PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF SCIENCE AND TECHNOLOGY HOUARI BOUMEDIENE

ACADEMIC PRESENTATION

Field : Computer Science

Topic: Large Language Models (LLMs)

Presented by:

- Sellami Mohamed Amine
- Alem Mohamed
- Bouabache Malak
- Fettal Nesrine

Supervised by:

Mme . Oulefki

Academic Year: 2024 – 2025

Department: Computer Science

USTHB – Algiers

Report of the AI presentation on the topic of LLMs

Introduction:

Large Language Models (LLMs) are specialized in natural language understanding and natural language text generation. They are widely used in the AIs we use every day.
Capable of writing texts, translating into several languages and conducting conversations, they represent a major advance in the way machines understand and interact with humans and what they write. In particular, they use Machine Learning and Deep Learning to operate.
This is the basis on which ChatGPT works, for example

Definition:

What is a Large Language Model?

A Large Language Model (LLM), is an advanced form of artificial intelligence (AI) model specialized in natural language processing (NLP). LLMs are essentially deep neural networks (**transformer model)**, capable of understanding, interpreting, and generating human language.

They are called "large" due to the sheer volume of data used for their training. LLMs are often based on so-called "transformer" architectures and are trained on immense textual datasets, ranging from literature to online content, news, and social media.

it is a computer program fed with enough examples to learn how to recognize and interpret human language or other types of complex data. However, the quality of the samples significantly affects how well the model learns natural language. For this reason, developers tend to use more structured datasets to improve learning outcomes.

Due to their size and complexity, LLMs can perform a variety of natural language tasks, from text generation and classification to conversational question answering and text translation from one language to another.

LLMs employ a form of machine learning called **deep learning**, which involves probabilistic analysis of unstructured data. This allows the model to distinguish between different content elements without human intervention,Among the most popular LLMs are:

GPT-4 (Generative Pre-trained Transformer 4): Developed by OpenAI, GPT-4 is one of the most recent and advanced versions of the GPT series. It is the model that powers Chatgpt, as well as Microsoft's Copilot and numerous specialized AI tools.

Mistral 7B: Developed by Mistral AI, a French startup. It is a language model designed to offer high performance with an optimized architecture. It is capable of processing and generating text efficiently, even with a relatively small number of parameters (7 billion). Furthermore, Mistral 7B is open source, which is a major selling point for clients who would like to host the model on-site.

Types of Large Language Models:

1. Autoregressive language models (≃ Causal LMs):

Autoregressive models generate text by predicting the next word given the preceding words in a sequence. Models such as GPT-3 fall into this category. Autoregressive models are trained to maximize the likelihood of generating the correct next word, conditioned by context. While they excel at generating coherent and contextually relevant text, they can be computationally expensive and may suffer from generating repetitive or irrelevant responses.

Example: GPT-3

2. Transformer-based models:

Transformers are a type of deep learning architecture used in large language models. The transformer model, introduced by Vaswani et al. in 2017 is a key component of many LLMs. This transformer architecture allows the model to process and generate text effectively, capturing long-range dependencies and contextual information.

Example: RoBERTa (Robustly Optimized BERT Pretraining Approach) by Facebook AI

3. Encoder-decoder models:

Encoder-decoder models are commonly used for machine translation, summarization, and question-answering tasks. These models consist of two main components: an encoder that reads and processes the input sequence and a decoder that generates the output sequence. The encoder learns to encode the input information into a fixed-length representation, which the decoder uses to generate the output sequence. The transformer-based model known as the 'Transformer' is an example of an encoder-decoder architecture.

Example: MarianMT (Marian Neural Machine Translation) by the University of Edinburgh

4. Pre-trained and fine-tuned models:

Many large language models are pre-trained on large-scale datasets, enabling them to understand language patterns and semantics broadly. These pre-trained models can then be fine-tuned on specific tasks or domains using smaller task-specific datasets. Fine-tuning allows the model to specialize in a particular task, such as sentiment analysis or named entity recognition. This approach saves computational resources and time compared to training a large model from scratch for each task.

Example: ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately)

5. Multilingual models:

Multilingual models are trained on text from multiple languages and can process and generate text in several languages. They can be useful for tasks such as cross-lingual information retrieval, machine translation, or multilingual chatbots. By leveraging shared representations across languages, multilingual models can transfer knowledge from one language to another.

Example: XLM (Cross-lingual Language Model) developed by Facebook AI Research

6. Hybrid models:

Hybrid models combine the strengths of different architectures to achieve improved performance. For example, some models may incorporate both transformer-based architectures and recurrent neural networks (RNNs). RNNs are another type of neural network commonly used for sequential data processing. They can be integrated into LLMs to capture sequential dependencies in addition to the self-attention mechanisms of transformers.

Example: UniLM (Unified Language Model) is a hybrid LLM that integrates both autoregressive and sequence-to-sequence modeling approaches

These are just a few examples of the different types of large language models developed. Researchers and engineers continue to explore new architectures, techniques, and applications to advance the capabilities of these models further and address the challenges of natural language understanding and generation.

Examples of Large Language Models:

Several notable examples of large language models that have been developed are available, each with its unique characteristics . Here are a few prominent examples.

1. GPT-4:

GPT-4 is an advanced version of its predecessors, GPT-3 and GPT-3.5. It outperforms the previous models regarding creativity, visual comprehension, and context. This LLM allows users to collaborate on projects, including music, technical writing, screenplays, etc. Besides text, GPT-4 can accept images as input. Moreover, according to OpenAI, GPT-4 is a multilingual model that can answer thousands of questions across 26 languages. When it comes to the English language, it shows a staggering 85.5% accuracy, while for Indian languages such as Telugu, it shows 71.4% accuracy.

2. BERT (Bidirectional Encoder Representations from Transformers):

BERT, developed by Google, introduced the concept of bidirectional pre-training for LLMs. Unlike previous models that relied on autoregressive training, BERT learns to predict missing words in a sentence by considering both the preceding and following context. This bidirectional approach enables BERT to capture more nuanced language dependencies. BERT has been influential in tasks such as question-answering, sentiment analysis, named entity recognition, and language understanding. It has also been fine-tuned for domain-specific applications in industries such as healthcare and finance.

3. T5 (Text-to-Text Transfer Transformer):

T5, developed by Google, is a versatile LLM trained using a text-to-text framework. It can perform a wide range of language tasks by transforming the input and output formats into a text-to-text format. T5 has achieved state-of-the-art results in machine translation, text summarization, text classification, and document generation. Its ability to handle diverse tasks with a unified framework has made it highly flexible and efficient for various language-related applications.

4. XLNet (eXtreme Language Understanding):

XLNet, developed by researchers from Carnegie Mellon University and Google, addresses some limitations of autoregressive models such as GPT-3. It leverages a permutation-based training approach that allows the model to consider all possible word orders during pre-training. This helps XLNet capture bidirectional dependencies without needing autoregressive generation during inference. XLNet has demonstrated impressive performance in tasks such as sentiment analysis, Q&A, and natural language inference.

5. Turing-NLG:

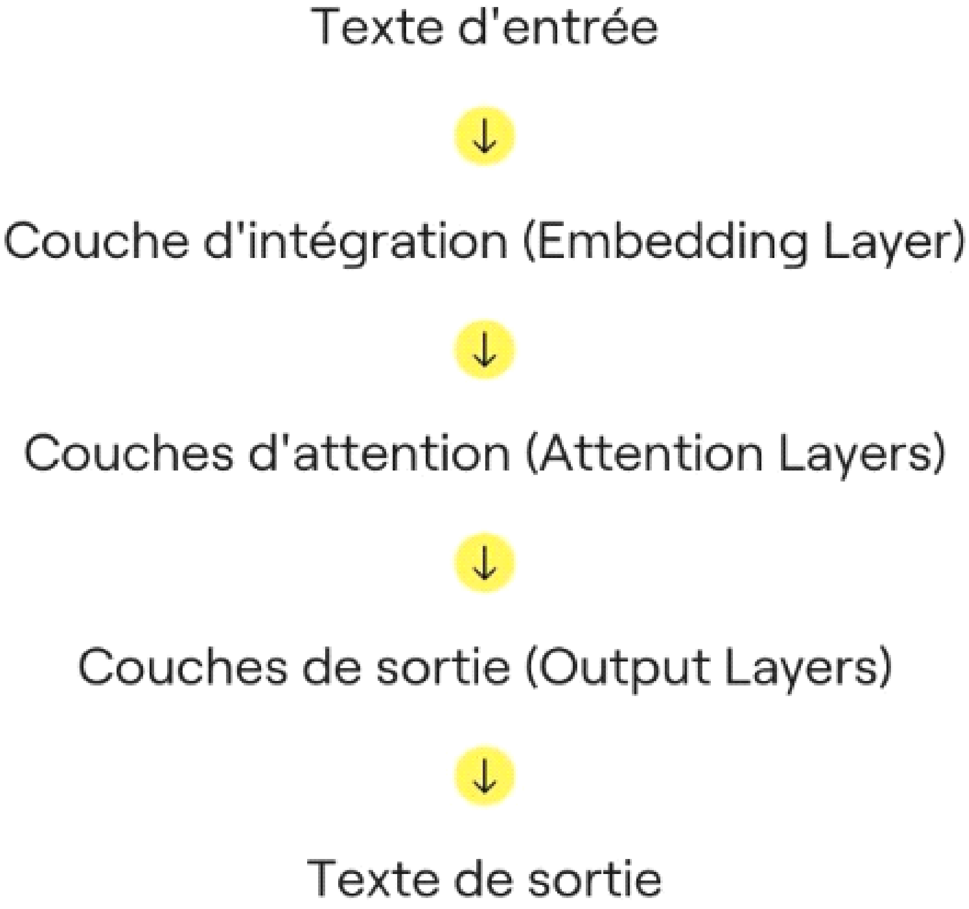
Turing-NLG, developed by Microsoft, is a powerful LLM that focuses on generating conversational responses. It has been trained on a large-scale dataset of dialogues to improve its conversational abilities. Turing-NLG performs well in chatbot applications, providing interactive and contextually appropriate responses in conversational settings.

These examples showcase the capabilities of LLMs in various language-related tasks and their potential to revolutionize NLP applications. Continued research and development in this field will likely bring further advancements and refinements to LLMs in the future.

Architecture of LLMs:

When we talk about architecture, we refer to the technical principles that allow LLMs to function.

Architecture des LLMs



Texte d'entrée

↓

Couche d'intégration (Embedding Layer)

↓

Couches d'attention (Attention Layers)

↓

Couches de sortie (Output Layers)

↓

Texte de sortie

The architecture of LLMs is generally based on **transformer models**, which are deep neural networks. They consist of several key components, including:

- **Embedding layers**: These layers convert words into numerical vectors, allowing the models to analyze textual data effectively.

- **Attention layers**: These layers enable the model to focus on the most relevant parts of the text when generating responses.
- **Output layers**: These produce the model's final predictions.

LLMs contain a large number of **parameters**, which are adjustable values within the model. These parameters are fine-tuned during training to improve the model's performance.

The size of LLMs can vary greatly, from a few million to hundreds of billions of parameters. The more parameters a model has, the more capable it is of performing complex tasks and generating accurate, relevant responses.

However, this precision comes at a cost:
A larger number of parameters increases the **computational power** required to run the model. Additionally, it affects the **fine-tuning process**, which involves adapting a pre-trained model to a specific dataset or task. A model with more parameters may require more **resources and time** for this process.

LLMs training phase:

Training large language models is not a singular process. Rather, it's a multi-layered stack of training processes, each with its unique role and contribution to the model's performance.The three main phases are:

- self-supervised learning
- supervised learning
- reinforcement learning

Phase 1: self-supervised learning for language understanding

Self-supervised learning, the first stage of training, is what traditionally comes to mind when we talk about language modeling.

It involves feeding the model with vast amounts of unannotated or raw data and having it predict ‘missing’ pieces of it. The model learns something about language and the domain of the data to generate plausible answers.

For example, if we feed the model with text from a weather website and ask it to predict the next word, it has to understand something about the language and the weather domain. In the presentation, I gave the example sentence “A flash flood watch will be in effect all _____.”

In an intermediate state, the model ranks a list of predictions, from likely answers (“day”, “night”, “hour”) to less plausible ones (“month”), and even nonsensical ones (“giraffe”) that should be assigned very low probability. This process is called self-supervision (as opposed to unsupervised learning) because there is a specific right answer—the word that actually appeared in the text we collected—which was “night” in my example. Self-supervision is similar to unsupervised learning in that it can use abundant, unannotated data.

Phase 2: Supervised learning for instruction understanding

Supervised learning, also known as instruction tuning, is the second stage in the training process of large language models (LLMs). It’s a crucial phase that builds upon the foundational knowledge acquired during the self-supervised learning stage.

In this phase, the model is explicitly trained to follow instructions. This goes beyond the basic prediction of words and sentences, which is the main focus of the self-supervised learning stage. The model now learns to respond to specific requests, making it far more interactive and useful.

The effectiveness of instruction tuning in enhancing the capabilities of LLMs has been demonstrated in various studies, several of which included Snorkel researchers. One notable outcome was that the model showed improved performance in generalizing to new, unseen tasks. This is a significant achievement as one of the main objectives of machine learning models is to perform well on unseen data.

Due to its proven effectiveness, instruction tuning has become a standard part of LLM training. With the completion of the instruction tuning phase, the model is now explicitly trained to be a helper, doing more than just predicting the next words and sentences. It’s now ready to interact with users, understand their requests, and provide helpful responses.

Phase 3: reinforcement learning to encourage desired behavior

The final stage in the training stack is reinforcement learning. This encourages desired behavior and discourages unwanted outputs. This stage is unique as it does not provide the model with exact outputs to produce, but rather grades the outputs it generates.

The concept of reinforcement learning predates LLMs, but Open AI first proposed it in the context of LLM training shortly after the introduction of instruction tuning. The process starts with a model, already enriched with the ability to follow instructions and predict language patterns. Next, data scientists use human annotations to distinguish between better and worse outputs. These data annotations serve as a guideline for the model, helping it understand which responses are preferred and which are not. The feedback from these annotations is then used to train a reward model.

The reward model is a critical component of this process. It provides rewards at scale, effectively guiding the model towards producing more desirable responses and discouraging less desirable ones. This method is particularly effective in promoting fuzzier concepts like brevity and discouraging harmful language, enhancing the overall quality of the language model’s output.

This approach to reinforcement learning is often referred to as reinforcement learning with human feedback. It emphasizes the importance of human involvement in the training process, ensuring that the model's learning is aligned with users' expectations.

How does LLMs work:

- **Tokenization**: The input text is broken down into smaller units called *tokens* (words or subwords), making it easier for the model to process language in parts.

- **Embedding**: Each token is converted into a numerical vector through an *embedding layer*, allowing the model to represent and understand the meaning of words mathematically.

- **Positional Encoding**: The model adds information about word order to each token's vector, since it doesn't inherently understand the sequence of words. This helps it differentiate between sentences like "the cat chased the dog" and "the dog chased the cat."

- **Self-Attention Mechanism**: The model examines how each word relates to the others in the sentence. This mechanism lets it focus on the most relevant words depending on the context, improving comprehension.

- **Feedforward Layers & Normalization**: These layers process and refine the information from attention. They apply mathematical transformations and stabilize learning to ensure the model builds accurate representations of the input.

- **Prediction**: The model uses the refined information to predict the next token in a sequence, generating text step by step in a coherent and context-aware way.

- **Training (Backpropagation)**: During training, the model compares its predictions to the correct words, calculates the error, and updates its internal weights to improve over time.

After being trained on massive datasets, the model can understand and generate human-like text for a wide variety of tasks, including translation, summarization, question answering, and more.

Limitations and Challenges:

- **Computational Constraints:** LLMs have a limited context window and can't process large texts all at once.

- **Hallucinations:** They may generate incorrect or made-up information that seems believable.

- **Outdated Knowledge:** Their knowledge is fixed up to a certain point in time and does not update automatically.

- **No Long-Term Memory:** They don't retain information from past interactions unless specifically designed to.

- **Limited Reasoning:** They struggle with tasks that require deep logic or multi-step reasoning.

- **Inconsistencies:** They can produce contradictory or internally inconsistent responses.

- **Lack of True Understanding:** They don't genuinely understand meaning, emotions, humor, or context.

Future of Large Language Models (LLMs):

The future of LLMs focuses on improving **accuracy**, **efficiency**, and **ethics**. Key developments include:

- **Real-Time Fact-Checking:** Future LLMs will connect to live data sources to provide accurate, up-to-date answers with references.
  Example: Microsoft Copilot integrates GPT-4 with internet access.

- **Self-Training with Synthetic Data:** LLMs will generate their own training data and improve performance without constant human supervision.
  Example: Google's model increased accuracy by creating its own Q&A for training.

- **Sparse Models (Efficient Specialization):** Instead of using the whole network, sparse models activate only the relevant parts, making them faster and more efficient.

- **Domain-Specific Fine-Tuning:** Future LLMs will be customized for specific industries like healthcare, finance, or law to reduce hallucinations and improve precision.
  Examples: Med-PaLM 2 (healthcare), BloombergGPT (finance), ChatLAW (legal domain).

REAL-LIFE APPLICATIONS OF LLMs:

1. Content generation

LLM applications are especially good at content generation. They can be used to automatically create texts for various purposes, including articles, blog posts, marketing copy, video scripts, and social media updates. Moreover, LLM-backed generative AI apps can adapt to

different writing styles and tones, making them versatile for generating content that resonates with specific target audiences.

Real-world app

Chatgpt

Digital marketers, for example, turn to ChatGPT to craft compelling ad copy, blog posts, and social media content that captivates their specific target audience. Similarly, educators find ChatGPT helpful in creating instructional materials, quizzes, and interactive learning modules, making the educational content both informative and engaging for students.

2. Translation and localization

LLM applications can provide accurate, context-aware translations across numerous language pairs. These models are trained on vast collections of bilingual or multilingual text, allowing them to understand nuances, idioms, and grammatical structures of different languages. They can maintain the intent and style of the original text, which is crucial for literary translations, business communications, and legal documents.

As for localization, LLMs help adapt content culturally and contextually for different target audiences, ensuring that the translated material is culturally appropriate and resonant. They consider local customs, measurements, date formats, and cultural references, making the content relevant and accessible. This capability is particularly important in marketing and entertainment industries, where engagement heavily depends on cultural nuance.

Real-world apps

Let’s take a look at two large language model applications — Falcon LLM and NLLB-200.

Falcon LLM :

Falcon LLM, developed by the Technology Innovation Institute (TII), is an open-source AI model. In general, Falcon LLM excels across a spectrum of activities, including reasoning, programming, skill assessments, and knowledge evaluations.

NLLB-200 :

NLLB-200 is an artificial intelligence model introduced by Meta AI. It translates across 200 different languages, incorporating many that were previously unsupported or poorly served by existing translation tools, and notably includes support for 55 African languages.

3. Search and recommendation

LLMs are capable of understanding and processing natural language queries with unprecedented accuracy and context. When integrated into search engines, these models can interpret the intent behind a user's query and deliver more relevant and precise results. They can also generate summaries of content, making it easier for users to find the information they need quickly.

To fully leverage LLMs in mobile applications, it's essential to invest in both iOS app development services and Android app development as this ensures that your applications are optimized for performance and user experience across different platforms.

Real-world apps

Let's explore Bard — a compelling example of how LLM applications are enhancing search systems.

Bard :

Developed by Google and launched in March 2023, Bard is a good example of an LLM application in search. Originating from the LaMDA family and subsequently upgraded to PaLM and Gemini, Bard was introduced as a response to the rise of OpenAI's ChatGPT.

As a research LLM, Bard leverages Google's extensive knowledge base and predictive capabilities to generate responses, offering creative and flexible answers to user prompts.

4. Virtual assistants

LLMs for virtual assistants

At the core of AI-powered virtual assistants are LLMs that understand and process natural language. When a user asks a question or gives a command, the LLM interprets the intent and context of the request. Once the intent is understood, the LLM generates an appropriate response.

Modern virtual assistants also learn from interactions to provide personalized responses and improve over time. They analyze feedback, remember users' preferences, and adapt to their unique way of communication.

Here is what LLM-based virtual assistants do specifically :

Performing tasks : Virtual assistants can perform a variety of tasks, from setting alarms and reminders to booking appointments, sending messages, or even ordering groceries. They interact with other applications and services to execute these tasks on behalf of the user.

Providing information : They can answer questions and provide information on a wide range of topics like weather forecasts, news, and traffic updates. LLMs enable them to pull and generate information from various sources quickly and reliably.

Facilitating conversations : Virtual assistants can engage in conversations, providing a more human-like interaction.

Enhancing accessibility : For individuals with disabilities or those needing hands-free support, virtual assistants offer a valuable tool for interacting with technology and accessing information effortlessly.

Real-world apps

Alexa :

Alexa is Amazone voice-controlled virtual assistant based on a cloud service. It is capable of voice interaction, music playback, setting alarms, streaming podcasts, and providing real-time information such as news or weather. Alexa can also control several smart devices, functioning as a home automation system.

5. Code development

Large language models can assist programmers in writing, reviewing, and debugging code. These models can understand and generate code snippets, suggest completions, and even write entire functions based on brief descriptions. For instance, a developer might input a comment like "sort a list of numbers in ascending order," and the LLM can provide the corresponding code.

Furthermore, LLMs can translate code between different programming languages, making it easier for developers to work with unfamiliar syntax or migrate projects to a new language.

Real-world apps

StarCoder is among the most popular LLM applications designed to make developers' lives a little bit easier. Let's dive deeper into it.

StarCoder

StarCoder is a collaborative effort between Hugging Face and ServiceNow. This open-source LLM is trained on a diverse and extensive dataset sourced from GitHub, which includes a wide array of programming languages, Git commits, GitHub issues, and Jupyter notebooks. The model itself is substantial, with approximately 15 billion parameters and training on 1 trillion tokens, with further fine-tuning on 35 billion Python tokens.

In terms of performance, StarCoder has been demonstrating strong capabilities in various coding tasks. It can handle a large context length of over 8,000 tokens, which is particularly useful for understanding and generating extensive code sequences. This makes it suitable for code autocompletion, modification, and providing explanations in natural language.

One of the notable features of StarCoder is its multilingual support, allowing it to understand and generate code in over 80 languages.

6. Sentiment analysis

Large language model applications, often enhanced through cross-platform app development or mobile app development services, can be utilized for sentiment analysis, thanks to their deep understanding of language nuances and context. Trained on extensive datasets, they can quite accurately determine the sentiment behind texts, ranging from social media posts to customer reviews.

LLM applications work by classifying the text into categories such as positive, negative, or neutral, often accompanied by associated confidence scores. For instance, in customer feedback analysis, large language models can discern specific emotions or attitudes towards products or services. This enables businesses to gain valuable insights into customer satisfaction and tailor their strategies accordingly.

Real-world apps

Grammarly :

Grammarly is a widely used writing enhancement tool and a browser extension. It provides grammar and spell checking as well as plagiarism detection services, among others, to ensure that the text is clear and mistake-free. At its core, Grammarly uses powerful large language models to understand the context and offer suggestions to enhance writing style, tone, and clarity.

7. Question answering

Question answering is among typical and very widespread LLM applications. These models easily understand and generate human-like text, making them ideal for providing accurate and contextually relevant answers to a wide array of questions.

Users can interact with large language models through search engines, virtual assistants, customer service bots, or educational platforms.

Real-world apps

Let’s explore an example of how LLMs are utilized for question answering from Meta.

LLaMA

LLaMA, short for Large Language Model Meta AI, is trained on a vast corpus of 1.4 trillion tokens, enabling it to predict and generate text by taking a sequence of words as input. This is why LLaMA is particularly good at answering questions across various domains, understanding context, and providing accurate, relevant information.

Its versatility allows it to be fine-tuned for specific tasks, especially sophisticated question answering scenarios. By offering LLaMA in various sizes and sharing its code, Meta aims to make AI research more accessible and encourage further work on enhancing large language models for better problem-solving and question-answering.

8. Market research

Large language models are able to provide deep insights into consumer behavior, trends, and preferences. They can analyze customer feedback, identify patterns and sentiments, predict market trends, and generate reports summarizing complex data into actionable insights.

For instance, an LLM can evaluate thousands of product reviews to determine the most appreciated features or common complaints, guiding companies in product development and marketing strategies.