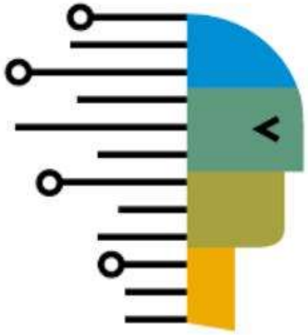
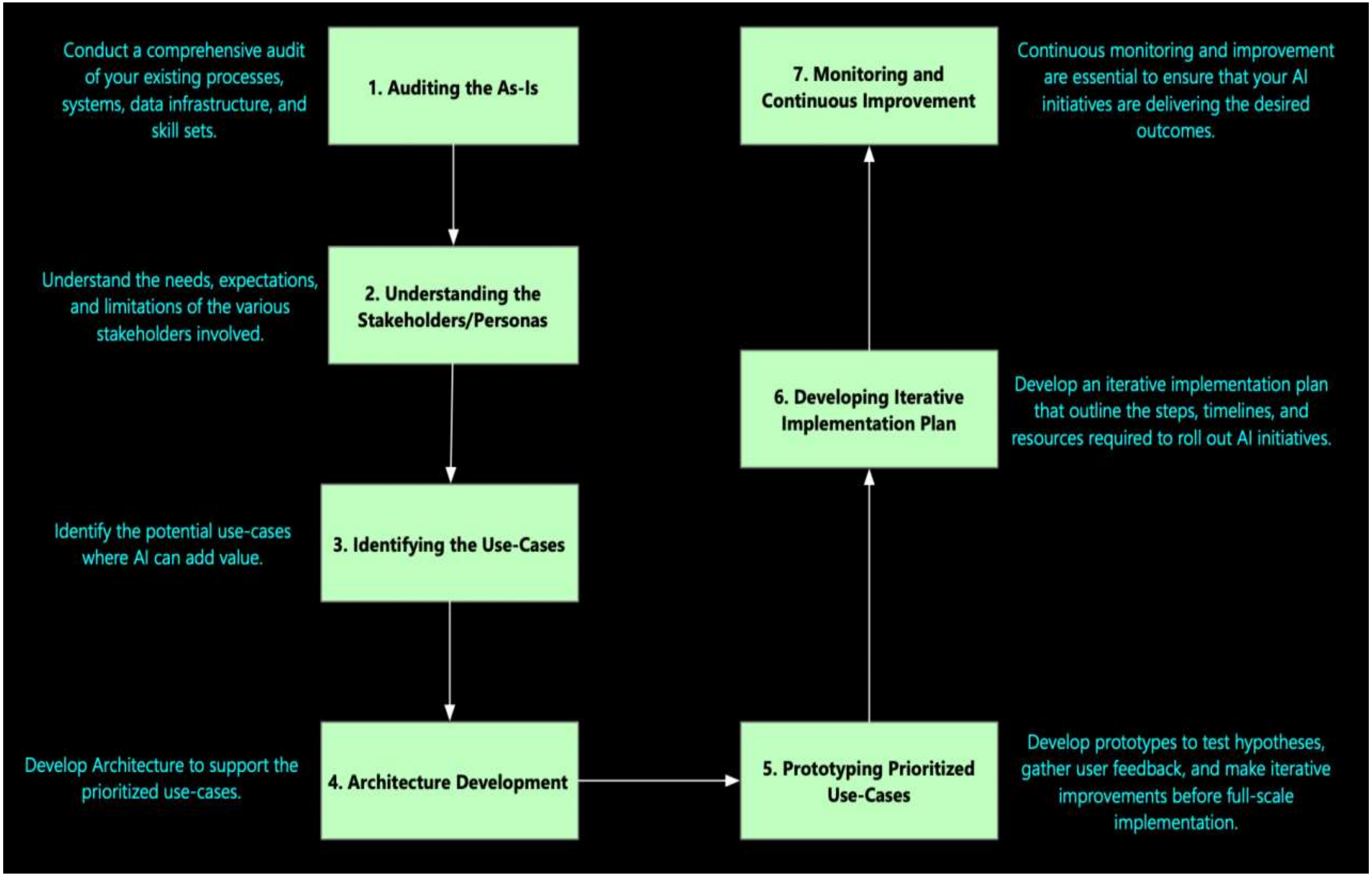
INT426
Generative Artificial Intelligent

Lecture 2



Integrating AI into various facets of a business requires a meticulous approach. We will dissect these seven steps using a structured process called the Input-Transformation-Output (ITO) model.



Conduct a comprehensive audit of your existing processes, systems, data infrastructure, and skill sets.

1. Auditing the As-Is

Understand the needs, expectations, and limitations of the various stakeholders involved.

2. Understanding the Stakeholders/Personas

Identify the potential use-cases where AI can add value.

3. Identifying the Use-Cases

Develop Architecture to support the prioritized use-cases.

4. Architecture Development

5. Prototyping Prioritized Use-Cases

Develop prototypes to test hypotheses, gather user feedback, and make iterative improvements before full-scale implementation.

6. Developing Iterative Implementation Plan

Develop an iterative implementation plan that outline the steps, timelines, and resources required to roll out AI initiatives.

7. Monitoring and Continuous Improvement

Continuous monitoring and improvement are essential to ensure that your AI initiatives are delivering the desired outcomes.

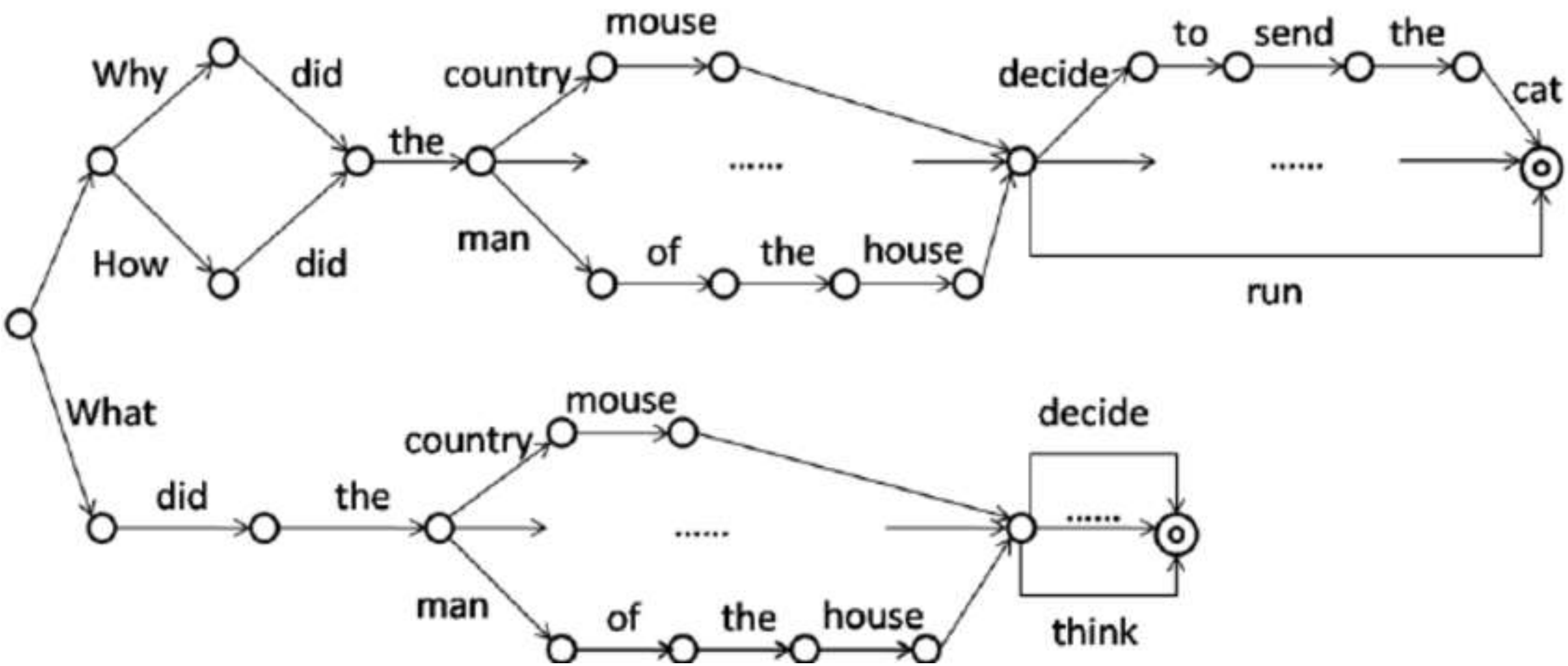
What is a language model?

A language model is a machine learning model that aims to predict and generate plausible language. Autocomplete is a language model, for example.
These models work by estimating the probability of a token or sequence of tokens occurring within a longer sequence of tokens. Consider the following sentence:

When I hear rain on my roof, I _____ in my kitchen.

A "sequence of tokens" could be an entire sentence or a series of sentences. That is, a language model could calculate the likelihood of different entire sentences or blocks of text.
Estimating the probability of what comes next in a sequence is useful for all kinds of things: generating text, translating languages, and answering questions, to name a few.

EXAMPLE OF LANGUAGE MODEL



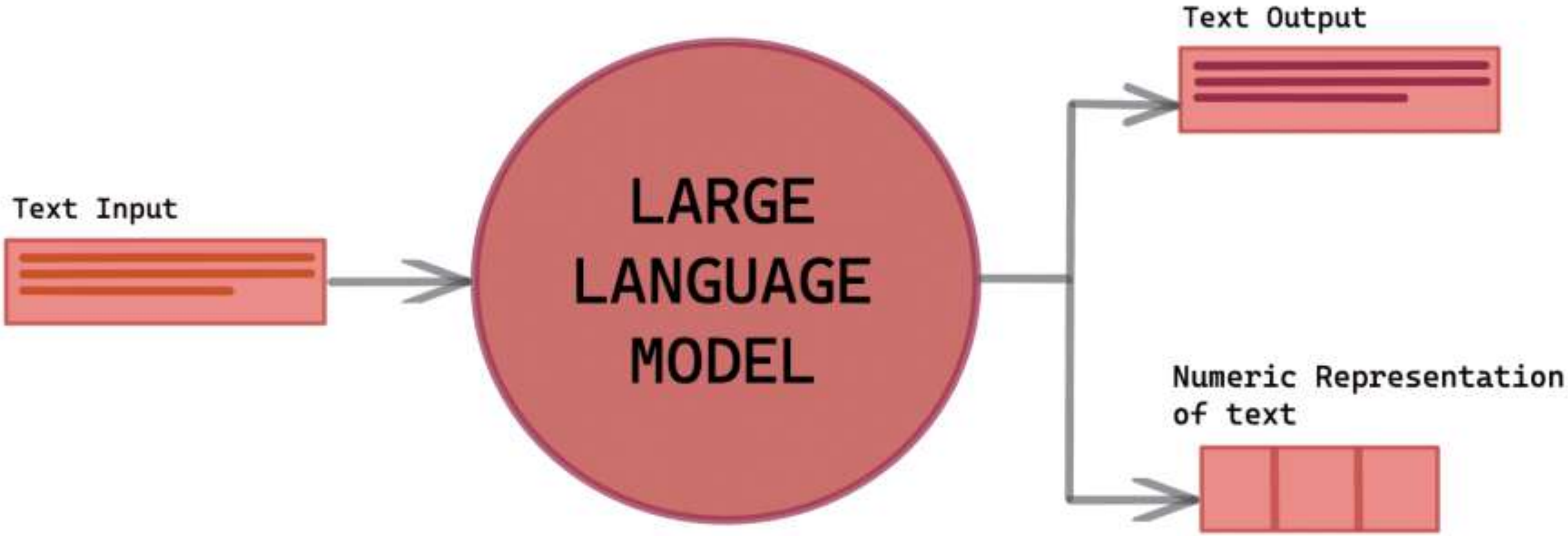
What is a large language model?

Modeling human language at scale is a highly complex and resource-intensive endeavor. The path to reaching the current capabilities of language models and large language models has spanned several decades.

i. As models are built bigger and bigger, their complexity and efficacy increases. Early language models could predict the probability of a single word; modern large language models can predict the probability of sentences, paragraphs, or even entire documents.

i. The size and capability of language models has exploded over the last few years as computer memory, dataset size, and processing power increases, and more effective techniques for modeling longer text sequences are developed.

SIMPLE STRUCTURE OF LLM



How large is large?

The definition is fuzzy, but "large" has been used to describe BERT (110M parameters) as well as PaLM 2 (up to 340B parameters).
Parameters are the **weights** the model learned during training, used to predict the next token in the sequence. "Large" can refer either to the number of parameters in the model, or sometimes the number of words in the dataset.

Transformers

Transformers are the state-of-the-art architecture for a wide variety of language model applications, such as translators.
If the input is **"I am a good dog."**, a Transformer-based translator transforms that input into the output **"Je suis un bon chien."**, which is the same sentence translated into French.
Full Transformers consist of an encoder and a **decoder.** An encoder converts input text into an intermediate representation, and a decoder converts that intermediate representation into useful text.

Self-attention

Transformers rely heavily on a concept called self-attention. The self part of self-attention refers to the "egocentric" focus of each token in a corpus. Effectively, on behalf of each token of input, self-attention asks, **"How much does every other token of input matter to me?**" To simplify matters, let's assume that each token is a word and the complete context is a single sentence. Consider the following sentence:

“The animal didn't cross the street because it was too tired.”

There are 11 words in the preceding sentence, so each of the 11 words is paying attention to the other ten, wondering how much each of those ten words matters to them. For example, notice that the sentence contains the pronoun **it**. Pronouns are often ambiguous. The pronoun **it** always refers to a recent noun, but in the example sentence, which recent noun does **it** refer to: the animal or the street?

The self-attention mechanism determines the relevance of each nearby word to the pronoun **it**.

What are some use cases for LLMs?

LLMs are highly effective at the task they were built for, which is generating the most plausible text in response to an input. They are even beginning to show strong performance on other tasks; for example, summarization, question answering, and text classification.

These are called emergent abilities. LLMs can even solve some math problems and write code (though it's advisable to check their work).
LLMs are excellent at mimicking human speech patterns. Among other things, they're great at combining information with different styles and tones.

However, LLMs can be components of models that do more than just generate text. Recent LLMs have been used to build sentiment detectors, toxicity classifiers, and generate image captions.

LLM Considerations

Models this large are not without their drawbacks.

The largest LLMs are expensive. They can take months to train, and as a result consume lots of resources.

They can also usually be repurposed for other tasks, a valuable silver lining.

i. Training models with upwards of a trillion parameters creates engineering challenges. Special infrastructure and programming techniques are required to coordinate the flow to the chips and back again.

ii. There are ways to mitigate the costs of these large models. Two approaches are offline inference and distillation.

Bias can be a problem in very large models and should be considered in training and deployment.

iii. As these models are trained on human language, this can introduce numerous potential ethical issues, including the misuse of language, and bias in race, gender, religion, and more.

iv. It should be clear that as these models continue to get bigger and perform better, there is continuing need to be diligent about understanding and mitigating their drawbacks. Learn more about Google's approach to responsible AI.

WHAT IS PROMPT ENGINEERING?

A *prompt* is the input you provide, typically text, when interfacing with an AI model like ChatGPT or Midjourney. Here is a simple example of a prompt input, and the resulting output.

Input:
Can I have a list of product names for a pair of shoes that can fit any foot size?

Output:
Certainly! Here's a list of product names for a pair of shoes that can fit any foot size: UniFit SoleStrides FlexiSize All-Fit Shoes OmniFit FootFlex ... AnyStep AdaptiWalkers FlexiRange UniversalSteps PerfectGlide MultiFit Shoes

The Five Principles of Prompting are as follows:

1. Give Direction: Describe what you’re imagining, to get an output matching your vision.

2. Specify Format: Define the response you want, and minimize time spent parsing errors.

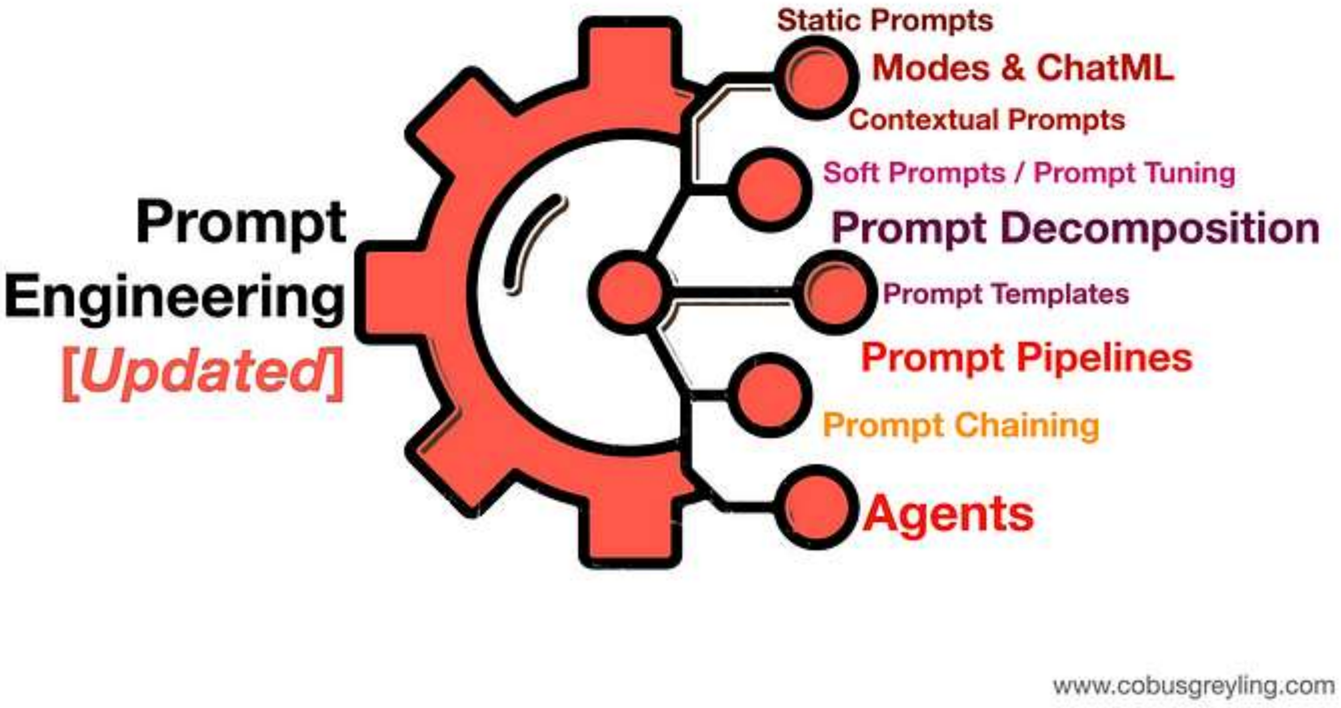
3. Provide Examples: Insert examples in your prompts, to improve the reliability of the output.

4. Evaluate Quality: Identify errors and rate responses, testing what drives performance.

5. Divide Labor: Split tasks into multiple prompts, chained together for complex goals.

What is Prompt Tuning?

Prompt tuning is the process of adapting LLMs to new tasks by training a small set of parameters known as prompts. These prompts are prepended to the input text to guide the LLM towards generating the desired output.



Benefits of Prompt Tuning

1. Efficient

Prompt tuning offers a more efficient approach compared to fine-tuning the entire LLM. It requires training only a small number of prompt parameters, resulting in faster adaptation to new tasks.

2. Flexible

Prompt tuning can be applied to a wide range of tasks in various domains, including natural language processing, image classification, and code generation. This flexibility makes prompt tuning a versatile technique for adapting LLMs.

3. Interpretable

With prompt tuning, it is possible to inspect the prompt parameters to understand how the LLM is being guided towards generating the desired output. This interpretability provides valuable insights into the model’s decision-making process.

Biggest Challenges of Prompt Tuning

1. Designing Complex Prompts

Designing effective prompts can be challenging, especially for complex tasks. It requires careful consideration of the language, structure, and context to ensure optimal performance.

2. Overfitting

Prompt tuning is prone to overfitting, particularly when the prompt is too large or too specific. Overfitting can lead to poor generalization on unseen data, limiting the performance of the adapted LLM.

3. Scalability

Scaling prompt tuning to tasks with large amounts of training data can be challenging. As the size of the training data increases, efficiently adapting the LLM requires careful optimization strategies.

THANK YOU