Graduation Thesis

To obtain the Industrial Engineering diploma

OPTION : DATA SCIENCE & ARTIFICIAL INTELLIGENCE

Unlocking Potential: Open Source LLMs in Rag Systems

Written by:

Ghribi Ouassim Abdelmalek

Jury:

President:

   Prof. — —   - National Polytechnic School of Algiers

Examiners:

   Prof. — —   - National Polytechnic School of Algiers

Supervisors:

   Mr. Oussama ARKI   - National Polytechnic School of Algiers

   Mr. Hakim FOURAR LAIDI   - National Polytechnic School of Algiers

   Mr. Hachem BETROUNIE   - BIGmama Technology

بِسْمِ ٱللّٰهِ ٱلرَّحْمٰنِ ٱلرَّحِيمِ

In the Name of God, Most Gracious, Most Merciful

Acknowledgments

I extend my deepest gratitude to my thesis supervisors, Mr. Arki Oussama and Mr. Betrouni Hachem, for their indispensable guidance and unwavering support throughout this project. Their expertise and continuous encouragement have been pivotal in shaping its direction. I am sincerely thankful for their patience, insightful feedback, and generous investment of time, all of which have been crucial in refining this work.

On a personal level, I am profoundly grateful to my family and friends for their unwavering support and encouragement throughout this demanding yet rewarding journey. Their unwavering belief in my abilities has been a constant source of motivation, particularly during challenging times.

Moreover, I would like to express my appreciation to all individuals who directly or indirectly contributed to the completion of this thesis. Your support and encouragement have been invaluable and deeply cherished.

Special recognition goes to Mr. Hadj Khelil for his belief in me over the past two years. Progressing from a junior front-end developer to a team lead, overseeing an exceptional team of ambitious individuals, has been an enriching experience. Together, we have strived to develop software with the potential to make a positive impact on the world.

Abstract

Table of Contents

List of Figures

List of Tables

Listings

Glossary

Here are the main acronyms used in this document. The meaning of an acronym is usually indicated once, when it first appears in the text.

ABC	Asset Based Consulting
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
LLM	Large Language model
GPT	Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
RAG	Retrieval-Augmented Generation
NN	Neural Networks
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
LSTM	Recurrent Neural Networks
AGI	Artificial General Intelligence
SLM	Statistical Language Model
NLM	Neural Language Model
PLM	Pre-trained Language Model
RLHF	Reinforcement Learning With Human Feedback

State of Play

"The true power of AI lies in its ability to augment human intelligence, not replace it."
Andrew Ng, Co-founder of Coursera and former Chief Scientist at Baidu

In this chapter, we embark on a journey through the landscape of existing literature and research pertinent to the convergence of open source Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) systems, in collaboration with BIGmama Technology. Our objective is to establish a robust foundation for understanding the intricacies and potentials inherent in this symbiotic relationship. As we delve into the state of play, we aim to illuminate key theoretical underpinnings and empirical insights that underpin our study of open source LLMs and Rag systems within the context of BIGmama Technology. To initiate this exploration, we contextualize our inquiry within the broader discourse of natural language processing and information retrieval, elucidating the multifaceted challenges and opportunities presented by the integration of open source LLMs and Rag systems. Subsequently, we navigate through seminal concepts and methodologies, including the principles of LLM architecture, Rag framework components, and their applications in various domains. These foundational insights serve as pillars supporting our quest to explore the potential synergies and advancements offered by the convergence of open source LLMs and Rag systems within the framework of BIGmama Technology.

1.1 Presenting BIGmama Technology

BIGmama is an innovative startup founded in France with a subsidiary in Algeria, specializing in data science and artificial intelligence solutions. With over 9 years of experience as of 2024, the company has established itself as a leading provider of bespoke predictive applications tailored to meet the unique needs of its clients.

Guided by a distinguished board of directors comprising former CEOs of global conglomerates such as Danone and Safran, BIGmama boasts a team of highly skilled data scientists and software engineers. This multidisciplinary team, consisting of more than a dozen experts in their respective fields, brings a wealth of knowledge and expertise to the company's endeavors.

BIGmama's commitment to innovation and cutting-edge technologies has allowed them to deliver state-of-the-art solutions that leverage the power of data science and

artificial intelligence. With a strong focus on providing customized solutions, the company has consistently exceeded its clients' expectations, enabling them to gain valuable insights and make data-driven decisions.

Through its strategic partnerships and collaborations, BIGmama continues to push the boundaries of what is possible in the realms of data science and AI (Artificial Intelligence), positioning itself as a driving force in the ever-evolving technological landscape.

1.1.1   Mission

Beyond speeches, BIGmama's mission is to propose effective methodologies, action plans, and tools to:

- Solve problems with artificial intelligence tools.

- Put people at the heart of technology, i.e., the hybridization of AI.

- Make technology a common good, shareable, and accessible to the maximum number of people who can participate and contribute.

The company's commitment extends far beyond mere rhetoric; it is dedicated to developing practical solutions, methodologies, and actionable plans to leverage the power of artificial intelligence for problem-solving. BIGmama places a strong emphasis on ensuring that technology remains human-centric, fostering a harmonious integration and hybridization of AI with human intelligence.

Moreover, BIGmama recognizes the importance of democratizing access to technology, ensuring that it becomes a shared resource that can benefit society as a whole. The company aims to create an environment where the maximum number of individuals can actively participate and contribute to the advancement of technology.

Ultimately, BIGmama's mission is to position technology as a catalyst for freedom, empowering individuals and communities to unlock new possibilities and overcome barriers through innovative solutions.

1.1.2   Vision

"Standing on the Shoulders of Giants"

We find ourselves at a pivotal moment in human history, where the rapid advancements in artificial intelligence are poised to reshape our world in unprecedented ways. Groundbreaking technologies like ChatGPT, developed by industry giants, are at the forefront of this transformative wave, mobilizing vast resources to redefine the boundaries of what is possible.

However, BIGmama's approach is not to compete directly with these titans but rather to harness their innovations as a springboard for its own technological endeavors. By building upon the foundations laid by these industry leaders, BIGmama aims to leverage their advancements as a catalyst for its own visionary project.

Rather than reinventing the wheel, BIGmama recognizes the value in standing on the shoulders of giants, capitalizing on their groundbreaking work to propel its own unique solutions forward. This strategic approach not only allows for more efficient progress but also fosters an environment of collaboration and synergy within the broader technological ecosystem.

With a keen understanding of the rapidly evolving landscape, BIGmama remains agile and adaptable, poised to seize opportunities presented by the advancements of industry giants. By embracing a collaborative mindset and harnessing the collective wisdom of the technological community, BIGmama is well-positioned to contribute its own innovative solutions, shaping the future in alignment with its vision and mission.

1.1.3   BIGmama Specificities

One of the valuable heritages that BIGmama acquired during the 9 years of actively developing bespoke AI applications to its clients is its unique methodology of work. This methodology is centered around the following ideas :

- **Data science starts with problematization:**

  At BIGmama, the approach to data science does not begin with data science itself but rather with reframing and problematization. The company's clients often arrive with a subject or topic rather than a clearly defined problem. BIGmama has developed a methodology to convert these topics into well-defined problems that data scientists can effectively tackle.

- **The data scientist is a "maverick":**

  BIGmama recognizes that the data scientist is a "maverick" who cannot be restrained. The company understands that it cannot impose specific tools or methods on data scientists when it comes to solving problems.

- **Data science a tool to solve problems:**

  BIGmama views data science not as an end in itself but as a means to solve problems. Often, the solution to their clients' problems lies outside the realm of data science tools.

- **Hybridization:**

  BIGmama believes that the future of AI lies in what is commonly referred to as Hybrid AI (Hybrid Artificial Intelligence). This approach encompasses a set of

methodologies and techniques aimed at combining the potential of models with purely human knowledge. The company believes that this hybridization allows for putting humans at the heart of technological development and producing tools that are more efficient, easier to maintain, and less expensive.

1.1.4   Products

Historically, BIGmama is a startup specialized in the development of predictive applications for third parties. The company's approach is specific and based on the principles of hybridization between human and model capabilities. These specificities have led BIGmama to develop its own internal tools. These tools have evolved as the company's needs have changed, transitioning from Iko[1] to Hyko[2]. BIGmama is now willing to make these tools available to the broader technology ecosystem.

As a startup with a rich history in predictive application development, BIGmama has cultivated a unique approach that emphasizes the seamless integration of human expertise and model capabilities. This distinctive methodology, rooted in the principles of hybridization, has driven the company to develop its own suite of proprietary internal tools.

Over time, as BIGmama's requirements have evolved, these tools have undergone continuous refinement and adaptation. The company has transitioned from its initial Iko platform to the more advanced Hyko solution, ensuring that its tools remain aligned with its ever-changing needs and the dynamic technological landscape.

In recognition of the value these tools hold for the broader tech community, BIGmama has made the strategic decision to make them available to the larger ecosystem. By sharing its internally developed tools, the company aims to foster collaboration, knowledge exchange, and the collective advancement of technological solutions.

1.1.5   Service: Asset based consulting (ABC)

The evolving business landscape, driven by rapid technological advancements and heightened competition, necessitates innovative solutions for optimizing operations and fostering sustainable growth. Traditional consulting approaches, such as report writing, are no longer sufficient to meet the demand for specialized solutions tailored to individual client needs. This shift has led to the emergence of asset-based consulting (ABC), a specialized field focused on creating, leveraging, and maximizing the value of a company's assets. In ABC, consultants collaborate closely with clients to identify and harness tangible and intangible assets, including technology, intellectual property, and human capital, to drive operational efficiency and gain a competitive edge. Unlike traditional consulting, ABC takes a holistic and internal perspective,

---

[1]https://big-mama.io/en/iko
[2]https://www.hyko.ai

recognizing the unique assets and capabilities of each organization. Consultants draw on multidisciplinary expertise to assess, diagnose, and unlock hidden value through innovative asset management strategies. As industries become increasingly complex and competitive, the demand for asset-based consulting, particularly in leveraging AI assets, continues to grow across sectors. From optimizing supply chains to capitalizing on intellectual property, asset-based consulting offers a strategic framework and expertise to help organizations achieve their objectives in a dynamic business environment.

1.1.6   Hyko

Hyko is a platform that empowers you to become an AI toolmaker. It can be thought of as using "Lego" pieces from its vast toolbox, filled with AI models (over 130,000), APIs (such as Stability.ai, OpenAI, Cohere), web scraping tools, utility functions, and more.

Hyko is a powerful platform that puts the tools and resources necessary for building AI applications at your fingertips. By offering a comprehensive toolbox and streamlined workflow, Hyko empowers users to unleash their creativity and bring their AI-powered visions to life.

Many companies grapple with numerous manual tasks performed repetitively by employees, often numbering in the hundreds or thousands each month. Traditional automation tools struggled to automate these tasks fully due to their reliance on human reasoning. However, LLMs have revolutionized automation by exposing reasoning capabilities through an API.

In response to this advancement, Hyko emerged as an AI-first platform, designed not merely to streamline employee workflows but to entirely replace them from start to finish. The platform's automation builder offers unparalleled flexibility, enabling the automation of highly complex workflows with ease.

The future value of AI is anticipated to derive not solely from AI engineers or data scientists, but also from field experts who possess the capability to integrate their domain knowledge with AI technologies to address everyday challenges. This shift underscores the growing importance of domain-specific expertise in harnessing the potential of AI for practical applications.

This imperative is frequently encountered among Hyko users, who often find themselves in need of extracting vital insights from their diverse data repositories, which encompass web sources, PDF documents, and email archives. In response to this demand, Rag systems emerge as key facilitators, particularly when coupled with open-source LLMs such as Llama3, Phi3, and mixtral7B. Rag systems excel in the retrieval, aggregation, and generation of pertinent information, empowering users to distill valuable insights from their data reservoirs efficiently and effectively. This
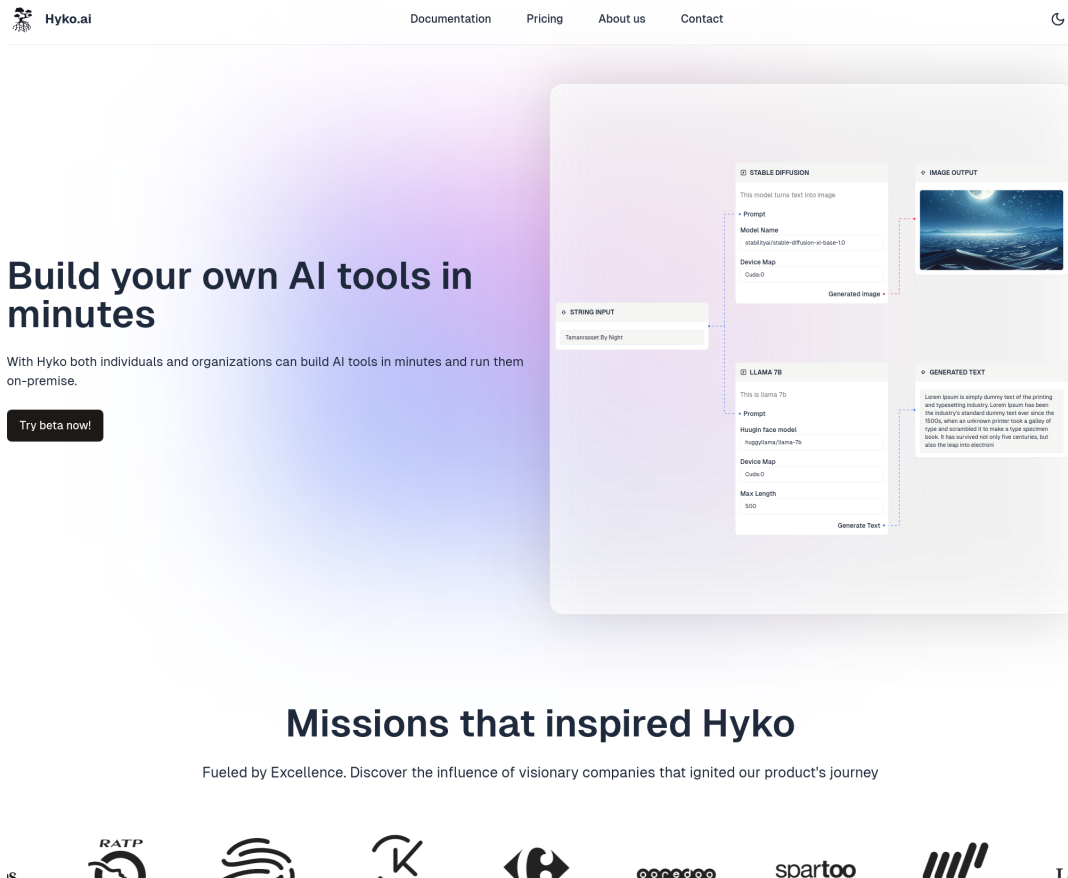
Figure 1.1: *Hyko's Landing page*

seamless integration enhances the capacity for nuanced understanding and actionable intelligence extraction, thereby augmenting the utility and versatility of Hyko across diverse information retrieval and analysis endeavors.

1.2   RAG: Enhancing LLMs with External Knowledge

LLMs showcase impressive capabilities but encounter challenges like hallucination [1], outdated knowledge, and non-transparent, untraceable reasoning processes. RAG has emerged as a promising solution by incorporating knowledge from external databases. This enhances the accuracy and credibility of the generation, particularly for knowledge-intensive tasks, and allows for continuous knowledge updates and integration of domain specific information. RAG synergistically merges LLMs' intrinsic knowledge with the vast, dynamic repositories of external databases [2].

General-purpose language models can be fine-tuned to achieve several common tasks such as sentiment analysis and named entity recognition. These tasks generally don't require additional background knowledge.

For more complex and knowledge-intensive tasks, it's possible to build a language model-based system that accesses external knowledge sources to complete tasks. This enables more factual consistency, improves reliability of the generated responses, and helps to mitigate the problem of "hallucination".

Meta AI researchers introduced a method called Retrieval Augmented Generation (RAG) [3] to address such knowledge-intensive tasks. RAG combines an information retrieval component with a text generator model. RAG can be fine-tuned and its internal knowledge can be modified in an efficient manner and without needing retraining of the entire model.

RAG takes an input and retrieves a set of relevant/supporting documents given a source (e.g., Wikipedia). The documents are concatenated as context with the original input prompt and fed to the text generator which produces the final output. This makes RAG adaptive for situations where facts could evolve over time. This is very useful as LLMs's parametric knowledge is static. RAG allows language models to bypass retraining, enabling access to the latest information for generating reliable outputs via retrieval-based generation.
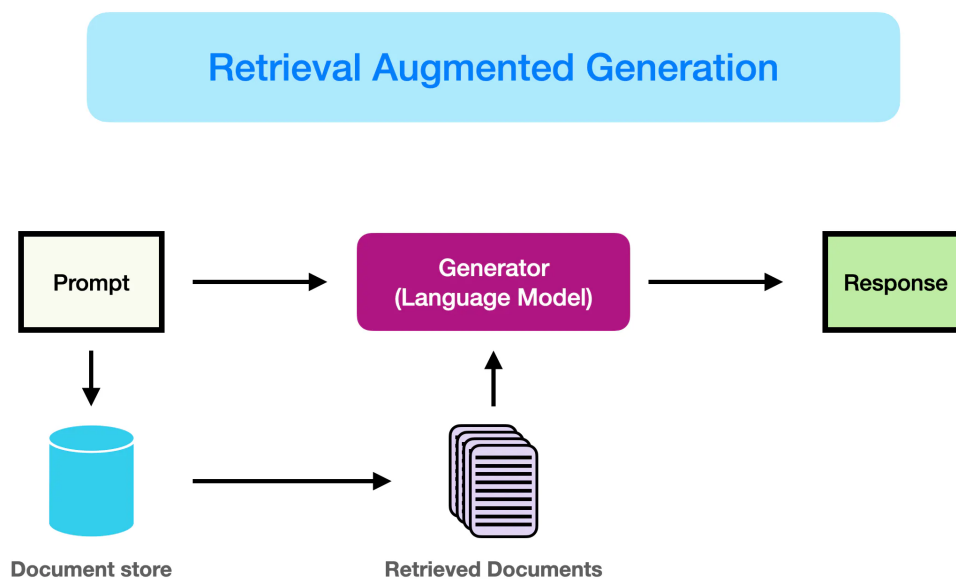
Figure 1.2: *Retrieval Augmented Generation.*

In short, the retrieved evidence obtained in RAG can serve as a way to enhance the accuracy, controllability, and relevancy of the LLM's response. This is why RAG can help reduce issues of hallucination or performance when addressing problems in a highly evolving environment.

1.3   Conclusion

In this chapter, we lay the foundation for the subsequent sections of this thesis. We introduce fundamental concepts in Retrieval-Augmented Generation (RAG) systems and Large Language Models (LLMs), along with a basic overview of the host company, BIGmama Technology.

CHAPTER 2

State of the Art

In recent years, the remarkable advancements in Natural Language Processing (NLP) have been primarily driven by the development of LLMs. These LLMs, such as GPT (Generative Pretrained Transformer)[4] and BERT (Bidirectional Encoder Representations from Transformers)[5], have demonstrated remarkable capabilities in understanding and generating human-like text. However, despite their impressive performance, LLMs still face challenges in effectively retrieving and incorporating relevant context for generating accurate and coherent responses.

Enter RAG systems, a novel approach that seeks to overcome the limitations of traditional LLMs by integrating retrieval mechanisms with generation models. RAG systems combine the strengths of both retrieval and generation techniques to enhance the quality and relevance of generated text.

This chapter provides a comprehensive exploration of RAG systems, delving into their architecture, components, training processes, applications, advantages, and challenges. We begin by establishing a foundational understanding of LLMs and their evolution, laying the groundwork for understanding the need for RAG systems. We then proceed to dissect the intricacies of RAG systems, discussing the role of retrieval in providing context and the role of generation in producing fluent responses.

Through detailed examination and analysis, we uncover the inner workings of RAG systems, exploring how retrieval and generation components interact within the architecture. Real-world applications and use cases of RAG systems across various domains are elucidated, demonstrating their potential to revolutionize tasks such as question answering, dialogue generation, and content creation.

Furthermore, we evaluate the advantages and limitations of RAG systems compared to traditional LLMs and other approaches in NLP. By examining performance metrics, challenges, and future directions, we gain insights into the transformative impact of RAG systems on the field of natural language processing.

In summary, this chapter serves as a comprehensive guide to RAG systems, offering readers a deep dive into one of the most promising advancements in NLP. As we navigate through the complexities and potentials of RAG systems, we pave the way for understanding their role in shaping the future of human-computer interaction and language understanding.

2.1   Machine Learning

2.1.1   Definition

Machine Learning, often abbreviated as ML, is a subset of artificial intelligence (AI) that focuses on the development of computer algorithms that improve automatically through experience and by the use of data. In simpler terms, machine learning enables computers to learn from data and make decisions or predictions without being explicitly programmed to do so [6].

At its core, machine learning is all about creating and implementing algorithms that facilitate these decisions and predictions. These algorithms are designed to improve their performance over time, becoming more accurate and effective as they process more data.

In traditional programming, a computer follows a set of predefined instructions to perform a task. However, in machine learning, the computer is given a set of examples (data) and a task to perform, but it's up to the computer to figure out how to accomplish the task based on the examples it's given.

For instance, if we want a computer to recognize images of cats, we don't provide it with specific instructions on what a cat looks like. Instead, we give it thousands of images of cats and let the machine learning algorithm figure out the common patterns and features that define a cat. Over time, as the algorithm processes more images, it gets better at recognizing cats, even when presented with images it has never seen before.

This ability to learn from data and improve over time makes machine learning incredibly powerful and versatile. It's the driving force behind many of the technological advancements we see today, from voice assistants and recommendation systems to self-driving cars and predictive analytics.

2.1.2   Relationships to other fields

Artificial Intelligence (AI) encompasses the field of computer science dedicated to creating systems capable of emulating human-like intelligence, problem-solving, and decision-making. Machine Learning (ML) is a subset of AI focused on enabling computers to learn from data without being explicitly programmed. Within ML, Deep Learning stands out as a subfield that employs neural networks with multiple layers to learn complex representations of data, particularly effective in tasks like image and speech recognition. ML and Deep Learning are integral components of AI, providing the framework for developing intelligent systems capable of learning, reasoning, and adapting to new information, thereby advancing the capabilities of AI across various domains.
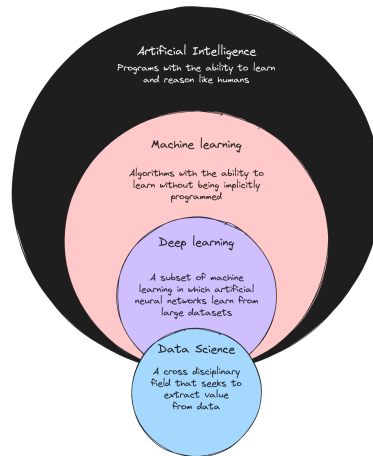
Figure 2.1: *Machine learning as subfield of AI [6].*

2.1.3 Types of Machine Learning

Machine Learning can be broadly categorized into several types based on the learning approach, the availability of labeled data, and the feedback mechanism. The main types of machine learning are:
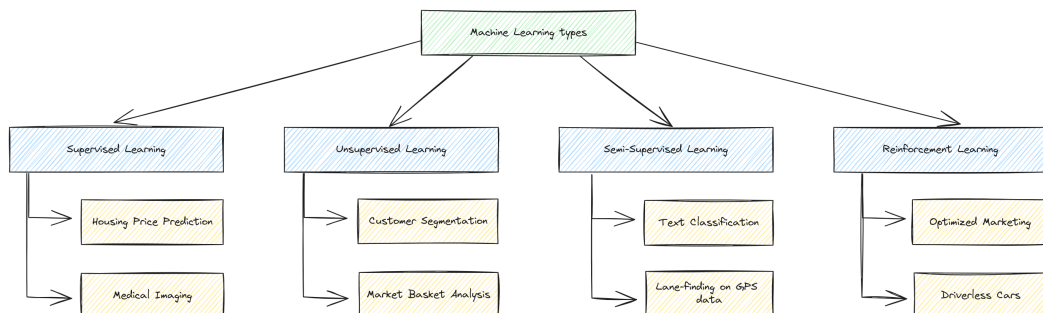
Figure 2.2: *Types of machine Learning.*

Supervised Learning

Supervised machine learning learns patterns and relationships between input and output data. It is defined by its use of labeled data. A labeled data is a dataset that contains a lot of examples of Features and Target. Supervised learning uses algorithms that learn the relationship of Features and Target from the dataset. This process is referred to as Training or Fitting [7].

Unsupervised Learning:

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Unsupervised learning models are utilized for three main tasks—clustering, association, and dimensionality reduction [8].

Semi-Supervised Learning:

Semi-supervised learning is a relatively new and less popular type of machine learning that, during training, blends a sizable amount of unlabeled data with a small amount of labeled data. Semi-supervised learning is between supervised learning (with labeled training data) and unsupervised learning (unlabeled training data).

Semi-supervised learning offers a lot of real-world applications. There is a dearth of labeled data in many fields. Because they involve human annotators, specialized equipment, or expensive, time-consuming studies, the labels (target variable) could be challenging to get [7].

Reinforcement Learning:

Reinforcement learning (RL) is a machine learning technique that trains software to make decisions to achieve the most optimal results. It mimics the trial-and-error learning process that humans use to achieve their goals [9].

2.1.4 Limitations of Machine Learning: Challenges and Considerations

Although machine learning is a powerful technique for extracting knowledge from data, it also has certain limitations that are important to consider:

- **Data dependency:** Machine learning heavily relies on the quality and quantity of training data. If the data is poorly labeled or unrepresentative, it can lead to errors in predictions.

- **Overfitting:** When the model is too complex compared to the training data, it can overfit and not generalize well to test data. This can also result in poor performance for new data.

- **Explainability:** Machine learning models can be very complex and difficult to understand. It can be challenging to understand how the model makes decisions and to explain these decisions to users.

- **Biased data:** Training data can be biased due to factors such as data selection, human biases, or measurement errors. This can lead to biased predictions for test data.

- **Lack of diversity:** Machine learning models may lack diversity in the types of data they can handle. For example, machine learning models may struggle to process unstructured data such as images, sounds, and texts.

- **Computational cost:** Machine learning algorithms may require high computational power and significant storage resources to process large amounts of data. This can be costly and time-consuming to train and implement models.

2.2 Deep Learning

2.2.1 Definition

Deep learning is a type of machine learning that teaches computers to perform tasks by learning from examples, much like humans do. Imagine teaching a computer to recognize cats: instead of telling it to look for whiskers, ears, and a tail, you show it thousands of pictures of cats. The computer finds the common patterns all by itself and learns how to identify a cat. This is the essence of deep learning.

In technical terms, deep learning uses something called "neural networks," which are inspired by the human brain. These networks consist of layers of interconnected nodes that process information. The more layers, the "deeper" the network, allowing it to learn more complex features and perform more sophisticated tasks [10].

2.2.2 Deep Learning vs. Machine Learning

Deep learning stands apart from traditional machine learning in its approach to data and learning methods. Machine learning algorithms typically rely on structured, labeled data for predictions, where specific features are defined and organized into tables. While machine learning can handle unstructured data, it often requires preprocessing to structure it. In contrast, deep learning streamlines this process by directly processing unstructured data such as text and images. It automates feature extraction, reducing reliance on human experts. For instance, in categorizing pet photos, deep learning algorithms autonomously identify key features, like ears, crucial for distinguishing between animals. In contrast, machine learning requires manual feature hierarchy establishment by human experts.

2.2.3 Deep Learning Applications

Deep learning has a wide range of applications across various domains due to its ability to learn complex patterns from large volumes of data. Some of the different

types of applications for deep learning include:

Image Recognition and Computer Vision:

- Deep learning is extensively used for tasks such as image classification, object detection, facial recognition, and image segmentation.

- Applications include self-driving cars, medical image analysis, surveillance systems, and augmented reality.

Natural Language Processing (NLP):

- Deep learning is employed for understanding and generating human language, enabling tasks such as sentiment analysis, machine translation, text summarization, and chatbots.

- Applications include virtual assistants, language translation services, social media sentiment analysis, and customer support chatbots.

These are just a few examples of the diverse applications of deep learning, demonstrating its versatility and impact across various industries and fields.

2.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) serves as a crucial technology within artificial intelligence, facilitating communication between humans and computers. It represents a multidisciplinary field empowering machines to comprehend, analyze, and produce human language, thus facilitating seamless human-machine interaction. The importance of NLP manifests in its diverse applications, spanning automated customer support to instantaneous language translation, showcasing its pivotal role in modern computing.

2.3.1 What is Natural Language Processing?

Natural Language Processing (NLP) is a sub-discipline of computer science providing a bridge between natural languages and computers. It helps empower machines to understand, process, and analyze human language. NLP's significance as a tool aiding comprehension of human-generated data is a logical consequence of the context-dependency of data. Data becomes more meaningful through a deeper understanding of its context, which in turn facilitates text analysis and mining. NLP enables this with the communication structures and patterns of humans [11].

NLP encompasses the task of enabling machines to comprehend, interpret, and generate human language in a manner that is not only valuable but also meaningful. OpenAI[1], renowned for pioneering sophisticated language models such as ChatGPT[2], underscores the significance of NLP in fostering the development of intelligent systems capable of comprehending, responding to, and generating text. This advancement in technology serves to enhance user-friendliness and accessibility across various applications.

2.3.2 How Does NLP Work?

NLP is a fascinating field that delves into the intricate mechanisms underlying human language comprehension and generation by machines. This section aims to unravel the complexities of NLP, shedding light on the fundamental principles and techniques that drive its functionality. By exploring the inner workings of NLP, we gain insight into how machines process and analyze natural language data, paving the way for groundbreaking applications in artificial intelligence and human-computer interaction. Through this exploration, we embark on a journey to discover the algorithms, models, and methodologies that empower machines to navigate the vast landscape of human language with precision and sophistication.

---

[1] https://openai.com/
[2] https://chat.openai.com

Components of NLP

Natural Language Processing is not a monolithic, singular approach, but rather, it is composed of several components, each contributing to the overall understanding of language. The main components that NLP strives to understand are Syntax, Semantics, Pragmatics, and Discourse.

Syntax: Syntax pertains to the arrangement of words and phrases to create well-structured sentences in a language.

Semantics: Semantics is concerned with understanding the meaning of words and how they create meaning when combined in sentences.

Pragmatics: Pragmatics deals with understanding language in various contexts, ensuring that the intended meaning is derived based on the situation, speaker's intent, and shared knowledge.

Discourse: Discourse focuses on the analysis and interpretation of language beyond the sentence level, considering how sentences relate to each other in texts and conversations.

NLP techniques and methods

NLP employs a diverse array of techniques and methodologies to analyze and comprehend human language. Below are some foundational techniques utilized in NLP:

Tokenization: This process involves segmenting text into individual units, such as words, phrases, or symbols, known as tokens.

Parsing: Parsing entails examining the grammatical structure of a sentence to extract its meaning and syntactic relationships.

Lemmatization: This technique involves reducing words to their base or root form, facilitating the grouping of different word forms with the same meaning.

Named Entity Recognition (NER): NER is utilized to identify and classify entities within text, such as persons, organizations, locations, and other named items.

Sentiment Analysis: This method enables the assessment of the sentiment or emotion expressed in a piece of text, aiding in understanding the underlying mood or opinion.

What is NLP Used For?

With some of the basic concepts now defined, one can explore how natural language processing is applied in the modern world.

Automatic Translation: Automatic translation systems use NLP techniques to translate texts from one language to another.

Chatbots and Virtual Assistants: Chatbots and virtual assistants use NLP to understand user's natural language and provide appropriate responses.

Automatic Summarization: NLP algorithms can be employed to summarize lengthy documents into a few sentences.

Sentiment Analysis: NLP is utilized to analyze sentiments expressed in text, which can be beneficial for businesses in assessing customer satisfaction.

Information Extraction: NLP systems can extract important information such as names, locations, and dates from texts.

Speech Recognition: Speech recognition systems utilize NLP techniques to convert speech into text.

Autocorrection: NLP algorithms are utilized in autocorrection programs to suggest grammatical and spelling corrections.

Text Analysis: NLP is used to analyze large volumes of text to detect trends, themes, and patterns.

Automatic Text Generation: NLP enables the automatic generation of text for various applications, such as report writing or content creation.

2.4   Neural networks

Neural Networks (NNs) are computational models composed of layers of neurons that can learn from data [12]. They are versatile and robust models capable of learning directly from raw data, without the need for manually selected features [13]. NNs employ a training algorithm known as backpropagation, which adjusts the model's parameters based on a loss function [13].

2.4.1   Feedforward networks

A feedforward neural network is characterized by its architecture, devoid of cycles, where the output of layer i can be computed using the outputs from layer $i - 1$ [13]. The architecture of a neural network encompasses its structure, including the number of hidden layers and neurons, as well as the functions employed for computations. These selections, known as hyperparameters, are parameters whose values are not determined by the learning algorithm [14]. Figure 2.3 shows a typical feedforward network architecture.
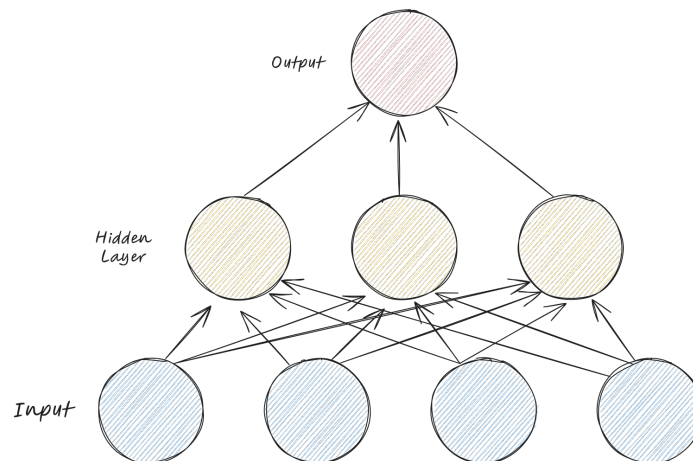

Figure 2.3: *Example of a feedforward neural network architecture. The network has an input layer with four neurons, one hidden layer with three neurons and an output layer with a single neuron [13].*

As depicted in Figure 2.3, neural networks comprise an input layer, n hidden layers, and an output layer. The input layer receives the data that the network needs to process, which then traverses through the hidden layers before reaching the output layer. The output layer provides the model's result for the given input data. When a neural network contains multiple hidden layers, it qualifies as a deep neural network (DNN) and falls within the domain of deep learning [12]. Each layer consists of i neurons, with connections between each neuron in a layer and every neuron in the previous layer, except for the input layer.

Neurons serve as fundamental computational units and, as previously indicated, establish connections with all neurons in the preceding layer. Each connection is characterized by a numerical parameter referred to as a weight [13].
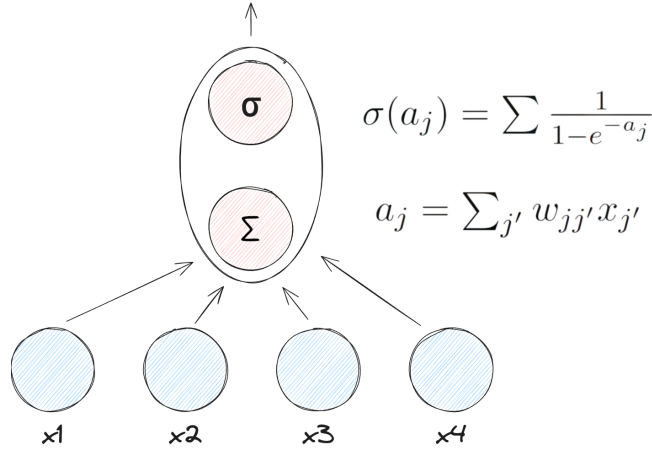

Figure 2.4: *Example of computation within a single neuron, where the weighted sum of inputs is passed through an activation function to get the output of the neuron [13].*

As show in Figure 2.4, the neuron receives inputs from neurons of the preceding layer, denoted as $x_1, x_2, ..., x_n$. Each neuron-to-neuron connection is assigned a weight, representing the strength of the connection. The output from a preceding neuron is multiplied by the weight of the connection and then summed for each connected neuron, yielding the weighted sum of values.

Although not visually represented in Figure 2.4, a bias parameter is introduced to the weighted sum, thereby altering a_j to

$$a_j = \sum_{j'} w_{jj'} x_{j'} + b \tag{2.1}$$

The bias serves to adjust the neuron's output independently of the input. This capability enables the model to alter the neuron's output during the learning process without necessitating adjustments to the weights, thereby facilitating finer control [14].

The weighted sum is subsequently forwarded to an activation function, which determines the neuron's output or its degree of "activation." An illustration of such an activation function is the sigmoid function denoted as σ, as depicted in Figure 2.4. The mathematical expression for the sigmoid function is:

$$\sigma(a_j) = \frac{1}{1 + e^{-a_j}} \tag{2.2}$$

According to Nielsen [14], two other common activation functions include $\tanh \phi$,

which is defined as

$$\phi(a_j) = \tanh(a_j) = \frac{e^{a_j} - e^{-a_j}}{e^{a_j} + e^{-a_j}} \tag{2.3}$$

and Rectified Linear Unit (ReLU), which is defined by equation

$$R(a_j) = max(0, a_j) \tag{2.4}$$

Symbol a_j refers to the result of adding together the weighted sum and bias. The outputs of these functions are visualized in Figure 2.5.
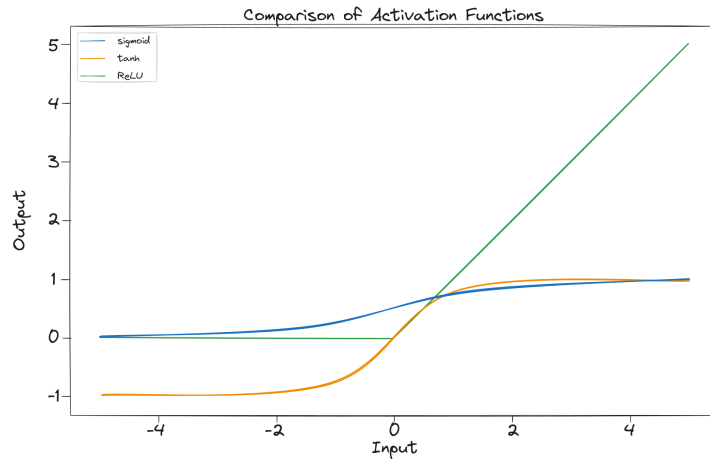

Figure 2.5: *Visualization of the sigmoid, tanh and ReLU activation function outputs.*

The outputs produced by these activation functions exhibit non-linearity, signifying that the input parameter does not linearly determine the output value, as evident from Figure 2.5. This concept holds significance in neural networks, enabling them to learn non-linear systems [15]. While several activation functions are commonly employed, ReLU has gained prominence in deep learning due to its demonstrated efficacy in enhancing the performance of numerous neural networks [13]. Typically, the same activation function is applied across all neurons, except for those in the output layer.

The choice of activation function for the output layer hinges on the specific task assigned to the neural network. In regression-based tasks, a linear activation function is employed to obtain the summed weighted output of the preceding neurons. Conversely, for classification problems involving k output neurons, a softmax function is often utilized. The softmax function is defined as:

$$\hat{y} = \frac{e^{a_k}}{\sum_{j=1}^{J} e^{a_j}} \tag{2.5}$$

and ensures that all neuron outputs sum to one on the output layer [13].

2.4.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are specialized neural networks designed for processing sequential data. They operate by incorporating iterations that retain past states, enabling the network to leverage previous inputs as context for the current input [16]. Text serves as a prime example of sequential data, where individual words can be viewed as singular data points within the sequence. Consequently, RNNs can effectively process textual input by utilizing preceding words to predict the subsequent ones. Figure 5 provides an illustration of a basic RNN architecture.
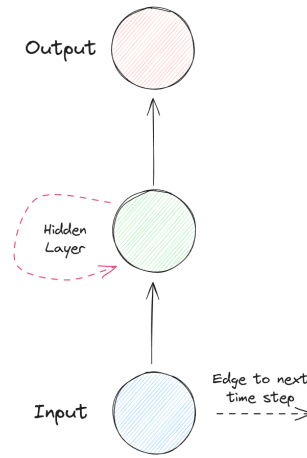

Figure 2.6: *Visualization of an RNN with one input neuron, one hidden neuron and one output neuron [13].*

The neuron on the hidden layer stores the hidden state, as visualized by the dashed line. The activation of the hidden state can be defined as

$$H_t = \phi_h(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \tag{2.6}$$

where H_t is the hidden state at time step t, ϕ_t is the activation function of the neuron, X_t is the neuron input at time step t, W_{xh} is a weight matrix, W_{hh} is the weight matrix of the hidden state and b_h is bias [16].

RNNs are trained utilizing the backpropagation through time (BPTT) algorithm, which is an adaptation of the standard backpropagation algorithm tailored for networks with a sequential order of computations. In this setup, the output at time step t depends on the states from preceding steps [17]. During training, the forward pass computes through all time steps, and the resultant loss is utilized to update the parameters across all time steps. One way to conceptualize this process is by envisioning an unrolled RNN, where the recurrent loops are eliminated, resulting in a network structure akin to a feedforward neural network. However, due to the nature of the hidden state dynamics, RNNs are notorious for encountering gradient-related

issues during training. As illustrated in Figure 2.7, if the weight associated with the dotted line is less than one, future values may decrease exponentially [13]. Conversely, weights that start to grow can lead to the exploding gradient problem [16]. When gradients become too small or too large, it can impede the training process of the model.
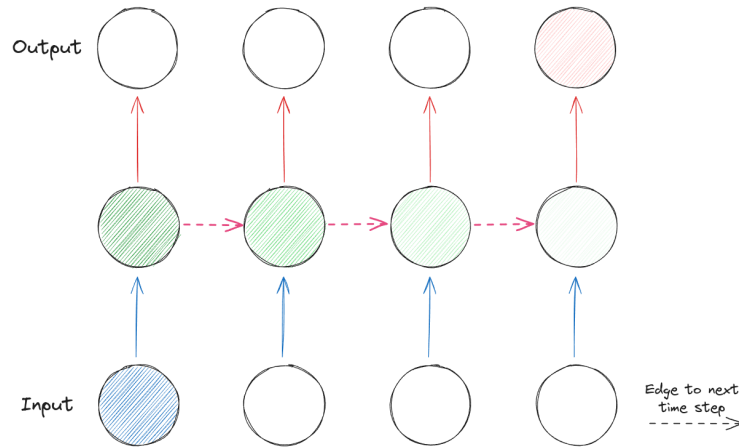

Figure 2.7: *Visualization of the vanishing gradient problem, which is indicated by the hidden neuron color fading away [13].*

Despite offering a robust framework for sequential learning, RNNs are hindered by significant limitations arising from the aforementioned issues. One critical limitation stems from the gradient problem encountered during training, which can constrain the effective handling of lengthy sequences. Consequently, RNNs may struggle to capture dependencies between inputs across extended sequences. To address these challenges, numerous techniques and alternative architectures have been proposed. Among these, the Long Short-Term Memory (LSTM) architecture stands out as one of the most prominent examples, and its discussion follows.

2.4.3   Long Short-Term Memory

The Long Short-Term Memory (LSTM) [18] architecture addresses the gradient-related challenges inherent in RNNs by incorporating constant error, memory cells, and gate units. This design enables the network to effectively handle sequences spanning over 1000 time steps. The memory cells employ three gate units: an input gate I_t, an output gate O_t, and a forget gate F_t, which regulate the flow of information. Specifically, the input gate facilitates the addition of information to the memory cell, the output gate facilitates information retrieval from the cell, and the forget gate facilitates cell resetting [16]. Moreover, the memory cells possess an internal state with a self-connected recurrent edge featuring a constant weight, ensuring consistent error propagation across time steps and mitigating previously discussed gradient-

related issues [13]. Additionally, a candidate memory cell $\tilde{C}$, utilized for proposing new information, is integrated with the old memory content C_{t-1} through the gates to govern the preservation of old memory in the new memory C_t [16]. Figure 2.8 illustrates the complete architecture of the LSTM memory cell.
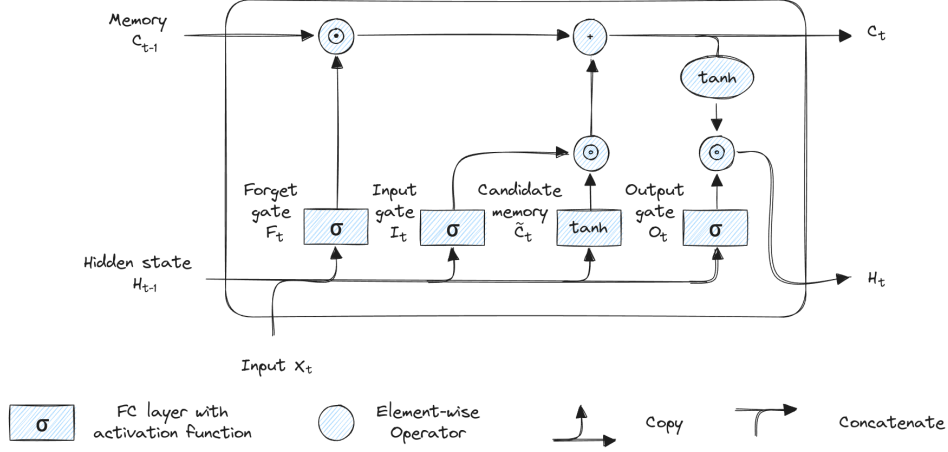

Figure 2.8: *Visualization of an LSTM memory cell [16].*

Although the LSTM architecture enhances performance compared to the RNN architecture discussed in subsection 2.4.2, it still possesses limitations. Like the RNN architecture, models utilizing LSTM must process the previous time step before computing the next, rendering them computationally intensive and impeding parallelization. Additionally, LSTM typically lacks an explicit attention mechanism and tends to prioritize the most recent words. These constraints are addressed by the Transformer model, which will be introduced next.

2.5   Transformer

LLMs are primarily based on the Transformer architecture, which has become the foundation for various state-of-the-art NLP models. In this section, we will discuss the main components of the Transformer architecture.

2.5.1   Transformer Architecture

The transformer architecture is a groundbreaking neural network architecture designed for NLP tasks. It was introduced by Vaswani et al. in the paper "Attention is All You Need"[19] The architecture relies on the self-attention mechanism to process and generate sequences, making it highly efficient and scalable compared to traditional RNNs and LSTM models. The problem in RNNs and LSTMs is that their network sequence makes it hard to process long sentences and, the ability to perform tasks in parallel is affected by sequential computation. Transformer approaches these problems by using encoders, decoders, and self-attention.
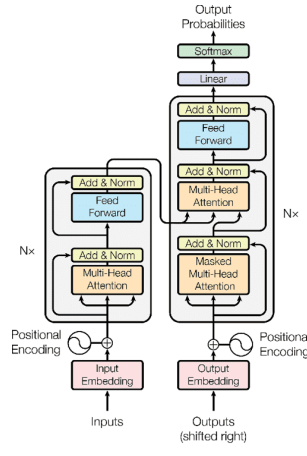The transformer model is shown in Figure 2.9 below.

Figure 2.9: *Transformer architecture model [19].*

2.5.2   Components of the Transformer Architecture

Figure 2.9 depicts the sequence flow beginning at the base of the embedding block, where input and output tokens undergo conversion into fixed-size vectors represented by numerical values corresponding to the tokens. Subsequently, positional encoding is applied to the tokens, assigning them position-specific values to track their sequential order. This step compensates for the absence of recursion and convolution within the network [19]. Progressing along the input pathway, the encoder stack follows, comprised of N identical layer structures, each containing two sub-layers: a multi-head attention network and a feedforward network. The output from the encoder

is then forwarded to the subsequent encoder. On the output side of the sequence, a decoder stack is present, also composed of N layers. Each decoder layer features a masked multi-head attention network for outputs, followed by multi-head attention incorporating encoder stack outputs. Analogous to the encoder, a feedforward network concludes the decoder. Normalization is performed between all sub-layers. Subsequently, the decoder output is linearized, and the softmax activation function is applied to transform output token values into a probability distribution. The resulting sequence yields a list of token probabilities, representing the transformer's predictions for suitable tokens given the input sequence.

2.5.3 Self-Attention Mechanism

In the article "Attention is all you need" the self-attention network is called Multi-Head Attention and consists of linearly projecting inputs, Scaled Dot-Product Attention layers, output concatenation, and finally, projecting the output values [19]. The self-attention network is introduced in Figure 2.10 below.
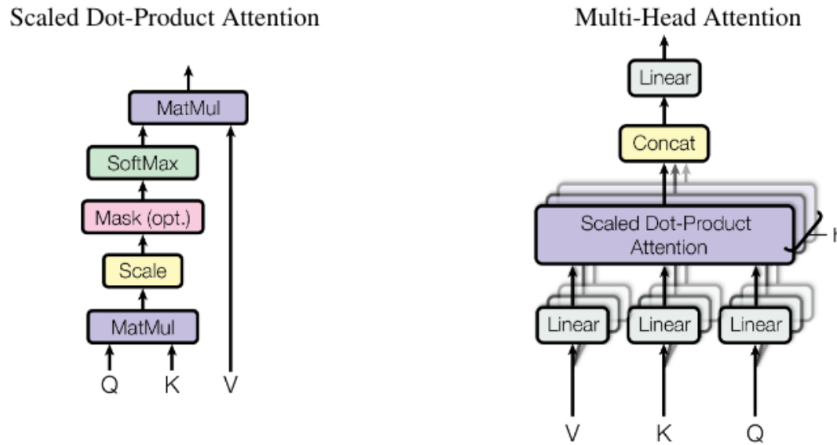

Figure 2.10: *Multi-Head attention network [19].*

The network's inputs comprise matrices containing query, key, and value vectors for each token. The matrix Q represents query vectors, K corresponds to keys, and V represents values. To simplify these terms, envision the query as akin to a search engine query initiated by a user, where the key value signifies the title of the search results, and the value encapsulates the content within the titles. These matrices correspond to sentences composed of tokens. Utilizing the scaled dot-product attention mechanism, the network produces tokens with associated attention scores. Multiple self-attention layers, denoted by h, operate concurrently, a process referred to as multi-head attention. This mechanism enables the model to track token positions effectively. The attention scores generated by self-attention can be expressed using

Equation 2.7.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt[2]{d_k}})V \tag{2.7}$$

The dot product, performed on matrices Q and K, is divided by the square root of d_k, representing the dimensionality of the key vectors [19]. This division serves as a scaling factor. Subsequently, the $softmax$ activation function is applied to this result, which is then multiplied by V to obtain the attention matrix. This process is repeated h times to generate multiple multi-head attention matrices. These matrices are subsequently concatenated into a single linearized matrix, which is then fed into the feedforward network.

The feedforward network employed within the transformer architecture shares similarities with the network outlined in Figure 2.9, but with variations in the number of layers, input and output dimensions, activation function, and other parameters across different models. In the "Attention is all you need" article, two linear transformations with ReLU activation were utilized, featuring output and input dimensions of $d_{model} = 512$, and an inner-layer dimensionality of 2048 [19]. Within the realm of LLMs, some of these parameters can be regarded as hyperparameters, exhibiting variability across models. For instance, in the Chinchilla model, the dimensionality is $d_{model} = 8192$, with the feedforward network size consistently four times the former value [20].

2.5.4   Variants of Transformer Architecture

Various LLMs are built on top of the Transformer architecture with slight modifications or adaptations. Some popular variants include:

GPT: The Generative Pre-trained Transformer (GPT) is an autoregressive model that utilizes only the decoder part of the Transformer architecture to generate text [4].

BERT: Bidirectional Encoder Representations from Transformers (BERT) is based on the encoder part of the Transformer architecture and is pre-trained using masked language modeling and next sentence prediction tasks [5].

T5: The Text-to-Text Transfer Transformer (T5) adapts the original Transformer architecture to a unified text-to-text format, enabling it to be used for various NLP tasks with minimal task-specific modifications [21].

2.6 Large language models (LLMs)

2.6.1 Introduction

Language modeling is a long-standing research topic, dating back to the 1950s with Shnnon's application of information theory to human language, where he measures how well simple n-gram language models predict or compress natural language text [22]. Since then, statistical language modeling became fundamental to many natural language understanding and generation tasks, ranging from speech recognition, machine translation, to information retrieval [23] [24].

Recent advances in transformer-based large language models (LLMs), pretrained on web-scale text corpora, have significantly enhanced the capabilities of these models. For instance, OpenAI's ChatGPT and GPT-4 are now used not only for natural language processing but also as general task solvers, exemplified by their integration into Microsoft's Co-Pilot systems. These models can follow complex human instructions and perform multi-step reasoning when necessary. Consequently, LLMs are becoming fundamental building blocks for the development of general-purpose AI agents or artificial general intelligence (AGI).

LLMs are large-scale, pre-trained, statistical language models based on neural networks. Their recent success is the result of decades of research and development in language modeling, which can be divided into four distinct waves, each with its own starting points and progression: statistical language models, neural language models, pre-trained language models, and LLMs.

Early **neural language models (NLMs)** [25], [26], [27], [28] address data sparsity by mapping words to low-dimensional continuous vectors, known as embedding vectors, and predicting the next word based on the aggregation of these vectors using neural networks. The embedding vectors learned by NLMs create a hidden space where the semantic similarity between vectors can be easily computed as their distance. This enables the computation of semantic similarity between any two inputs, regardless of their forms (e.g., queries versus documents in web search [29], [30], sentences in different languages in machine translation [31], [32]) or modalities (e.g., image and text in image captioning [32], [33]). However, early NLMs are task-specific models, as they are trained on task-specific data, resulting in a hidden space that is also task-specific.

Pre-trained language models (PLMs), unlike early neural language models (NLMs), are task-agnostic. This generality also extends to the learned hidden embedding space. The training and inference of PLMs follow the pre-training and fine-tuning paradigm, where language models utilizing recurrent neural networks [34] or transformers [5], [35], [36] are pre-trained on web-scale unlabeled text corpora for general tasks such as word prediction. They are then fine-tuned for specific tasks using small amounts of labeled, task-specific data. Recent surveys on PLMs include

[37], [38], [39].

Large language models (LLMs) mainly refer to transformer-based neural language models that contain tens to hundreds of billions of parameters. These models, such as PaLM [40], LLaMA [41], and GPT-4 [42], are pre-trained on massive text datasets. Compared to pre-trained language models (PLMs), LLMs are not only significantly larger in model size but also exhibit stronger language understanding and generation abilities. More importantly, they demonstrate emergent abilities not present in smaller-scale language models. As illustrated in Figure 2.11, these emergent abilities include:

In-context learning: LLMs can learn a new task from a small set of examples presented in the prompt at inference time.

Instruction following: After instruction tuning, LLMs can follow instructions for new types of tasks without explicit examples.

Multi-step reasoning: LLMs can solve complex tasks by breaking them down into intermediate reasoning steps, as demonstrated in the chain-of-thought prompting [43].

LLMs can also be augmented with external knowledge and tools [44], [44], enabling them to effectively interact with users and the environment [44], and continually improve using feedback data collected through interactions, such as via reinforcement learning with human feedback (RLHF).
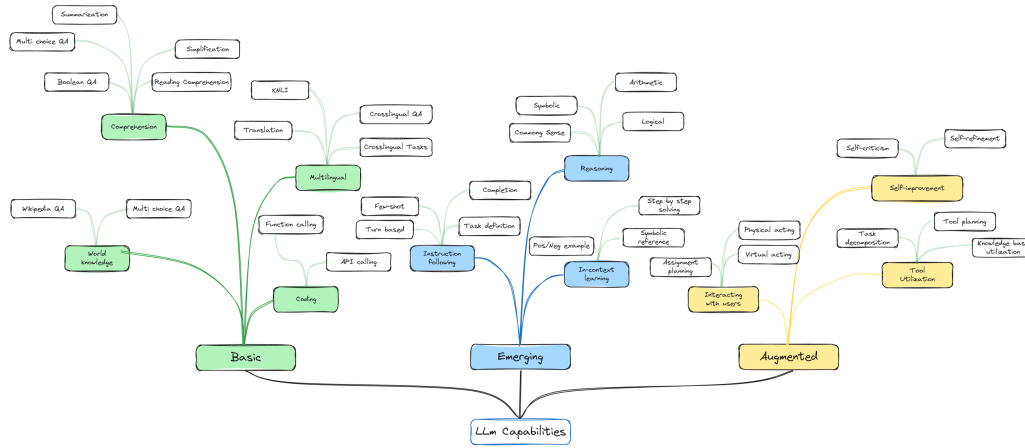

Figure 2.11: *LLM capabilities [45]*

Bibliography

[1] Yue Zhang et al. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models.* 2023. arXiv: 2309.01219 [cs.CL].

[2] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey.* 2024. arXiv: 2312.10997 [cs.CL].

[3] Sebastian Riedel et al. *Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models.* 2020. URL: %5Curl%7Bhttps://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/%7D.

[4] Alec Radford. *Improving language understanding with unsupervised learning.* https://openai.com/index/language-unsupervised. 2018.

[5] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019. arXiv: 1810.04805 [cs.CL].

[6] Matt Crabtree. *What is Machine Learning? Definition, Types, Tools & More.* https://www.datacamp.com/blog/what-is-machine-learning. 2023.

[7] Moez Ali. *Supervised Machine Learning.* 2022. URL: %5Curl%7Bhttps://www.datacamp.com/blog/supervised-machine-learning%7D.

[8] IBM. *Supervised Machine Learning.* URL: %5Curl%7Bhttps://www.ibm.com/topics/unsupervised-learning%7D.

[9] AWS. *What is Reinforcement Learning?* URL: %5Curl%7Bhttps://aws.amazon.com/what-is/reinforcement-learning/#:~:text=Reinforcement%20learning%20(RL)%20is%20a,use%20to%20achieve%20their%20goals.%7D.

[10] Abid Ali Awan. *What is Deep Learning? A Tutorial for Beginners.* https://www.datacamp.com/tutorial/tutorial-deep-learning-tutorial. 2023.

[11] Amirsina Torfi et al. *Natural Language Processing Advancements By Deep Learning: A Survey.* 2021. arXiv: 2003.01200 [cs.CL].

[12] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks.* 2015. arXiv: 1511.08458 [cs.NE].

[13] Zachary C. Lipton, John Berkowitz, and Charles Elkan. *A Critical Review of Recurrent Neural Networks for Sequence Learning.* 2015. arXiv: 1506.00019 [cs.LG].

[14] Michael Nielsen. *Neural Networks and Deep Learning*. 2015. URL: %5Curl%7Bhttps://www.ise.ncsu.edu/fuzzy-neural/wp-content/uploads/sites/9/2022/08/neuralnetworksanddeeplearning.pdf%7D.

[15] S. A. BILLINGS S. CHEN and P. M. GRANT. "Non-linear system identification using neural networks". In: *International Journal of Control* (). URL: https://doi.org/10.1080/00207179008934126.

[16] Robin M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: 1912.05911 [cs.LG].

[17] P.J. Werbos. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560. DOI: 10.1109/5.58337.

[18] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[19] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[20] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models*. 2022. arXiv: 2203.15556 [cs.CL].

[21] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG].

[22] C. E. Shannon. "Prediction and entropy of printed English". In: *The Bell System Technical Journal* 30.1 (1951), pp. 50–64. DOI: 10.1002/j.1538-7305.1951.tb01366.x.

[23] Frederick Jelinek. "Statistical methods for speech recognition". In: 1997. URL: %5Curl%7Bhttps://api.semanticscholar.org/CorpusID:12495425%7D.

[24] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[25] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. "A Neural Probabilistic Language Model". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press, 2000. URL: %5Curl%7Bhttps://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf%7D.

[26] Holger Schwenk, Daniel Dechelotte, and Jean-Luc Gauvain. "Continuous Space Language Models for Statistical Machine Translation". In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 723–730. URL: `%5Curl%7Bhttps://aclanthology.org/P06-2093%7D`.

[27] Tomáš Mikolov et al. "Recurrent neural network based language model". In: *Proc. Interspeech 2010*. 2010, pp. 1045–1048. DOI: `10.21437/Interspeech.2010-343`.

[28] Alex Graves. *Generating Sequences With Recurrent Neural Networks*. 2014. arXiv: `1308.0850 [cs.NE]`.

[29] Po-Sen Huang et al. "Learning deep structured semantic models for web search using clickthrough data". In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. CIKM '13. San Francisco, California, USA: Association for Computing Machinery, 2013, pp. 2333–2338. ISBN: 9781450322638. DOI: `10.1145/2505515.2505665`. URL: `%5Curl%7Bhttps://doi.org/10.1145/2505515.2505665%7D`.

[30] Jianfeng Gao et al. *Neural Approaches to Conversational Information Retrieval*. 2022. arXiv: `2201.05176 [cs.IR]`.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: `1409.3215 [cs.CL]`.

[32] Kyunghyun Cho et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. arXiv: `1409.1259 [cs.CL]`.

[33] Oriol Vinyals et al. *Show and Tell: A Neural Image Caption Generator*. 2015. arXiv: `1411.4555 [cs.CV]`.

[34] Matthew E. Peters et al. *Deep contextualized word representations*. 2018. arXiv: `1802.05365 [cs.CL]`.

[35] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: `1907.11692 [cs.CL]`.

[36] Pengcheng He et al. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. arXiv: `2006.03654 [cs.CL]`.

[37] Ce Zhou et al. *A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT*. 2023. arXiv: `2302.09419 [cs.AI]`.

[38] Xu Han et al. *Pre-Trained Models: Past, Present and Future*. 2021. arXiv: `2106.07139 [cs.AI]`.

[39]   XiPeng Qiu et al. "Pre-trained models for natural language processing: A survey". In: *Science China Technological Sciences* 63.10 (Sept. 2020), pp. 1872–1897. ISSN: 1869-1900. DOI: 10.1007/s11431-020-1647-3. URL: http://dx.doi.org/10.1007/s11431-020-1647-3.

[40]   Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways.* 2022. arXiv: 2204.02311 [cs.CL].

[41]   Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models.* 2023. arXiv: 2302.13971 [cs.CL].

[42]   OpenAI et al. *GPT-4 Technical Report.* 2024. arXiv: 2303.08774 [cs.CL].

[43]   Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.* 2023. arXiv: 2201.11903 [cs.CL].

[44]   Grégoire Mialon et al. *Augmented Language Models: a Survey.* 2023. arXiv: 2302.07842 [cs.CL].

[45]   Shervin Minaee et al. *Large Language Models: A Survey.* 2024. arXiv: 2402.06196 [cs.CL].