University of Verona
Department of Informatics
Master's Degree in Data Science

~ · ~

Academic Year 2021–2022

Leveraging LLMs with RAG to recommend Points of Interest to tourists

Supervisor
Prof. Niccolò Marastoni

Graduate Student
Patrick Hamzaj
VR474246

Dedica

Acknowledgments

Abstract

Qui va l'abstract

Contents

Introduction

Over the last decade, the field of artificial intelligence (AI from now on) has experienced significant growth and rapid advancements. Among the different branches of AI, *deep learning* has stood out for its ability to tackle complex tasks that were considered infeasible or extremely challenging. In particular, *Natural Language Processing* (NLP) has gained prominence due to the increasing volume of unstructured and textual data generated and subsequently gathered online (e.g., social media, blogs, scientific articles...); it is estimated that 328.77 million terabytes of data are created daily, 80% of which are unstructured data. [1] This phenomenon and the rising demand for language-driven applications, such as virtual assistants, sentiment analysis, and automated content generation, has prepared the terrain for the advancement of languange processors to flourish.

Recently, the emergence of *Large Language Models* (LLMs), led by the release of GPT-3 in 2020 [2], has taken NLP to a new level. By employing architectures with billions of parameters, these models are capable of producing remarkably fluent, context-aware and human-like text. Despite this progress, several open challenges remain: questions regarding how best to *fine-tune* LLMs, how to incorporate domain-specific knowledge, and how to design effective prompting mechanisms are active areas of research. Additionally, issues related to computational resources, scalability, ethical implications, and potential biases call for continuous investigation.

Motivations. The primary motivation behind this thesis is to explore the practical methods and architectural choices to enable the construction of an LLM-powered application to be effectively adapted for the domain of tourism in Verona, Italy. In particular, the aim is to utilize and demonstrate how techniques such as *prompt engineering*, and *retrieval-augmented generation* (RAG) can facilitate the deployment of state-of-the-art LLMs in real-world applications, including scenarios where computational resources may be limited or costly.

As such, the **objectives** are:

- **Analyze** the foundational principles and evolution of modern neural network architectures, particularly focusing on the *Transformer* model and its role in Large Language Models.
- **Investigate** the current state of the art for LLMs, highlighting successful applications and the most common methodologies (fine-tuning, prompt engineering, RAG, etc.).
- **Design and implement** a tourism domain application system leveraging a specific open-source LLM (LLaMA 3.1 8B Instruct), showcasing relevant techniques (8-bit quantization, agent-like conversational behavior) for an implicit recommender system for tourists.

- **Evaluate** the system with respect to performance and quality metrics, as well as potential limitations.

From a high-level perspective, Large Language Models represent a class of deep neural netowrks trained on extensive corpora, often comprising billions of parameters. A key turning point in their development was the introduction of the *Transformer* architecture, which provides a self-attention mechanism that allows to improve text tokens processing in a parallel manner during training.

Examples of prominent LLMs include GPT-based models (GPT-3, GPT-3.5, GPT-4), [2] Google's PaLM [3], Meta's LLaMA, [4] and various open-source initiatives such as Bloom. [5] These models have demonstrated remarkable capabilities in language understanding and generation, enabling:

- **Text completion** with human-like fluency.
- **Zero-shot or few-shot generalization** to new tasks with minimal prompt examples.
- **Conversational AI**, powering advanced chatbots and virtual assistants.

This thesis will leverage the open-source LLaMA model family as a foundation for an LLM-based chatbot to suggest points of interest to touristi visiting the city of Verona, Italy. By focusing on a quantized 8-bit instruct variant (LLaMA 3.1 8B Instruct), the aim is to highlight practical techniques (RAG, agent AI) that maintain acceptable performance implementing innovative strategies that harness context-awareness and text generation capabilities/question answering of modern LLM models.

The remainder of this document is organized as follows: Chapter 1 presents a deeper dive into the theoretical underpinnings of modern NLP, covering the history of neural networks and the transformation from classic word embeddings to large-scale language models, also detailing the breakthrough Transformer architecture. Chapter 2 reviews the current landscape of Large Language Models and NLP solutions, exploring established techniques (fine-tuning, prompt engineering, RAG) and highlighting real-world use cases, from chatbots to advanced content generation. Chapter 3 describes the design and implementation of an application based on LLaMA 3.1 8B Instruct for tourism purposes in the city of Verona, Italy. It covers the decision-making process behind the model choice, 8-bit quantization strategy, prompt engineering, retrieval-augmented generation, and the agent-like conversational framework. Chapter 4 discusses the evaluation with illustrative examples of the model's outputs and conversations, along with limitations and areas for improvement. Finally, Chapter 5 concludes the study with a summary of key findings, limitations, and potential future directions.

Chapter 1

Theoretical Foundations and Background

1.1 A Brief History of Neural Networks

The term *Neural Network* refers to the attempt of defining a mathematical view of the structure of the human brain, aiming at emulating it and transposing the logical functioning and the learning capabilities into computational models.

The very first attempt of studying the biological brain and its neural activity in terms of formal logic is attributed to McCulloch and Pitts,[6] who proposed that neurons could be represented as simple binary devices, whose key aspects are:

- **Logical Units:** neurons are basic units with an on/off switch, allowing them to describe neural activi'ty using the language of logic (i.e. logical propositions like *AND*, *OR*...).
- **Threshold:** neurons "fire" only when a certain threshold of input is met.
- **Links:** like synapses, links connect the logical units from the input to the output.

This basic understanding of neural processes influeced later developmentes in neuroscience and artificial intelligence, bringing Rosenblatt to produce the *Perceptron*: a single layer neural network that, upon previous work from McCulloch and Pitts, introduced:

- **Weights:** every connection has a weight, that is added to the input received.
- **Learning mechanism:** introduces a learning algorithm where the weights of the connections are adjusted based on errors in the output.

This linear architecture allows to generalize the classification tasks, using probabilistic rules that could perform nontrivial tasks like pattern recognition and information organization.

Neural networks have evolved dramatically over the decades

The enthusiasm was enormous and the field of cybernetics was born: however, it didn't take long for researchers to uncover the limitations of single-layer networks. Minsky's and Papert's demonstrated the limitations of the perceptron—that is, that certain classes of functions were simply out of reach for these early models (for example, the logical XOR

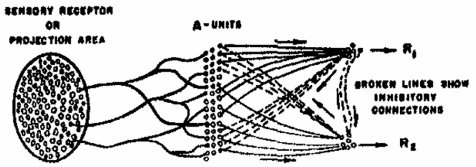

FIG. 2A. Schematic representation of connections in a simple perceptron.

Figure 1.1: The perceptron architecture.

function)—interest quickly waned. [7] This realization contributed to a period of reduced enthusiasm for neural network research, often referred to as the "AI Winter."

Interest was revived in the 1980s with the introduction of the backpropagation algorithm [8], which made it possible for deeper architectures to learn more complex functions. The core idea was to train neural networks by propagating the error from the output layer backward through the network layers; this approach allows the network to adjust its internal weights based on the error itself, so that it could "learn" using gradient descent to minimize the error function, changing the weights individuating the contribution of each neuron in constructing the output.

Although early progress was hampered by hardware constraints, continuous incremental improvements over the following decades, combined with advances in parallel computing, eventually paved the way for the deep learning revolution we see today [9].

By the early 2010s, the impact of Convolutional Neural Networks (CNNs) on computer vision tasks [10] highlighted the benefits of large datasets, GPU-based parallel training, and increasingly sophisticated network designs. At the same time, improvements in Recurrent Neural Networks (RNNs)—especially with LSTM [11]—opened up new possibilities in sequence modeling, including areas like language translation and speech recognition. These advancements ultimately set the stage for the development of Large Language Models (LLMs), particularly after the introduction of the Transformer architecture [12].

1.2 The Transformer Architecture

The introduction of the Transformer by Vaswani et al. [12] marked a clear departure from traditional recurrent and convolutional models. The focus of the original research was on translation tasks, by leaning heavily on a **self-attention** mechanisms which allows the model to weigh the importance of different words in a seuqence of words, taking into account the relationship to eacht ore.

The Transformer architecture consists of two main parts: an **encoder** and a **decoder**, each built from multiple layers that implement multi-head self-attention alongside feed-forward networks, which can be visualized in 1.2:

1. **Encoder:** it converts an input sequence (e.g., a sentence) into a series of representations that capture the contextual meaning of the input. Within each encode layer, a sub-layer allows every token in the input to consider the influence of every other token.

2. **Decoder:** it uses the encoder's representation along with other inputs to generate a target sequence using a masking mechanism (i.e. predicting the next token is only decided to previous tokens, and not future ones).

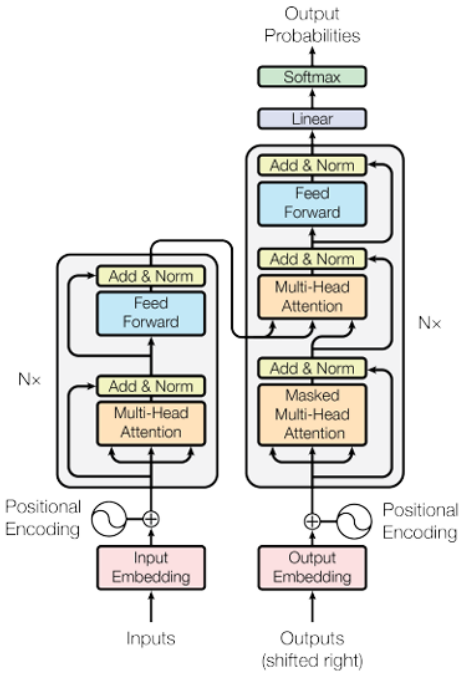

Figure 1.2: The Transformer architecture.

Positional encoding is another breakthrough of this paper, integrated in both parts, which compensates for the lack of sequence awareness. It incorporates information about the order of tokens in a sequence. Since these models process tokens in parallel rather than sequentially, positional embeddings are essential for conveying information about the order of tokens.

Each of these parts can be used independently, depending on the tasks: encoder-only models are good for tasks that require understanding of the input, such as sentence classification; decoder-only models are good for generative tasks such as text generation. Encoder-decoder models (called sequence-to-sequence models) are used for generative tasks that require an input, such as translation or summarization.

1.3 Transformers as Language Models

In recent applications, the Transformer model have been trained as *language models* meaning they have been trained on large amounts of raw text in a self-supervised fashion, which is a type of training in which the objective is automatically computed from the inputs of the model.

This type of models develop a statistical understanding of the language it has been trained on, but it's not very useful for specific practical tasks. Because of this, the general pretrained model then goes through a process called *transfer learning*. During this process, the model is fine-tuned in a supervised way — that is, using human-annotated labels — on a given task.

An example of a task is predicting the next word in a sentence having read the previous

words. This is called causal language modeling because the output depends on the past and present inputs, but not the future ones.

Training is a foundamental step in the adoption and implementation of a language model. The learning capacity of LLMs are generally dived into:[13]

1. **Pre-training:** it is the first stage in training an LLM, where the model learns general linguistic patterns, facts and knowledge from a vast corpus of text. It is the act of training a model from scratch: the weights are randomly initialized, and the training starts without any prior knowledge. Tecniques of pre-training phase are:

   - *Masked language modeling*, used in decoder-models, where certain words are masked and the model learns to predict them.[14]
   - *Causal language modeling*, used in encoder-only models like GPT,[2] where the model predicts the next word in a sequence.

2. **Fine-tuning:** after pre-training, the model undergoes a further training on a smaller, task-specific dataset to improve performance for particular and domain-specific applications. Types of fine-tuning are:

   - *Supervised fine-tuning* is used to train models on labeled data, such as question-answering datasets.
   - *Instruction tuning* involves training the model on a dataset of input-output pairs, where each input is phrased as an instruction and the output is the desired response. Most ready-to-use models are instruct-tuned, as they have improved generalization and natural responses. An example can be seen in Table 1.1.
   - *Parameter-Efficient fine-tuning (PEFT)*, methods like *Low-Rank Adaption (LORA)*[15] are innovative techniques that reduce the number of parameters to train, thus reducing computaional costs.

Generally, the strategy to achieve better performance is by increasing the models' sizes as well as the amount of data they are pretrained on, but higher performances lead to higher resources intensive trainings. This is why different strategies have been developed to achieve good performances without the need of training models.

Instruction	**Input (optional)**	**Expected Output**
Translate this sentence into Spanish	"Hello, how are you?"	"Hola, ¿cómo estás?"
Summarize the text in one sentence	"The global economy is facing uncertainty due to inflation and geopolitical issues."	"The global economy is unstable due to inflation and geopolitics."
Explain how photosynthesis works to a 5-year-old	*No input*	"Plants use sunlight to make food, like how we eat to get energy!"

Table 1.1: Examples of Instruction Tuning

1.4 Challenges and Limitations of LLMs

1.5 Summary

Chapter 2

Evolution and Utilization of Large Language Models

2.1 Principali LLM attualmente in uso (GPT, PaLM, LLaMA, Bloom, T5, ChatGPT, ecc.)

2.2 Approcci di personalizzazione dell'LLM (fine-tuning, prompt engineering, agent AI)

2.3 RAG: concetti chiave e vantaggi

2.4 3.4 Prompt Engineering: concetti chiave e vantaggi

2.5 3.5 Tool Usage: concetti chiave e vantaggi

Chapter 3

Methodology and Implementation

3.1 Descrizione degli obiettivi specifici del progetto

3.2 Scelta di LLaMA 3.1 8B Instruct: motivazioni e vantaggi

3.3 Quantizzazione a 8 bit: principi e tto sulle prestazioni

3.4 Prompt engineering e RAG

3.5 Integrazione di un comportamento di tipo agent AI" (tool usage)

3.6 Strumenti e ambienti di sviluppo utizzati (framework, hardware, librerie)

3.7 Descrizione step-by-step dell'implementazione (pipeline e flusso di lavoro)

Chapter 4

Evaluation

4.1 Analisi dei risultati ottenuti: prestazioni del modello, coerenza, accuratezza e limiti

4.2 Esempi di conversazioni e discussione delle principali osservazioni

4.3 Confronto con soluzioni alternative e best practice emerse

Conclusions

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis

cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Appendix A

Albero

A.1 Prova

Come funziona un'appendice

Nullam eleifend justo in nisl. In hac habitasse platea dictumst. Morbi nonummy. Aliquam ut felis. In velit leo, dictum vitae, posuere id, vulputate nec, ante. Maecenas vitae pede nec dui dignissim suscipit. Morbi magna. Vestibulum id purus eget velit laoreet laoreet. Praesent sed leo vel nibh convallis blandit. Ut rutrum. Donec nibh. Donec interdum. Fusce sed pede sit amet elit rhoncus ultrices. Nullam at enim vitae pede vehicula iaculis.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Aenean nonummy turpis id odio. Integer euismod imperdiet turpis. Ut nec leo nec diam imperdiet lacinia. Etiam eget lacus eget mi ultricies posuere. In placerat tristique tortor. Sed porta vestibulum metus. Nulla iaculis sollicitudin pede. Fusce luctus tellus in dolor. Curabitur auctor velit a sem. Morbi sapien. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Donec adipiscing urna vehicula nunc. Sed ornare leo in leo. In rhoncus leo ut dui. Aenean dolor quam, volutpat nec, fringilla id, consectetuer vel, pede.

Nulla malesuada risus ut urna. Aenean pretium velit sit amet metus. Duis iaculis. In hac habitasse platea dictumst. Nullam molestie turpis eget nisl. Duis a massa id pede dapibus ultricies. Sed eu leo. In at mauris sit amet tortor bibendum varius. Phasellus justo risus, posuere in, sagittis ac, varius vel, tortor. Quisque id enim. Phasellus consequat, libero pretium nonummy fringilla, tortor lacus vestibulum nunc, ut rhoncus ligula neque id justo. Nullam accumsan euismod nunc. Proin vitae ipsum ac metus dictum tempus. Nam ut wisi. Quisque tortor felis, interdum ac, sodales a, semper a, sem. Curabitur in velit sit amet dui tristique sodales. Vivamus mauris pede, lacinia eget, pellentesque quis, scelerisque eu, est. Aliquam risus. Quisque bibendum pede eu dolor.

Donec tempus neque vitae est. Aenean egestas odio sed risus ullamcorper ullamcorper. Sed in nulla a tortor tincidunt egestas. Nam sapien tortor, elementum sit amet, aliquam in, porttitor faucibus, enim. Nullam congue suscipit nibh. Quisque convallis. Praesent arcu nibh, vehicula eget, accumsan eu, tincidunt a, nibh. Suspendisse vulputate, tortor quis adipiscing viverra, lacus nibh dignissim tellus, eu suscipit risus ante fringilla diam. Quisque a libero vel pede imperdiet aliquet. Pellentesque nunc nibh, eleifend a, consequat consequat, hendrerit nec, diam. Sed urna. Maecenas laoreet eleifend neque. Vivamus purus odio, eleifend non, iaculis a, ultrices sit amet, urna. Mauris faucibus odio vitae risus. In nisl. Praesent purus. Integer iaculis, sem eu egestas lacinia, lacus pede scelerisque augue,

in ullamcorper dolor eros ac lacus. Nunc in libero.

Fusce suscipit cursus sem. Vivamus risus mi, egestas ac, imperdiet varius, faucibus quis, leo. Aenean tincidunt. Donec suscipit. Cras id justo quis nibh scelerisque dignissim. Aliquam sagittis elementum dolor. Aenean consectetuer justo in pede. Curabitur ullamcorper ligula nec orci. Aliquam purus turpis, aliquam id, ornare vitae, porttitor non, wisi. Maecenas luctus porta lorem. Donec vitae ligula eu ante pretium varius. Proin tortor metus, convallis et, hendrerit non, scelerisque in, urna. Cras quis libero eu ligula bibendum tempor. Vivamus tellus quam, malesuada eu, tempus sed, tempor sed, velit. Donec lacinia auctor libero.

Praesent sed neque id pede mollis rutrum. Vestibulum iaculis risus. Pellentesque lacus. Ut quis nunc sed odio malesuada egestas. Duis a magna sit amet ligula tristique pretium. Ut pharetra. Vestibulum imperdiet magna nec wisi. Mauris convallis. Sed accumsan sollicitudin massa. Sed id enim. Nunc pede enim, lacinia ut, pulvinar quis, suscipit semper, elit. Cras accumsan erat vitae enim. Cras sollicitudin. Vestibulum rutrum blandit massa.

Sed gravida lectus ut purus. Morbi laoreet magna. Pellentesque eu wisi. Proin turpis. Integer sollicitudin augue nec dui. Fusce lectus. Vivamus faucibus nulla nec lacus. Integer diam. Pellentesque sodales, enim feugiat cursus volutpat, sem mauris dignissim mauris, quis consequat sem est fermentum ligula. Nullam justo lectus, condimentum sit amet, posuere a, fringilla mollis, felis. Morbi nulla nibh, pellentesque at, nonummy eu, sollicitudin nec, ipsum. Cras neque. Nunc augue. Nullam vitae quam id quam pulvinar blandit. Nunc sit amet orci. Aliquam erat elit, pharetra nec, aliquet a, gravida in, mi. Quisque urna enim, viverra quis, suscipit quis, tincidunt ut, sapien. Cras placerat consequat sem. Curabitur ac diam. Curabitur diam tortor, mollis et, viverra ac, tempus vel, metus.

Curabitur ac lorem. Vivamus non justo in dui mattis posuere. Etiam accumsan ligula id pede. Maecenas tincidunt diam nec velit. Praesent convallis sapien ac est. Aliquam ullamcorper euismod nulla. Integer mollis enim vel tortor. Nulla sodales placerat nunc. Sed tempus rutrum wisi. Duis accumsan gravida purus. Nunc nunc. Etiam facilisis dui eu sem. Vestibulum semper. Praesent eu eros. Vestibulum tellus nisl, dapibus id, vestibulum sit amet, placerat ac, mauris. Maecenas et elit ut erat placerat dictum. Nam feugiat, turpis et sodales volutpat, wisi quam rhoncus neque, vitae aliquam ipsum sapien vel enim. Maecenas suscipit cursus mi.

Quisque consectetuer. In suscipit mauris a dolor pellentesque consectetuer. Mauris convallis neque non erat. In lacinia. Pellentesque leo eros, sagittis quis, fermentum quis, tincidunt ut, sapien. Maecenas sem. Curabitur eros odio, interdum eu, feugiat eu, porta ac, nisl. Curabitur nunc. Etiam fermentum convallis velit. Pellentesque laoreet lacus. Quisque sed elit. Nam quis tellus. Aliquam tellus arcu, adipiscing non, tincidunt eleifend, adipiscing quis, augue. Vivamus elementum placerat enim. Suspendisse ut tortor. Integer faucibus adipiscing felis. Aenean consectetuer mattis lectus. Morbi malesuada faucibus dolor. Nam lacus. Etiam arcu libero, malesuada vitae, aliquam vitae, blandit tristique, nisl.

Appendix B

Barca

B.1 Prova

Appendice B

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Bibliography

[1] Statista. *Amount of Data Created Worldwide 2010-2025*. 2025. URL: `https://www.statista.com/statistics/871513/worldwide-data-created/`.

[2] Tom Brown et al. "Language Models are Few-Shot Learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[3] Aakanksha Chowdhery et al. "PaLM: Scaling Language Modeling with Pathways". In: *arXiv preprint arXiv:2204.02311* (2022).

[4] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *arXiv preprint arXiv:2302.13971* (2023).

[5] Teven Le Scao, Angela Fan, Christopher Akiki, et al. "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model". In: *arXiv preprint arXiv:2211.05100* (2022).

[6] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *The Bulletin of Mathematical Biophysics* 5.4 (1943), pp. 115–133.

[7] Marvin Minsky and Seymour Papert. *Perceptrons*. MIT Press, 1969.

[8] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2012.

[11] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[12] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. Vol. 30. 2017.

[13] Yixin Liu, Haoyu Zhang, Zhanpeng Zhang, et al. "Understanding LLMs: A Comprehensive Overview from Training to Inference". In: *arXiv preprint arXiv:2401.02038* (2024).

[14] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2019.

[15] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *arXiv preprint arXiv:2106.09685* (2021).

List of Figures

List of Tables