To adapt the LLMs, the QLORA technique of finetuning is going to be used.

Chosen models

As stated before, in this project 3 different LLMs are going to be trained and analyzed. These 3 models are: LLaMA2 7B, FALCON 7B and Mistral 7B. All these models follow the theory of the architecture discussed above but have some variations between one another. It is important to emphasize that these are 3 completely different models, pre-trained from scratch, and not only finetuned versions of previous models.

LLaMA2 7B is a decoder-only large language model released by Meta in 2023. It was trained with 2 trillion tokens on several different languages but primarily English (Touvron, et al., 2023). This model is the most well documented open source LLM on 2023, with a lot of resources to finetune, pretrain and deploy its models. LLaMA2's main advantage is its Chat version that has been trained with Reinforcement Learning from Human Feedback. This is a process that takes enormous amounts of time because it needs many people to be effective. LLaMA2-Chat can achieve ChatGPT performance according to human evaluations, a feat that is remarkable for an open-source model (Schmid, Sanseviero, Cuenca, & Tunstall, 2023). While it is true that LLaMA2-chat was not trained on a significant amount of data in Spanish, due to transfer learning it is expected to perform above the other 2 models.

Mistral 7B is a decoder-based model released by MistralAI in 2023 as well. There is no precise information as to the number of tokens Mistral was trained on but according to early benchmarks, Mistral 7B can outperform even LLaMA2 13B and LLaMA1 34B (Jiang, et al.,

2023). Mistral also uses a Sliding Window Attention Mechanism implemented with FlashAttention and xFormers library that allows it to process longer sequences easier. Mistral also includes a Grouped-query attention for faster inference (Mitral AI, 2023). Mistral's main advantage is the implementation of various SOTA technologies in large language models.

FALCON 7B is a decoder-only model developed by the Technology Innovation Institute (TII) in Abu Dabhi, a research institution created in 2020. The TII had a different focus to Mistral AI and Meta when developing Falcon. Instead of focusing on new technologies such as Mistral, its focus was the careful creation and curation of its pretraining dataset. This dataset is called RefinedWeb and has 5 trillion high-quality tokens. According to zero-shot benchmarks, that is a benchmark that analyzes the answer of the model to a question it was never shown before, this high-quality dataset can outperform public and private models (Penedo, et al., 2023).

Datasets

As stated before, there are two main tasks at hand with the interest of completing this project. The first one is to fine-tune both versions of Llama to follow instructions in Spanish. There are two main publicly available datasets that are appropriate for this task. These are the Alpaca and Dolly datasets. The Alpaca dataset is a set of 52 000 instructions generated by Stanford using GPT-3 and curated by GitHub user gururise (Taori, et al., 2023). This dataset has been used by the LLM community to fine tune models to follow instructions in different languages with great results. The second dataset is Dolly, a dataset of 15 000 instructions created by DataBricks. This dataset is equally useful in training LLMs. There exist curated and translated versions of both datasets, which are going to be used. To improve results, both datasets have been joined to fine tune both models. This results in 67 000 examples of

instructions to follow in a JSON file, that is needed to train both models to follow instructions in Spanish.

The second task is to teach all three models about laws in Ecuador. There are no datasets available to do such a thing. As such, it is necessary to build the dataset from the ground up. The Tax Authority in Ecuador, Servicio de Rentas Internas (SRI), fortunately contains dozens of free resources to teach people about taxes and how to pay them. Alongside the official law, that is a 400-page file with all the details about tax laws, it is possible to build a dataset of a minimum of 1000 entries to fine tune all three models. Aiming to extract information from all these documents OpenAI's GPT-4 is going to be used. This model has one of the best problem-solving and information-extraction capabilities in the world (OpenAI, 2023). Alongside human double checking of most questions and appropriate corrections and additions in any answers.

The documents can be read uploaded to GPT-4, and there are no copyright restrictions around these public documents, so that is of no concern. After GPT-4 read the document, the following prompt is used:

This document is written on Spanish, so all answer you provide will be on Spanish. This document is about taxes law in Ecuador, for context. I do not need a summary of the document. Instead, I need question-answer pairs in order to build a dataset to train a large language model about this specific topic. Please double check all questions and answers. Answers should be easily read and understood by people. The questions should talk about daily problems that people of Ecuador could face. You answer will be provided in the following JSON format: {"instruction":" your question ", "input":" ", "output":" your answer " },

The Alpaca and Dolly datasets have been built using 3 categories, the instruction which basically contains the question to be asked, the input which is the variation of the question asked and the output which is basically the answer. The law dataset is going to follow a similar structure but will not include any content on the 'input' field because separating instruction and input will take a great amount of time and human supervision, something not within the reach of this project.

Finetuning configuration

The technique used to finetune the models is QLoRA on all 3 models because of the VRAM limitation at hardware level. All models will be loaded and tested using the Hugging Face 'transformers,' 'peft' and 'trl' libraries. Hugging Face has an organized and detailed documentation of the models and classes that are implemented. Quantization was implemented as shown in the tables below.

Parameter	Value used
Quantization	4 bits
Double Quantization	True
Quantization Data Type	nf4
Quantization Compute Data Type	bfloat16

Table 1. Quantization configuration

Hyperparameter	Value used
LoRA Rank	256
LoRA Alpha	256

Learning Rate	2e-4 (LLaMA2, FALCON), 2e-5 (Mistral)
Learning Scheduler	Cosine
LoRA dropout	0.20
Epochs	3
Optimizer	Paged Adam 8-bit
Weight decay	0.001
Warmup ratio	0.3

Table 2. Low Rank Adaptation Configuration

There are other parameters such as ‘batch size’ but those parameters were changed from model to model to be able to run on the 16GB of VRAM available. All models were trained using the same method. Additionally, the finetuning of all models was targeted specifically to the attention head of the models and not the underlying and bigger neural network over which the attention heads are based on. LLaMA2 and Mistral have similar structures in terms of the architecture of the models and so both models share the names of their attention layers: ‘q_proj’, ‘k_proj’, ‘v_proj’, ‘o_proj’. On the other hand, FALCON has a very different architecture and has 2 attention layers: ‘query_key_value’ and ‘dense’. The 3 models barely loaded on the 16GB of VRAM available for this project but after a lot of hyperparameter optimization, finetuning was successful.

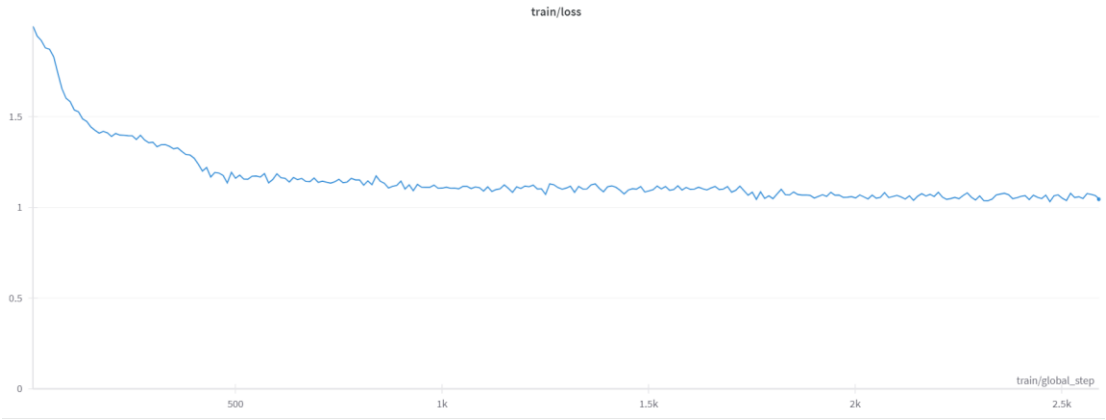

Figure 6: Mistral loss graph

The loss on the finetuning of all models was comparable to that of the graph shown in Figure 6. This loss has a decreasing tendency all around, but does not reach values less than 1, something not desirable in this type of finetuning because that would mean that the model is forgetting the knowledge it was pretrained on. This is called catastrophic forgetting (Oobabooga, 2023). Mistral used a different learning rate than the other two models but achieved similar results. As stated before, VRAM was the most limiting factor in this part of the project, because 16GB of VRAM is barely enough to load 7B models with a 16-bit full weights. But to finetune a model, it is necessary to load the target layers directly to the VRAM, so it takes up most of the memory that is available for this project.

Testing setup

To test the chatbot capabilities of a LLM there is no systematic testing system defined. LLMs as such, not as chatbots, are usually tested on their knowledge or truthfulness. However, the great majority of benchmarks like this are designed to test models entirely in English. As the 3 models are specially finetuned to work only in Spanish, there is just one reliable benchmark found. This benchmark is called Belebele and was developed by Meta to test the performance of models in up to 100 different languages. Belebele has 900 questions to test the reading

comprehension of the models. This dataset is a challenge to most of the State-of-the-art models currently (Barkandar, et al., 2023), so it is a very useful resource to benchmark the performance of the 3 models finetuned in Spanish. There exist other benchmarks such as TruthfulQA or HellaSwag, that are used by the HuggingFace community to benchmark all the models in the platform. However, none of these benchmarks are translated to Spanish so they are not useful to this project.

On the other hand, chatting is a subjective activity that cannot really be measured using testing datasets or conventional machine learning metrics. Also, the 3 LLMs are pre-trained with different datasets and have different architectures, it is not possible for the 3 models to answer the same questions with the same answers. As such, to test the general performance of the models in Spanish: the author and GPT-4 will evaluate the answers given by the model. To test the models' law knowledge 2 different judges will measure the quality of the answers: the author and a law professional, David Mena. These are the questions that are going to be asked:

- General Spanish chatting quality: 50 instructions.
    - 10 basic comprehension prompts
    - 10 complex task prompts
    - 5 creative task questions
    - 5 series of 5 contextual consecutive questions
- Law knowledge chatting quality: 30 instructions.
    - 15 basic law prompts.
    - 15 complex law prompts

The 3 models will be asked the same prompts, and their answers will be documented on a separate document. The 3 models will have 3 chances to generate an answer and the best

answer is going to be chosen. All 3 models used different generation hyperparameters and instruction formats and more configuration in the generation interface. The 3 models used the best hyperparameters found in the time given to the project. The 3 different answers will be compared to one another and be graded with a score from 1 to 5, with 1 being incorrect answer and 5 being perfect answer. This is the scale used in more detail.

Extremely poor **[1]**	• **Accuracy**: The response contains completely incorrect or irrelevant information. • **Relevance**: Does not address the question or requested topic. • **Coherence**: Generates incoherent, nonsensical, or out-of-context text. • **Contextual Understanding**: Shows a total lack of understanding of the topic or context of the question.
Deficient **[2]**	• **Accuracy**: Contains several major errors or misunderstandings about the topic. • **Relevance**: Partially addresses the question but significantly strays from the main topic. • **Coherence**: Text is somewhat coherent but with notable errors or inconsistencies. • **Contextual Understanding**: Displays a limited understanding of the topic or context of the question.
Acceptable **[3]**	• **Accuracy**: Generally correct information with some errors or inaccuracies. • **Relevance**: Adequately addresses the question, though it may lack depth or detail. • **Coherence**: Coherent and orderly text, with some minor issues in fluency or structure. • **Contextual Understanding**: Demonstrates a basic understanding of the topic, though not thoroughly detailed or deep.
Good **[4]**	• **Accuracy**: Accurate and well-founded information with minimal errors. • **Relevance**: Completely addresses the question. • **Coherence**: Well-structured and coherent text with clarity and

	fluency. • **Contextual Understanding**: Shows a solid understanding of the topic and the context of the question.
Perfect **[5]**	• **Accuracy**: Completely correct and detailed information. • **Relevance**: Exhaustively responds to the question, providing a complete understanding of the topic. • **Coherence**: Exceptionally clear, coherent, and well-structured text. • **Contextual Understanding**: Demonstrates a deep and exceptional understanding of the topic and context, including nuances and complex aspects of the question.

Table 3. Scale to rate responses

Likewise, the prompt to ask GPT4 to grade the answer to the given instruction was:

I need your help grading an answer to an instruction given to a Large Language Model chatbot. The instruction was written in Spanish and the answer was expected to be in Spanish as well. The instruction given was "[instruction]" and the answer obtained was "[answer]". In the grading system, a 5 is the best possible answer in your opinion and 1 was the worst possible answer. Following that, your grade will be number from 1 to 5.

Results

Results are going to be divided in three sections: one for the Belebele benchmark, and two for the Spanish conversation instructions and law-specific instructions because the results from the 2 types of instructions were very different from each other.

Spanish instructions LORA

The finetuning of the 3 models to follow instructions in Spanish was a success. All 3 models behaved differently but were able to follow most of the instructions given to them. This is the results that were obtained:

Tasks	LLaMA2-Chat	Mistral	FALCON
Simple	3.4	4	3.2
Complex	4.11	4.11	3.25
Creative	3.2	4	5
Contextual	3.76	4	3.36
Average	3.6175	**4.0275**	3.7025

Table 4. Result from the Spanish instructions testing.

These results were obtained after calculating the average for all the instructions given to the models. Overall, the results were satisfactory because all 3 models were able to follow most of the instructions given to them. An informal test was done to test if before finetuning any of the 3 models were able to follow instructions in Spanish and all 3 could do so. LLaMA2-Chat was the only model that could follow instructions although the responses were not great. However, this result was perfectly outlined in the Meta paper where LLaMA2 was introduced, where it is clearly stated that the model may not be suited for use in other languages other than English (Touvron, et al., 2023). Mistral could understand Spanish, but its responses were written mostly in English and FALCON responded in a combination of both languages that resulted in confusing and unreadable answers. FALCON's performance was worrying because it is stated in its paper that almost 10% of its total pretraining data is in Spanish (Penedo, et al., 2023).

However, after finetuning all 3 models performed a lot better when following instructions in Spanish. Mistral was the one that showed the most improvement over the other 2 models. As shown in Table 4, Mistral was the top performer among the 3 other models and it was a clear

difference, even though the numbers may not show it. Mistral had little trouble explaining concepts and remembering prior knowledge because it has the biggest context size of the 3 models (Mitral AI, 2023). FALCON was great at creative tasks, but most of the time it had trouble stopping itself after completing ideas and it just kept generating tokens until it hit the limit of tokens set up. LLaMA2-Chat was the easiest model to set up and had good results in all the tasks it was presented with except creative tasks. This makes sense because this model is a finetuned version of LLaMA2 that has been optimized to increase its chatting capabilities in English instead of creative capabilities. Maybe using the base LLaMA2 model would yield better results but the whole purpose of LLaMA2 was to create the best small LLM for chatting applications so that would defeat the purpose set up by Meta.

Belebele benchmark

	LLaMA2-Chat	**Mistral**	**FALCON**
Without LORA	54.7%	40.4%	29.5%
With LORA	47.2%	33.4%	28.7%

Table 5. Result from the Belebele benchmark

The results of the Belebele benchmark throw an interesting new question. Why are the base models able to perform better in this standardized test without being finetuned compared to their finetuned counterpart? This is even more confusing after stating earlier that 2 of the 3 models performed a lot better after being finetuned to answer in Spanish than their base models. Only LLaMA2 was able to follow some instructions in Spanish and even then, it had trouble when explaining complex topics. Mistral and FALCON couldn’t even hold a conversation in Spanish before the finetuning.

On the other hand, although the results do not look impressive, Mistral and LLaMA2 performed greatly in this benchmark. This benchmark was designed to be particularly challenging to even the best large language models in the world in 2023. Meta's paper show that the results presented here are in par to the ones obtained by themselves (Barkandar, et al., 2023). Only the bigger models, such as LLaMA2-70B and ChatGPT Turbo could easily achieve over 60% accuracy in this benchmark in Spanish. FALCON did the worst of the 3 models, something surprising considering that this model had the most Spanish included in its pretraining data, as stated before. However, something remarkable is the fact that it was the least affected model by the finetuning process. This means that the extra Spanish pretraining data better prepare FALCON to be finetuned to this language. Taking this into account, it would be an exaggeration to say that the finetuned models performed a lot worse than their base counterparts. All 3 models lost some of their reasoning skills after the finetuning process although it is extremely hard to pinpoint the exact reason for it, be it the quality of the training dataset or even catastrophic forgetting.

Law instructions LORA

Unlike the earlier tests performed on the models, this testing process had the worst outcome of the 3 testing processes and really shows the limitation of current technologies and techniques in Large Language Models.

	LLaMA2-Chat	**Mistral**	**FALCON**
Simple	3.86	3.35	3.64
Complex	2.23	2.36	3.16

Table 6. Result from the law-related instructions testing.

The results from this test were extremely poor, especially compared to the results obtained in the first two tests. LLaMA2-Chat and Mistral responses were nearly unreadable, full of contradictory information and hallucinations. On the other hand, and contrary to the 2 other tests done, FALCON did the best of the 3 models answering most questions correctly, although not precisely enough to have more rating. It is important to highlight that even FALCON did not produce helpful answers and the only way they made sense was by having previous knowledge of the topic, something which is simply not feasible in the eventual application that these 3 models were to have. This could also be because FALCON had the most Spanish data in its pretraining dataset and, as such, it is easier for it to build the necessary patterns and logic behind the new content the LORA introduces. In contrast, Mistral and LLaMA2 have very limited Spanish knowledge before the first finetuning process.

Another important highlight of this third experiment was that all 3 models produced the best results using the Contrastive Search generation method. This is a generation method that combines two types of generation: maximization-based methods and stochastic methods. The first type of method basically searches for the most likely word to follow, but sometimes it results in unwanted repetition meanwhile the second type of method introduces a factor of randomness considering less likely options however this may result in content that is semantically incorrect (Su, et al., 2022). Contrastive Search unifies both concepts, searching for the most likely word but at the same time comparing its options to with a degeneration penalty which allows it to avoid repetition but also maintaining semantic sense (Su & Collier, 2023). The implementation of Contrastive Seach in the interface used for generation considers very few words at the same time so I think this, alongside the degeneration penalty, helps the model make up for their extremely limited Ecuadorian law vocabulary so that

generation made the most sense, even though the responses were not always true. Other methods such as Beam Search or Top-K produced even worse results than the ones presented in Table 6.

CONCLUSSIONS

This project has explored the latest technologies in Large Language Models given reasonable hardware limitations. While it is true that more VRAM has available in the form of the institution's remote server, it is simply not feasible to analyze the impact of these technologies in Ecuador with such an expensive and exclusive hardware setup. The results of the project have been a moderate success and throw light on the future possibilities of LLMs.

The main limiting factor has been VRAM memory. While it is known that Mistral 7B is probably the most impressive and capable small LLM with incredible projects such as Intel's Neural Chat finetune (Hugging Face, 2023), it is still inferior to bigger projects such as Qwin 64b or even the same company's Mixtral 8x7b. Even the most impressive mistral finetunes such as OpenHermes 2.5 or Intel's Neural Chat only work in English and have no real Spanish-instruction following capabilities. A future project could explore the improvements of using bigger LLMs without the VRAM limitation and compare its results to this project. As a side note, these projects are not apt to be finetuned further because they are already finetuned and this usually creates generation problems.

Referring to finetuning, Low Rank Adaptation was the correct decision to adapt all models to the desired applications. It also allowed the project to maintain its VRAM limitation thanks to Hugging Face's useful collection of libraries. While there exists other libraries to use LLMs with CPU and RAM inference, these methods are usually slower and harder to setup and introduce another hardware limitation in both of those components. Currently, NVIDIA is the best hardware manufacturer for these types of projects thanks mainly to its CUDA architecture. This project has shown that there exists a possibility to create high-quality finetunes of these small models to open possibilities in Spanish-speaking countries with little access to extremely exclusive hardware setups.

Considering the results obtained earlier, it would be recommended to explore further possibilities with LLaMA2 and Mistral only. FALCON did not perform terribly all the time, but it did struggle to finish its generation and was the most trouble setting up, testing generation hyperparameters and finetune hyperparameters. Even the Hugging Face Hub shows very few FALCON finetunes, as the 7B version is not capable enough to dedicate time and other projects into. The bigger models, of 40B and 180B parameters may perform a lot better as those are the insignia models of the IIT and use different datasets. Models like LLaMA2 and Mistral have had more attention in the Hub, with projects such as Goliath-120B and the earlier mentioned Mistral finetunes. LLaMA2 was the model with the more natural and human-like responses of the 3, something which makes sense taking into account the big amount of effort Meta did to improve the model's conversational skill with techniques such as RLHF (Schmid, Sanseviero, Cuenca, & Tunstall, 2023). Mistral on the other hand had the best performance overall, highlighting its capability to produce concise and precise answers to difficult questions, something in which the 2 other models failed.

In terms of improvement, there have recently emerged new technologies that could drastically improve the performance of small LLMs such as Mixture of Experts (MoE) or Retrieval Augmented Generation (RAG). MoE is a technology in which the model is not a single neural network, but instead it is composed of several 'experts' which are different neural networks that specialize in certain tasks. MoE allows for easier pretraining of models, less computing cost, and better responses, although it may be difficult to finetune to specific use cases (Sanseviero, et al., 2023). In fact, this technology is used by ChatGPT-4, the best LLM in the world currently, with the model supposedly having 16 'experts' and each of one containing approximately 110 billion of parameters (Betts, 2023). This would mean that GPT-4 works with more than 1 trillion parameters, something which is not really surprising

considering how far ahead the model is of its Open-Source competition. On the qother hand, RAG is a technology in which an external database is used alongside the model so that the model can use it to expand its knowledge when asked about certain topics. This reduces the task of finetuning models and reduces the problem of hallucinations (Riedel, Kiela, Lewis, & Piktus, 2020). However, it also requires more capable models, as the model itself needs to access the information on the database and make sense of it. However, both technologies have only recently been used in LLMs and are still extremely difficult to implement in Spanish in 2023. On the other hand, it is important to consider Moore's Law, which will open a lot of possibilities for large language models and combined with these new technologies, which heights might these models achieve in the upcoming years?

BIBLIOGRAPHY

Abdi, H., Valentin, D., & Edelman, B. (1999). *Neural Networks.* London: The International Professional Publishers.

Abideen, Z. (2023, June). *Attention Is All You Need: The Core Idea of the Transformer*. Retrieved from Medium: https://medium.com/@zaiinn440/attention-is-all-you-need-the-core-idea-of-the-transformer-bbfa9a749937

Aggarwal, C. (2018). *Neural Networks and Deep Learning.* New York: Springer International Publishing.

Aghajanyan, A., Zettlemoyer, L., & Gupta, S. (2020). *Intrinsic dimensionality explains the effectiveness of language model fine-tuning.* Menlo Park: Meta.

Alabbas, W. (2023, September). *The ChatGPT Revolution: Transforming Our Everyday Lives and Careers*. Retrieved from LinkedIn: https://www.linkedin.com/pulse/chatgpt-revolution-transforming-our-everyday-lives-careers-alabbas/

Alammar, J. (2019, August). *The Illustrated GPT-2* . Retrieved from GitHub: http://jalammar.github.io/illustrated-gpt2/

Barkandar, L., Liang, D., Muller, B., Artetxe, M., Narayan, S., Husa, D., . . . Khabsa, M. (2023). *The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants.* Menlo Park : Meta AI.

Betts, S. (2023, July). *Peering Inside GPT-4: Understanding Its Mixture of Experts (MoE) Architecture*. Retrieved from Medium: https://medium.com/@seanbetts/peering-inside-gpt-4-understanding-its-mixture-of-experts-moe-architecture-2a42eb8bdcb3

DeepLearning.AI. (2023, January). *A complete guide to Natural Language Processing*. Retrieved from DeepLearning.AI: https://www.deeplearning.ai/resources/natural-language-processing/

Dettmers, T. (2015, December). *Deep Learning in a Nutshell: History and Training*. Retrieved from NVIDIA Developer: https://developer.nvidia.com/blog/deep-learning-nutshell-history-training/

Dettmers, T., Pagnoni, A., & Holtzman, A. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs.* Wasshington: University of Washington.

Devlin, J., Chang, M.-W., Kenton, L., & Toutanova, K. (2018). *Understanding, BERT: Pre-training of Deep Bidirectional Transformers for Language.* Mountain View: Google AI Language.

Hardesty, L. (2017, April). *Explained: Neural networks*. Retrieved from MIT News: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., . . . Chen, W. (2021). *LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS.* Montreal: arXiv.

Hugging Face. (2023, December). *Open LLM Leaderboard*. Retrieved from Hugging Face: https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Hugging Face. (2023). *Quantize Hugging Face Transformers models*. Retrieved from Hugging Face: https://huggingface.co/docs/transformers/main_classes/quantization#bitsandbytes-integration

ISO. (2022). *Artificial intelligence concepts and terminology.* Ginebra: ISO.

Jiang, A., Sablayrolles, A., Mensch, A., Bamford, C., Singh, D., De las Casas, D., . . . Le Scao, T. (2023). *Mistral 7B.* Paris: MistralAI.

Johnson, A. (2023, July). *Parameter Size vs Performance in Large Language Models*. Retrieved from Medium: https://medium.com/@andrew_johnson_4/parameter-size-vs-performance-in-large-language-models-c00611935258#:~:text=Parameter%20Size%20and%20Performance%20in%20Large%20Language%20Models&text=In%20general%2C%20this%20has%20been,and%20produce%20higher-quality%

Kapoor, A. (2021, October). *How Artificial Intelligence Is Revolutionizing The World*. Retrieved from LinkedIn: https://www.linkedin.com/pulse/how-artificial-intelligence-revolutionizing-world-anshul-kapoor/

Labonne, M. (2023, July). *Introduction to Weight Quantization*. Retrieved from Medium: https://towardsdatascience.com/introduction-to-weight-quantization-2494701b9c0c

Lane, H., & Dyshel, M. (2023). *Natural Language Processing in Action.* San Diego: Manning Publications.

Lee, A. (2023, January). *What Are Large Language Models Used For?* Retrieved from NVIDIA: https://blogs.nvidia.com/blog/2023/01/26/what-are-large-language-models-used-for/#:~:text=It%20can%20be%20used%20for,chatbots%2C%20AI%20assistants%20and%20more.

Lui, P., Saleh, M., Pot, E., Ben, G., Sepassi, R., Lukasz, K., & Shazeer, N. (2018). *GENERATING WIKIPEDIA BY SUMMARIZING LONG.* Mountain View: Google Brain.

Mahapatra, S. (2018, March). *Why Deep Learning over Traditional Machine Learning?* Retrieved from Towards Data Science: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-

1b6a99177063#:~:text=A%20big%20advantage%20with%20deep,new%20innovations%20in%20deep%20learning.

Microsoft. (2023, April). *What are Models?* Retrieved from Microsoft: https://learn.microsoft.com/en-us/semantic-kernel/prompt-engineering/llm-models

Mitral AI. (2023, September). *Mistral 7B*. Retrieved from MistralAI: https://mistral.ai/news/announcing-mistral-7b/

Muehmel, K. (2023, June). *What Is a Large Language Model, the Tech Behind ChatGPT?* Retrieved from dataiku: https://blog.dataiku.com/large-language-model-chatgpt

Nadkarni, P., Ohno-Machado, L., & Chapman, W. (2011). Natural language processing: an introduction . *JAMIA*, 544-551.

Oobabooga. (2023, October). *text-generation-webui - Training Tab*. Retrieved from GitHub: https://github.com/oobabooga/text-generation-webui/wiki/05-%E2%80%90-Training-Tab

OpenAI. (2023). *GPT-4*. Retrieved from OpenAI: https://openai.com/research/gpt-4

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., . . . Launay, J. (2023). *The RefinedWeb Dataset for Falcon LLM.* Abu Dabhi: TII.

Riedel, S., Kiela, D., Lewis, P., & Piktus, A. (2020). *Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models*. Retrieved from Meta: https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/

Riedl, M. (2023, April). *A Very Gentle Introduction to Large Language Models without the Hype*. Retrieved from Medium: https://mark-riedl.medium.com/a-very-gentle-introduction-to-large-language-models-without-the-hype-5f67941fa59e

Roberts, J. (2023). *On the Computational Power of Decoder-Only Transformer Language Models.* Nashville: Vanderbilt University.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction.* Berlín: Springer-Berlag .

Sanseviero, O., Tunstall, L., Schmid, P., Mangkrullar, S., Belkada, Y., & Cuenca, P. (2023, December). *Mixture of Experts Explained*. Retrieved from Hugging Face: https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe

Sarkar, A. (2022, February). *All you need to know about 'Attention' and 'Transformers' — In-depth Understanding — Part 1*. Retrieved from Medium: https://towardsdatascience.com/all-you-need-to-know-about-attention-and-transformers-in-depth-understanding-part-1-552f0b41d021#9c93

Schmid, P., Sanseviero, O., Cuenca, P., & Tunstall, L. (2023, July). *Llama 2 is here - get it on Hugging Face*. Retrieved from Hugging Face: https://huggingface.co/blog/llama2#why-llama-2

Su, Y., & Collier, N. (2023). *Contrastive Search Is What You Need For Neural Text Generation.* Cambridge: University of Cambridge.

Su, Y., Lan, T., Wang, Y., Yogatama, D., Kong, L., & Collier, N. (2022). *A Contrastive Framework for Neural Text Generation.* Cambridge: University of Cambridge.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., . . . Hashimoto, T. (2023). *Stanford Alpaca: An Instruction-following LLaMA model*. Retrieved from GitHub: https://github.com/tatsu-lab/stanford_alpaca

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., . . . Rodriguez, A. (2023). *LLaMA: Open and Efficient Foundation Language Models.* Menlo Park: Meta.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahari, A., Babaei, Y., . . . Esiobu, D. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Model.* Menlo Park: Meta.

Varma, R. (2020, August). *Why we need multi layer neural networks (MLP)?* Retrieved from Medium: https://medium.com/@Rohit_Varma/why-we-need-multi-layer-neural-network-mlp-d50425b8f37d

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., . . . Polosukhin, I. (2017). *Attention Is All You Need.* Long Beach: Google.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., & Borgeaud, S. (2022). Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*, 155-185.