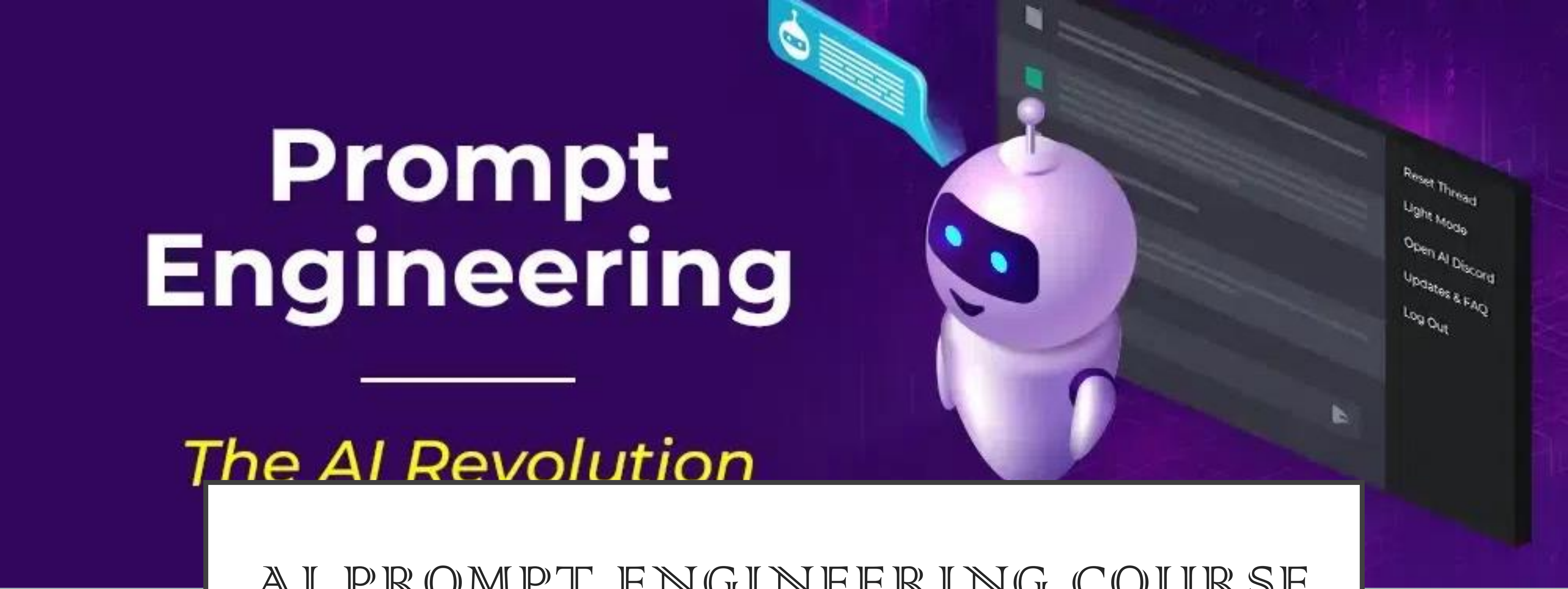


AI PROMPT ENGINEERING COURSE

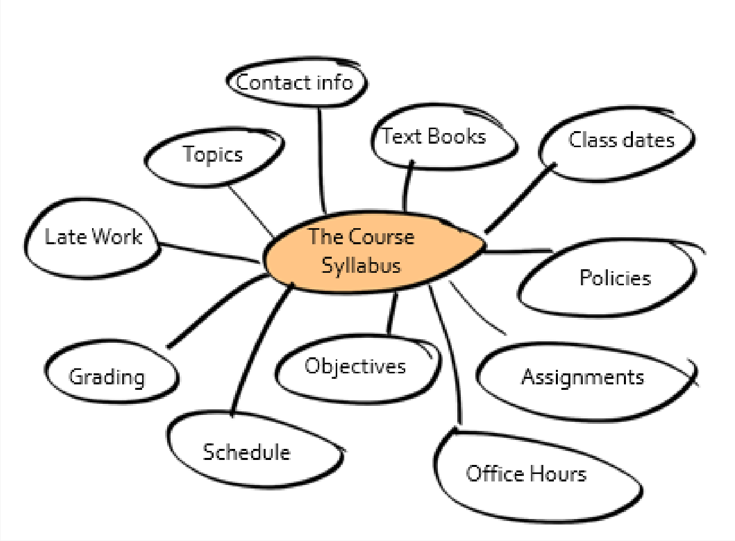
3d Grade IT Students

Lecturer: Mohamamd Salim Al-Othman

Tishk International University

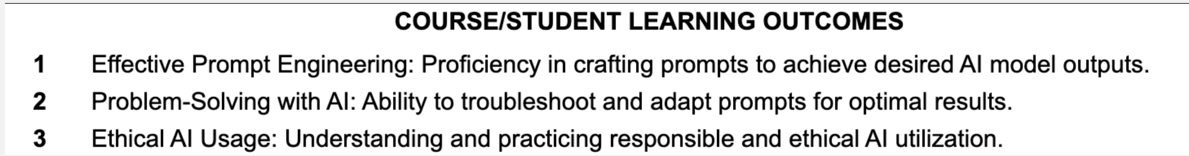
COURSE SYLLABUS

- Week 1
  - Theory: Introduction to prompt engineering, what are LLMs?, how do LLMs work?, the different types of prompts
  - Practical: Hands-on experience with LLMs, practicing writing prompts for different tasks
- Week 2
  - Theory: How to write effective prompts, best practices for prompt engineering
  - Practical: Evaluating the results of prompts, troubleshooting common problems with prompt engineering
- Week 3
  - Theory: Case studies of successful prompt engineering applications
  - Practical: Building a simple prompt engineering application
- Week 4
  - Theory: AI prompt engineering for code generation
  - Practical: Practicing writing prompts for code generation, evaluating the results of prompts, troubleshooting common problems
- Week 5
  - Theory: AI prompt engineering for creative writing
  - Practical: Practicing writing prompts for creative writing, evaluating the results of prompts, troubleshooting common problems
- Week 6
  - Theory: AI prompt engineering for question answering
  - Practical: Practicing writing prompts for question answering, evaluating the results of prompts, troubleshooting common problems



COURSE SYLLABUS – CONT.

- Week 7
  - Theory: Other case studies of prompt engineering applications
  - Practical: Building a prompt engineering application for a specific task
- Week 8
  - Theory: Advanced prompt engineering techniques (e.g., prompt chaining, prompt tuning, prompt ensembles, few-shot learning with prompts)
  - Practical: Practicing advanced prompt engineering techniques
- Week 9
  - Theory: Advanced prompt engineering techniques (e.g., prompt chaining, prompt tuning, prompt ensembles, few-shot learning with prompts)
  - Practical: Practicing advanced prompt engineering techniques
- Week 10
  - Theory: Advanced prompt engineering techniques (e.g., prompt chaining, prompt tuning, prompt ensembles, few-shot learning with prompts)
  - Practical: Practicing advanced prompt engineering techniques
- Week 11
  - Theory: Advanced prompt engineering techniques (e.g., prompt chaining, prompt tuning, prompt ensembles, few-shot learning with prompts)
  - Practical: Working on the final project
- Week 12
  - Theory: Review of key concepts
  - Practical: Final project presentations

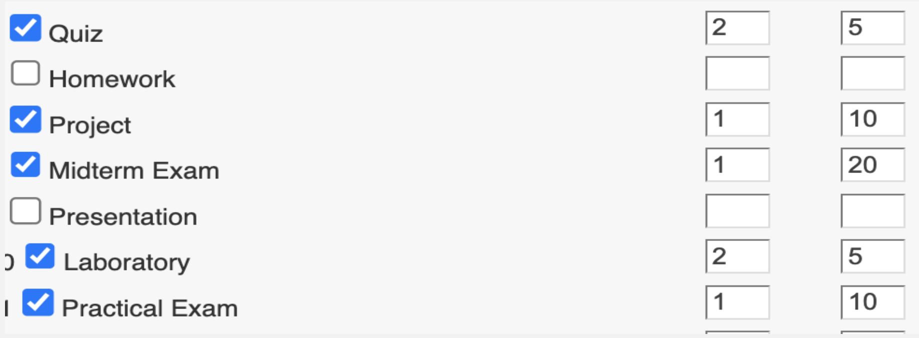


COURSE/STUDENT LEARNING OUTCOMES

1 Effective Prompt Engineering: Proficiency in crafting prompts to achieve desired AI model outputs.

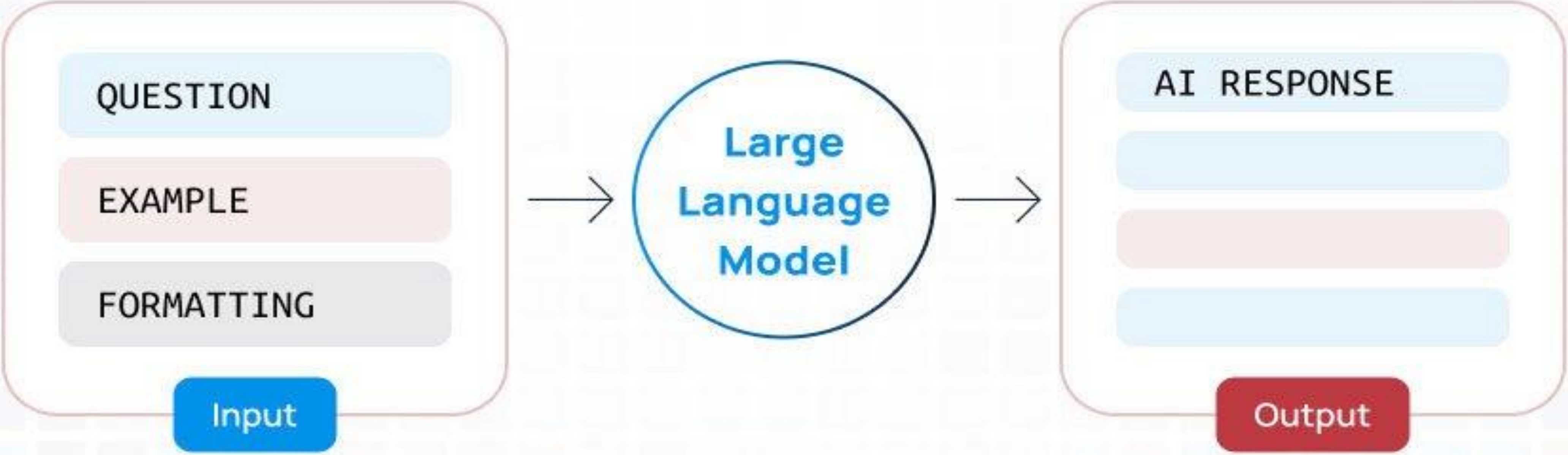
2 Problem-Solving with AI: Ability to troubleshoot and adapt prompts for optimal results.

3 Ethical AI Usage: Understanding and practicing responsible and ethical AI utilization.



☑ Quiz	2	5
□ Homework		
☑ Project	1	10
☑ Midterm Exam	1	20
□ Presentation		
☑ Laboratory	2	5
☑ Practical Exam	1	10

What is Prompt Engineering?



QUESTION

EXAMPLE

FORMATTING

Input

Large Language Model

AI RESPONSE

Output

INTERACTIVE DISCUSSION ON REAL-WORLD APPLICATIONS OF LLMS

Intro:

- Can you think of any app or tool that responds to your questions in natural language?
- How might these tools understand what you're asking?
- **Examples**:
- **ChatGPT** or **Google Assistant** for conversational AI.
- **Grammarly** or **DeepL Translator** for language editing and translation.
- **Code Assistants** like GitHub Copilot for generating code.

INTRODUCTION TO PROMPT ENGINEERING

What is prompt engineering?

- It is the process of designing prompts for AI models to guide them towards desired outputs. Prompts can be used to control the style, tone, and content of the output, as well as to provide additional information or context. Prompt engineering is an important skill for anyone who wants to use AI models to **generate creative content, translate languages, or answer questions accurately and informatively.**

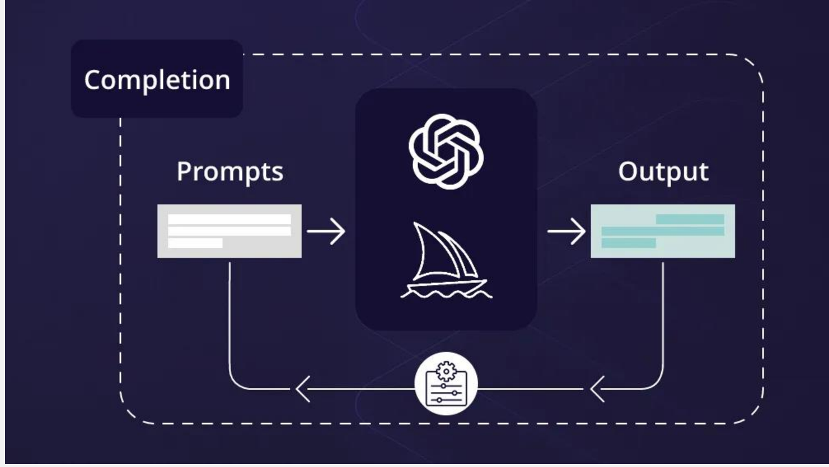
Why is prompt engineering important?

- It is important because it allows us to control the output of AI models. By carefully crafting prompts, we can ensure that AI models generate the desired results. This is especially important for tasks such as generating creative content, translating languages, and answering questions accurately and informatively.

The benefits of using prompt engineering

- **Improved accuracy and informativeness:** Prompt engineering can help to improve the accuracy and informativeness of AI outputs. By providing clear and concise instructions, we can help AI models to better understand what we are asking them to do.
- **Greater control**: Prompt engineering gives us greater control over the output of AI models. By carefully crafting prompts, we can control the style, tone, and content of the output.
- **Enhanced creativity:** Prompt engineering can be used to enhance the creativity of AI outputs. By providing AI models with additional information or context, we can help them to generate more creative and innovative responses.

EXAMPLES OF PROMPT ENGINEERING APPLICATIONS

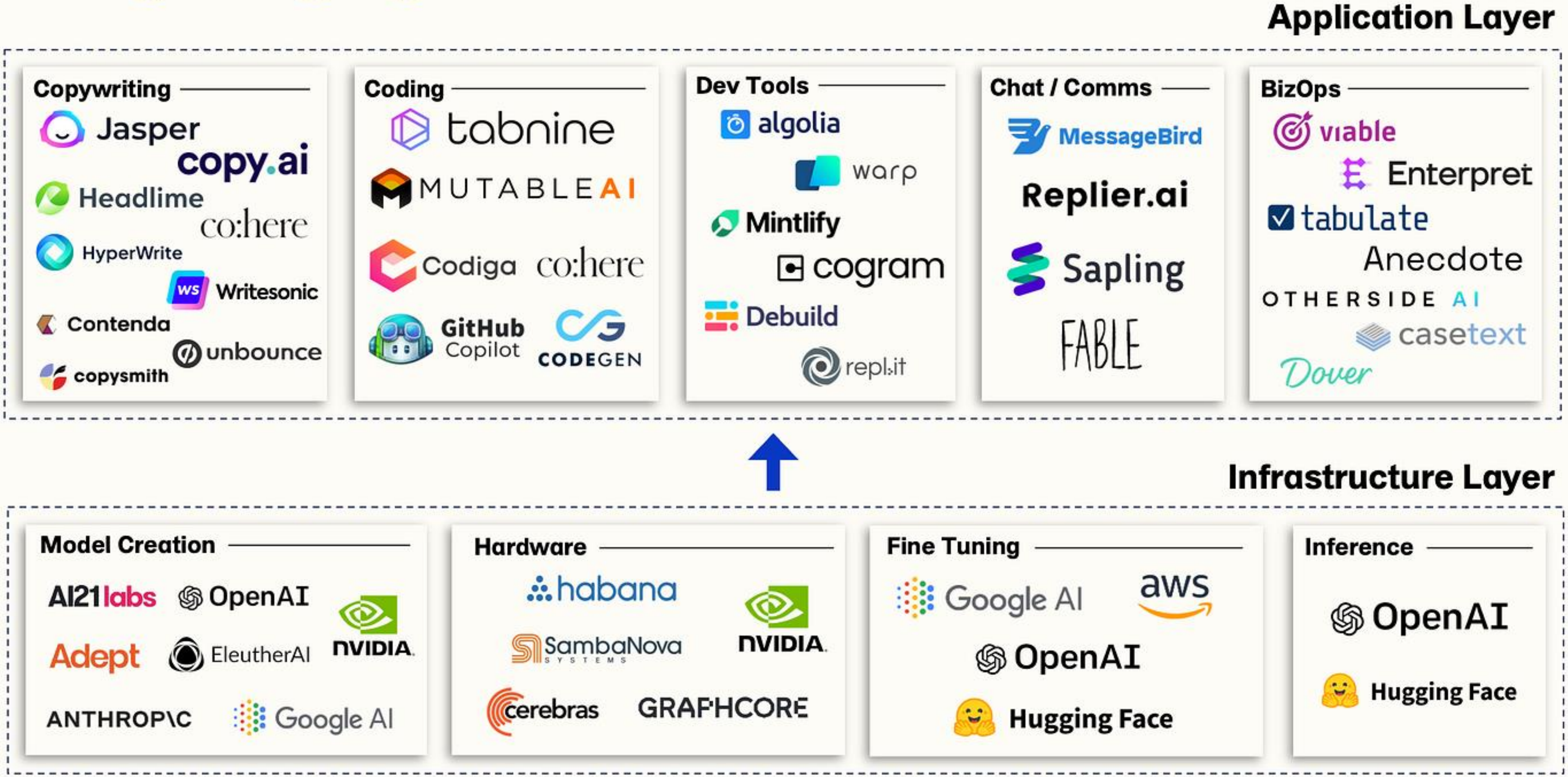


Prompt engineering can be used for a variety of applications, including:

- **Creative writing:** Prompt engineering can be used to generate creative content, such as poems, stories, and scripts.
- **Translation:** Prompt engineering can be used to improve the quality of machine translation.
- **Question answering:** Prompt engineering can be used to improve the accuracy and informativeness of question answering systems.
- **Code generation:** Prompt engineering can be used to generate code for different programming languages.
- **Debugging:** Prompt engineering can be used to help debug code.

Large Language Models

BCV



Application Layer

Copywriting
- Jasper
- copy.ai
- Headlime
- co:here
- HyperWrite
- Writesonic
- Contenda
- unbounce
- copysmith

Coding
- tabnine
- MUTABLEAI
- Codiga
- co:here
- GitHub Copilot
- CODEGEN

Dev Tools
- algolia
- warp
- Mintlify
- cogram
- Debuild
- repl.it

Chat / Comms
- MessageBird
- Replier.ai
- Sapling
- FABLE

BizOps
- viable
- Enterpret
- tabulate
- Anecdote
- OTHERSIDE AI
- casetext
- Dover

Infrastructure Layer

Model Creation
- AI21labs
- OpenAI
- NVIDIA
- Adept
- EleutherAI
- ANTHROP\C
- Google AI

Hardware
- habana
- NVIDIA
- SambaNova SYSTEMS
- cerebras
- GRAPHCORE

Fine Tuning
- Google AI
- aws
- OpenAI
- Hugging Face

Inference
- OpenAI
- Hugging Face

WHAT ARE LARGE LANGUAGE MODELS (LLMS)?

What are **LLMs**?

- Large language models (LLMs) are a subset of artificial intelligence (AI) designed to understand and generate human language. They are trained on vast amounts of text data, enabling them to learn patterns, context, and nuances in language.

Large Language Models (LLMs): A specific type of generative AI focused on understanding and generating human language in text format.

- **Characteristics**:
  - Trained on vast textual data
  - Performs tasks like translation, summarization, and question answering
- **Examples**:
  - OpenAI's GPT series, Google's BERT and Meta's LLaMA

Generative AI: A broad category of AI that creates new content, including text, images, music, and code.

- **Applications**:
  - Image generation (e.g., DALL-E), and Music composition (e.g., MuseNet)
  - Video creation, and Code generation (e.g., GitHub Copilot)
  - Text generation (e.g., story writing)

Summary: All LLMs are generative AI, but not all generative AI systems are LLMs.

HOW DO LLMS WORK?

- **LLMs** work by using a technique called **deep learning**. Deep learning is a type of **machine learning** that uses **artificial neural networks** to learn from **data**. Artificial neural networks are inspired by the structure and function of the human brain.
- **LLMs** are trained on a massive amount of text data. This data can include **books, articles, code, and other forms of text**. As the LLM is trained, it learns the patterns in the data and how to generate text that is similar to the data it was trained on.

NEURAL NETWROK VS DEEP LEARNING

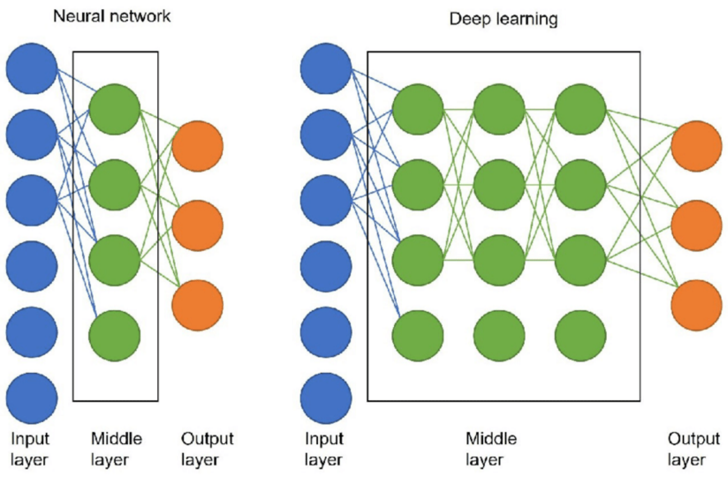
Neural Network:

- Think of a neural network as a basic building block.
- It's inspired by the structure of the human brain and consists of layers of interconnected nodes, similar to neurons.
- These networks are used to solve various machine learning tasks, like image recognition or language processing.
- They can be shallow (few layers) or deep (many layers).

Deep Learning:

- Deep learning is like a more advanced version of neural networks.
- It refers specifically to neural networks with **many layers**, often called deep neural networks.
- Deep learning has gained popularity because it can **automatically learn** and extract **complex patterns and features from data.**
- It's used in a wide range of applications, including **speech recognition, autonomous vehicles, and recommendation systems.**

In essence, neural networks are the basic components, while deep learning is the broader field that focuses on using very deep neural networks to solve complex problems. So, deep learning is like taking neural networks to the next level by making them deeper and more capable of handling intricate tasks.

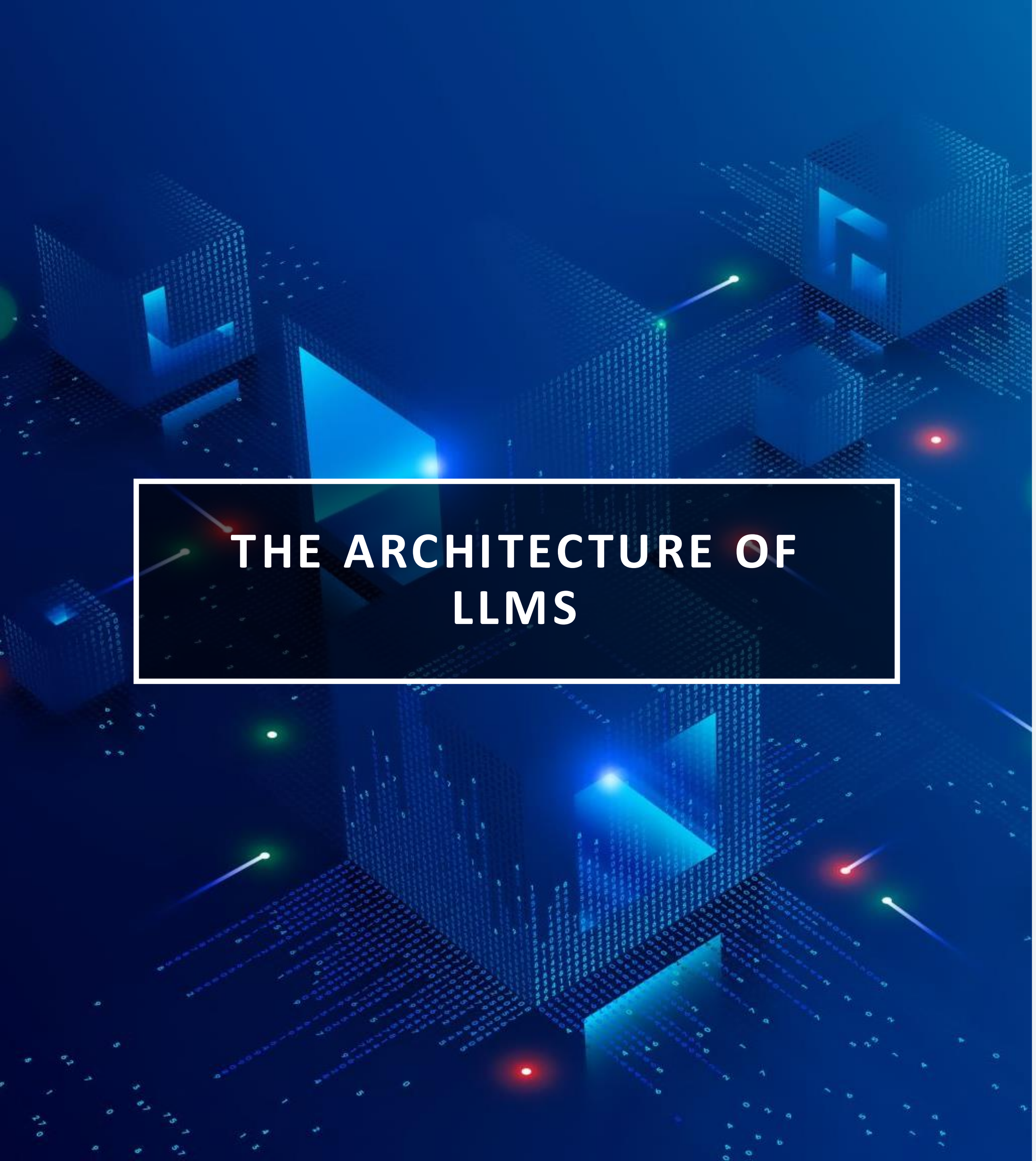


THE DIFFERENT TYPES OF LLMS

There are many different types of **LLMs**. Some of the most popular **LLMs** include:

- GPT-3
- LaMDA
- Megatron-Turing NLG
- Jurassic-1 Jumbo
- Wu Dao 2.0

These LLMs differ in their size, architecture, and training data. However, they all work on the same basic principle of **using deep learning to generate text**.



THE ARCHITECTURE OF LLMS

The architecture of LLMs

LLMs are typically built using a type of neural network called a **transformer**.

- **Transformers** are a state-of-the-art neural network architecture that is well-suited for natural language processing tasks.
- **Transformers** are made up of two main components: **encoders** and **decoders**. Encoders encode the input text into a sequence of hidden states. Decoders then decode the hidden states into the **output text.**

The training process for LLMs

- **LLMs** are trained using a technique called **supervised learning.** Supervised learning is a type of machine learning that uses labeled data to train a model.
- To train an **LLM**, researchers first collect a massive dataset of text data. This data is then labeled with the desired output

THE DIFFERENT TYPES OF PROMPTS

Prompts are instructions that are provided to a large language model (LLM) to guide its output. There are many different types of prompts, each with its own strengths and weaknesses.

Instruction prompts

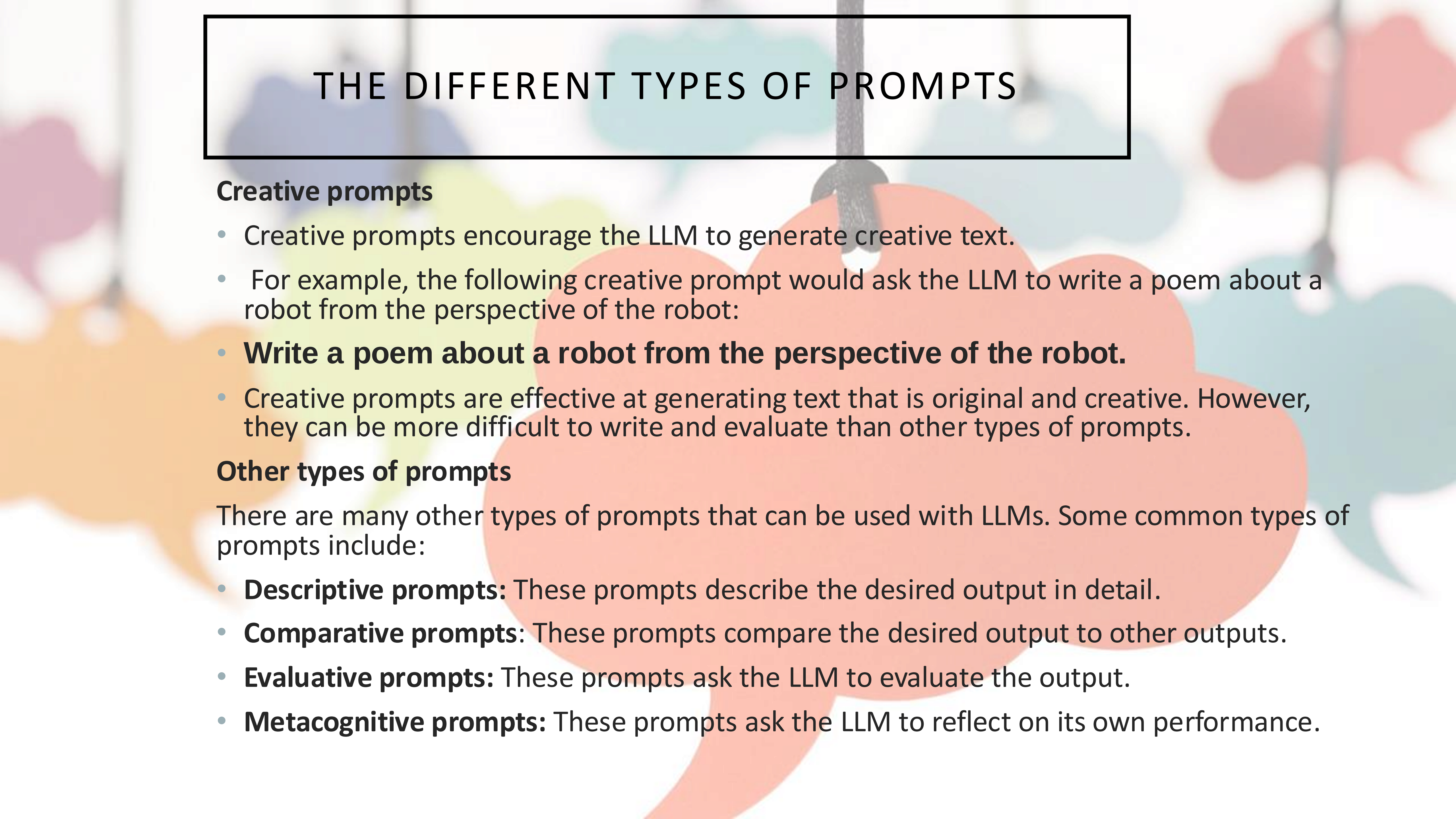
- Instruction prompts are the simplest type of prompt. They provide the LLM with clear and concise instructions on what to do. For example, the following instruction prompt would ask the LLM to generate a poem about a robot:
- Example: **Write a poem about a robot.**
- Instruction prompts are effective at generating text that is consistent with the prompt. However, they can be limiting in terms of the creativity and originality of the output.

Example prompts

- Example prompts provide the LLM with examples of the desired output. For example, the following example prompt would ask the LLM to generate a poem about a robot, similar to the poem "Ode to a Nightingale" by John Keats:
- Example: **Write a poem about a robot, similar to the poem "Ode to a Nightingale" by John Keats**.
- Example prompts are effective at generating text that is similar to the example. However, they can be limiting in terms of the creativity and originality of the output.

Query prompts

- Query prompts ask the LLM a question. The LLM will then generate text that answers the question. For example, the following query prompt would ask the LLM to generate a poem about a robot:
- Example: What is a poem about a robot?
- Query prompts are effective at generating text that is informative and comprehensive. However, they can be limiting in terms of the creativity and originality of the output.



THE DIFFERENT TYPES OF PROMPTS

Creative prompts

- Creative prompts encourage the LLM to generate creative text.
- For example, the following creative prompt would ask the LLM to write a poem about a robot from the perspective of the robot:
- **Write a poem about a robot from the perspective of the robot.**
- Creative prompts are effective at generating text that is original and creative. However, they can be more difficult to write and evaluate than other types of prompts.

Other types of prompts

There are many other types of prompts that can be used with LLMs. Some common types of prompts include:

- **Descriptive prompts:** These prompts describe the desired output in detail.
- **Comparative prompts**: These prompts compare the desired output to other outputs.
- **Evaluative prompts:** These prompts ask the LLM to evaluate the output.
- **Metacognitive prompts:** These prompts ask the LLM to reflect on its own performance.

SIMPLE PRACTICE PROMPTS ACTIVITY

- **Activity**:
  - Present a basic prompt like:“**Write a poem about a robot.**”
  - Modify the prompt progressively to show how subtle changes yield different results. For instance:
    - “**Write a happy poem about a robot exploring spa**ce.”
    - “**Write a sad poem about a robot missing its creator**.”

Try out variations on your devices, experimenting with prompts such as:

- “Summarize this text in one sentence.”
- “Rewrite this email in a more professional tone.”

HOW TO WRITE EFFECTIVE PROMPTS

When writing prompts for LLMs, it is important to be clear, concise, and informative. You should also provide the LLM with enough information to generate the desired output.

Here are some tips for writing effective prompts:

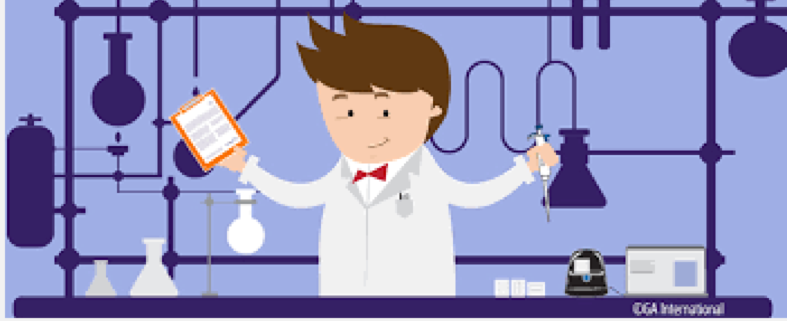
- **Be clear and concise:** Your prompt should be clear and easy for the LLM to understand. Avoid using ambiguous language or complex sentence structures.
- **Provide enough information:** Your prompt should provide the LLM with enough information to generate the desired output. For example, if you are asking the LLM to generate a poem about a robot, you should specify the tone, style, and length of the poem.
- **Use simple language:** Avoid using complex language or jargon in your prompts. The LLM is more likely to generate accurate and informative outputs if you use simple language.
- **Avoid ambiguity**: Your prompt should be unambiguous. Avoid using language that could be interpreted in multiple ways.
- **Experiment with different prompts**: There is no one-size-fits-all approach to writing prompts. Experiment with different prompts to see what works best for you and the specific task you are trying to accomplish

BEST PRACTICES FOR PROMPT ENGINEERING

Here are some best practices for prompt engineering:

- **Use a variety of prompts:** Don't rely on a single prompt to generate the desired output. Experiment with different prompts to see what works best.
- **Evaluate the results of prompts**: Evaluate the results of your prompts to ensure that they are generating the desired output. If you are not satisfied with the results, try modifying the prompts.
- **Troubleshoot common problems**: There are a number of common problems that can occur with prompt engineering. For example, the **LLM** may generate text that is inaccurate, incomplete, or uninformative. If you encounter any problems, try troubleshooting by modifying the prompts or using a different type of prompt.
- **Keep up with the latest research**: The field of prompt engineering is constantly evolving. Keep up with the latest research to learn about new techniques and best practices.

LAB WORK (WEEK1)



These are some ideas for practical activities for the first week of an AI prompt engineering course, without using any programming language:

- Step 1: Create an account with a cloud-based LLM provider (e.g., OpenAI, Google AI Platform).
- Step 2: Set the prompt for the LLM. For example, you could prompt the LLM to write a poem about a black hole.
- Step 3: Generate text with the LLM.
- Step 4: Evaluate the results of the prompt by checking for accuracy, completeness, and fluency.
- If you are not satisfied with the results, you can try modifying the prompt and generating text again. You can also experiment with different prompts to generate different types of creative content, such as stories, scripts, and musical pieces.
- Here are some other interesting prompts that students could try:
- Write a poem about a future where humans and AI coexist peacefully.
- Write a story about a robot who falls in love with a human.
- Write a script for a movie about a team of scientists who travel through time to prevent a global catastrophe.
- Write a musical piece about the beauty of the universe.

REFERENCES

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. ... Ramesh, A. Kaplan, M. Dhariwal, P. Sadhankar, and I. Sutskever, “Language models are few-shot learners,” arXiv preprint arXiv:2005.14165, 2020.

[2] F. Gao, S. Yao, Y. Wu, M. Zhou, and S. Huang, “Prompt tuning: Efficient and effective fine-tuning of large language models,” arXiv preprint arXiv:2201.10360, 2022.

[3] T. Gao, S. Yao, Y. Wu, M. Zhou, and S. Huang, “The power of scale: Scaling student models for robust and scalable prompt engineering,” arXiv preprint arXiv:2204.07066, 2022.

[4] P. J. Liu, Y. Zhang, X. Sun, H. Zhang, and X. Zhou, “Prompt engineering for creative text generation,” arXiv preprint arXiv:2201.10553, 2022.

[5] X. Chen, C. Sutton, and M. Allaman, “Prompt engineering for code generation,” arXiv preprint arXiv:2103.14656, 2021.

[6] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Vinyals, and V. Mnih, “Prompt engineering for machine translation,” arXiv preprint arXiv:2201.08237, 2022.

[7] Y. Zhang, X. Gao, and M. Wang, “Prompt engineering for question answering,” arXiv preprint arXiv:2105.13283, 2021.