

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES

SAN JOSE, CA, USA
MARCH 4-7, 2024

Large Language Models to generate SystemC Model Code

Shravan Belagalmath, Sandeep Pendharkar,
Karthick Gururaj, Santhosh Selvin

vayavyalabs.com

Agenda

Part I LLM

- AI Overview
- Large Language Model
- Prompt Engineering

Part II

SystemC Code Generation

- Challenges
- Mitigations
- Demo

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Part I - LLM

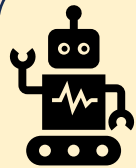
AI ML DL RL SL NLP LLM



AI Overview

Artificial Intelligence

Enabling machines to learn, think and act



Narrow Intelligence

Performs specific task efficiently

Example:

Virtual voice assistants, Recommender systems

General Intelligence



No difference between humans & machines

Example:

Sonny in I, Robot movie, Voice in Her movie

Human Capabilities



Vision



Natural Language



Movement



Vehicle Driving

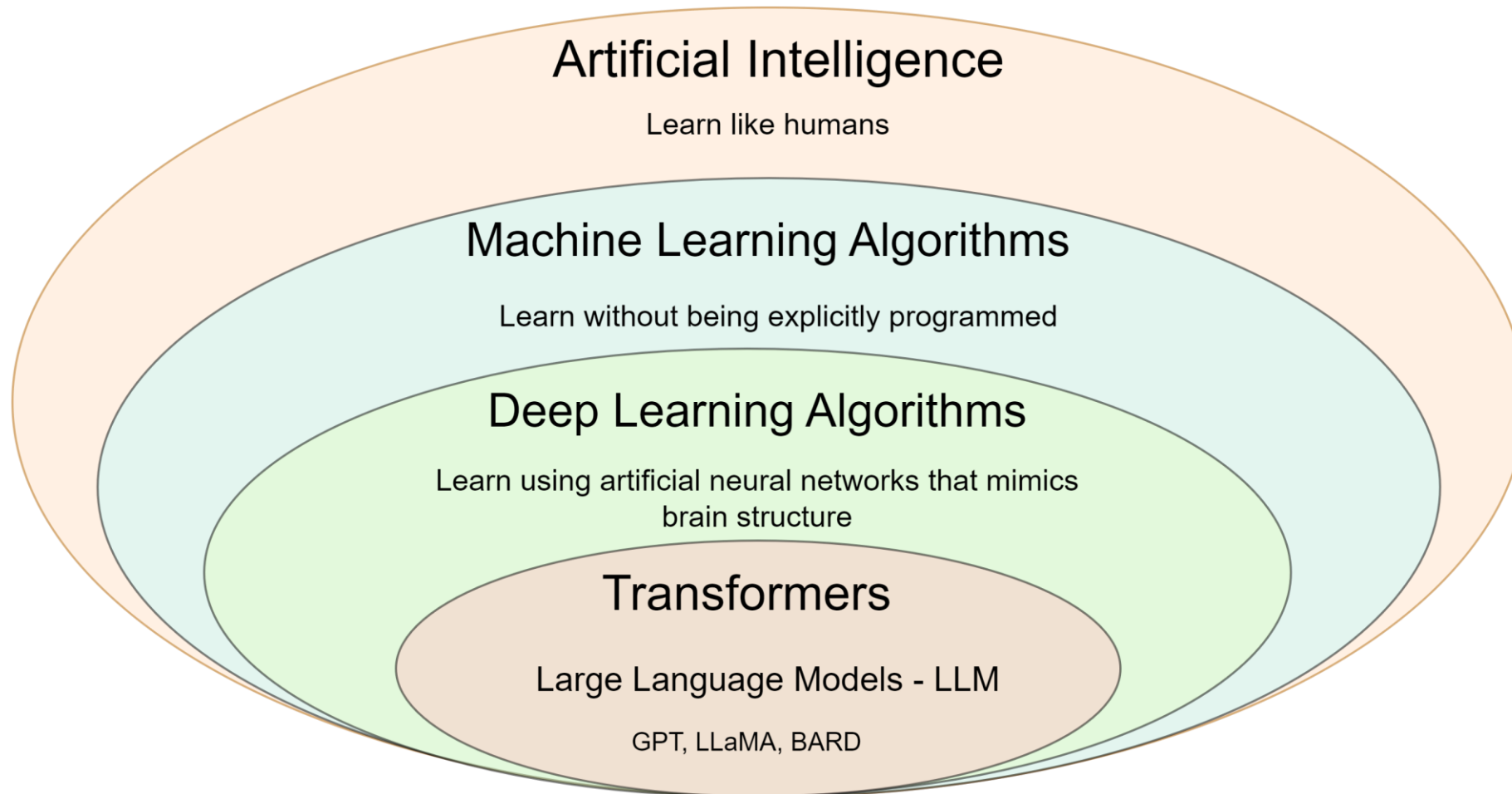


Creativity

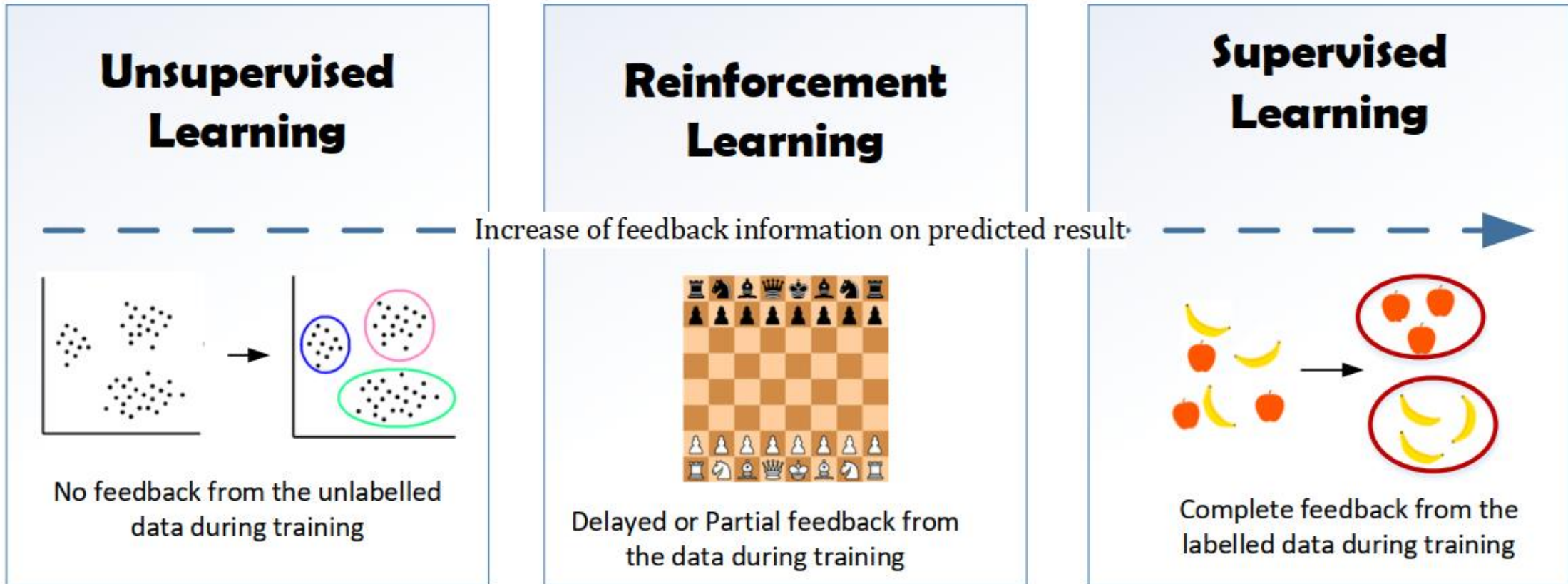


Hearing
Many more ...

LLM Overview



Learning from Data & Feedback



Source: Practical Mathematics for AI and Deep Learning Book by Tamoghna Ghosh and Shravan Belagalmath - <https://amzn.eu/d/8i6MPMY>

Chat with LLM

You
Explain about languages



Large
Language
Model



ChatGPT

Certainly! The term "languages" can refer to a variety of things depending on the context. Here are a few possible interpretations:

1. Human Languages:

- **Natural Languages:** These are languages spoken and written by humans, such as English, Spanish, Mandarin, French, etc. Each natural language has its own grammar, vocabulary, and syntax. There are thousands of languages spoken around the world, each with its own unique characteristics.
- **Constructed Languages (Conlangs):** Some languages are intentionally created, often for artistic, fictional, or experimental purposes. Examples include Klingon from Star Trek, Dothraki from Game of Thrones, and Esperanto, a constructed international auxiliary language.

2. Programming Languages:

- **High-Level Programming Languages:** These are languages designed for human readability and ease of use. Examples include Python, Java, C++, and JavaScript. Developers use these languages to write software, applications, and scripts.
- **Low-Level Programming Languages:** These languages are closer to machine code and provide more direct control over hardware. Examples include Assembly language. They are often used for system-level programming and optimization.

3. Animal Languages:

- Some animals communicate through various forms of language, though not as complex as human language. Examples include the waggle dance of bees, the songs of whales, and the vocalizations of primates.

4. Sign Languages:

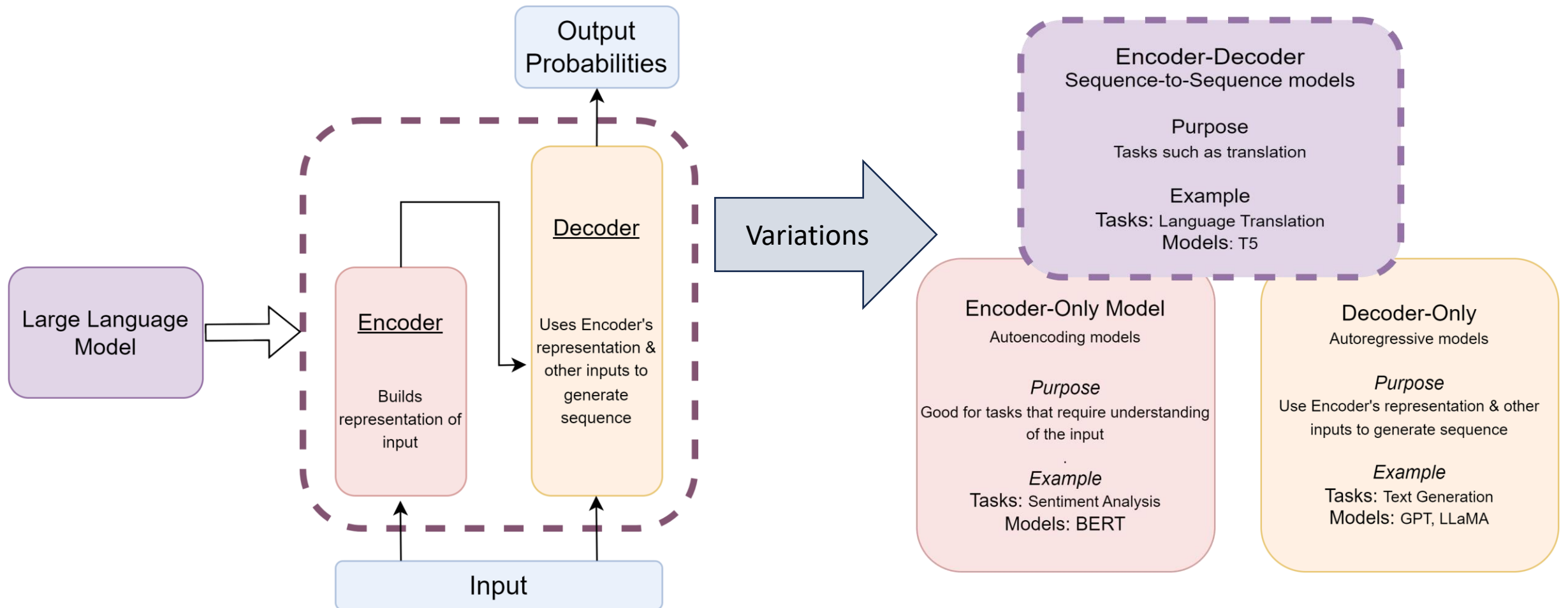
- These are languages that use visual-gestural communication, primarily through manual communication and body language. American Sign Language (ASL) is an example of a widely used sign language.

5. Computer Languages:

- **Markup Languages:** These languages are used to annotate text with information about the structure, formatting, or presentation. HTML (Hypertext Markup Language) is an example used in web development.
- **Query Languages:** These are languages used to interact with databases. SQL (Structured Query Language) is a common example.
- **Scripting Languages:** These are languages often interpreted and used for automating tasks, such as shell scripting or JavaScript.

Languages, in any form, are crucial for communication and information exchange, whether between humans, machines, or other forms of life. They are fundamental tools for expressing ideas, conveying information, and building connections within and across different communities.

LLM Transformer Variations



Tokenization

Break down text into indivisible units called tokens that are understood by LLM

Special Tokens

Tokens that are introduced to convey specific information.
Example: <stop>token

Context Length

Maximum number of tokens that can be input to LLM
Example: 4096 tokens

Max New Tokens

Maximum number of new tokens that would be generated for given input tokens

Prompt

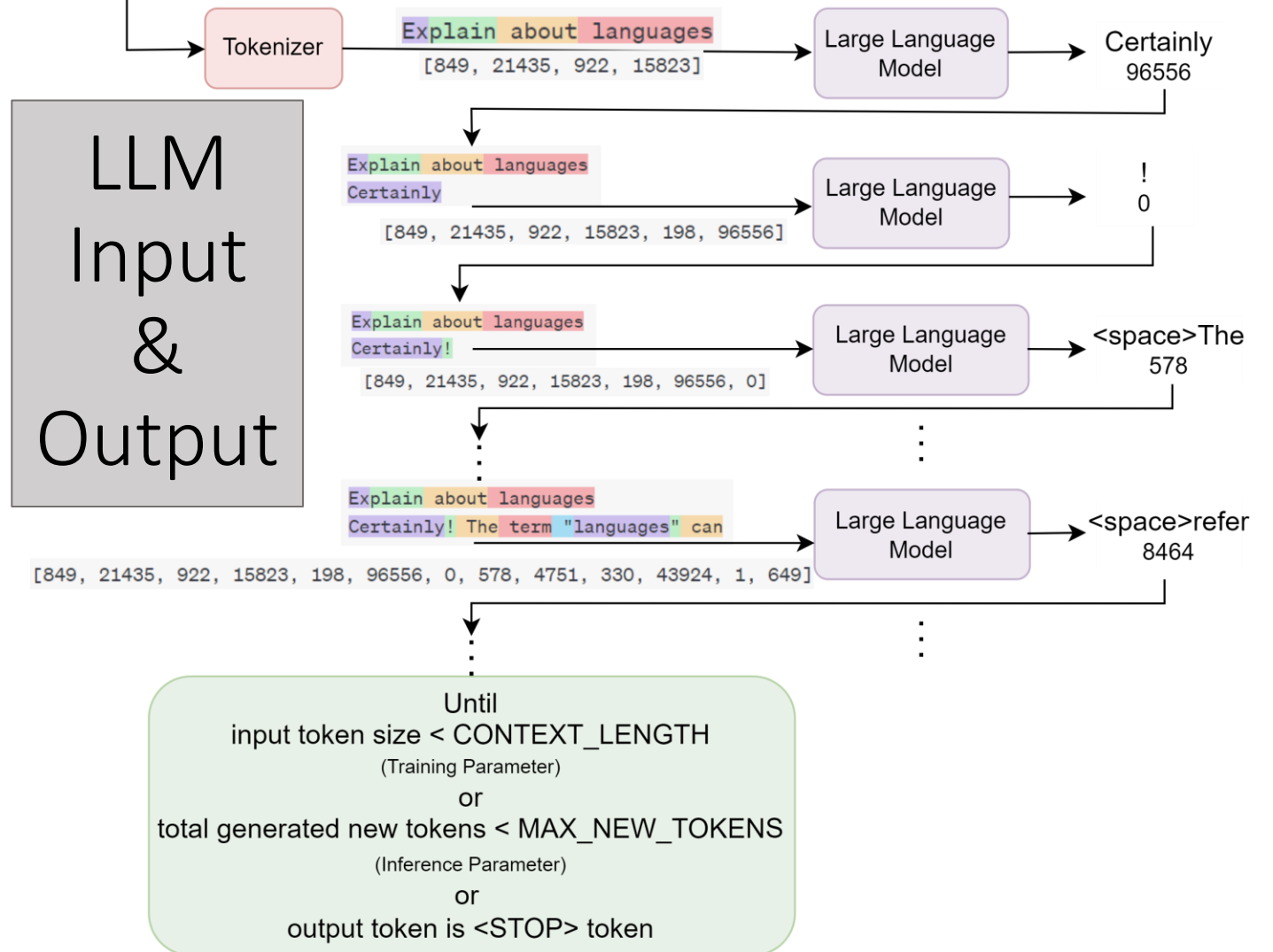
Text input to LLM

Input Text

Explain about languages

Input Tokens

Output Token



Inference Parameter

Assigning Probabilities: SoftMax & Temperature

Softmax

Accepts vector input and outputs probabilities that sum to 1.

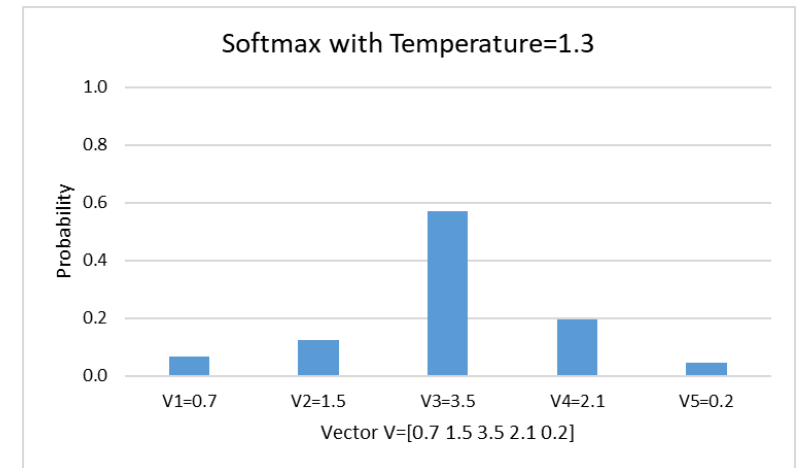
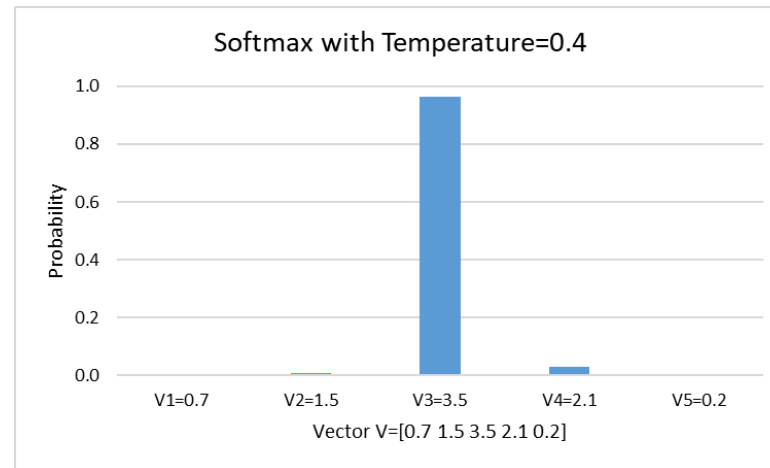
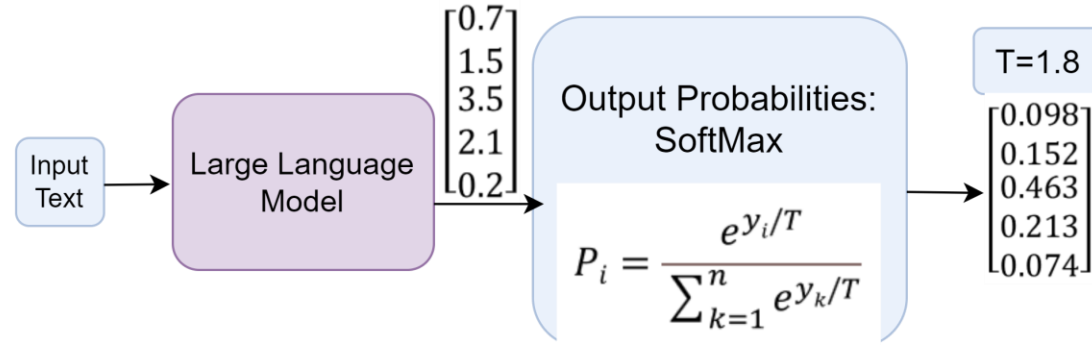
Probability Distribution

Distribution of probability depends on Temperature Value & Input Vector.

Temperature

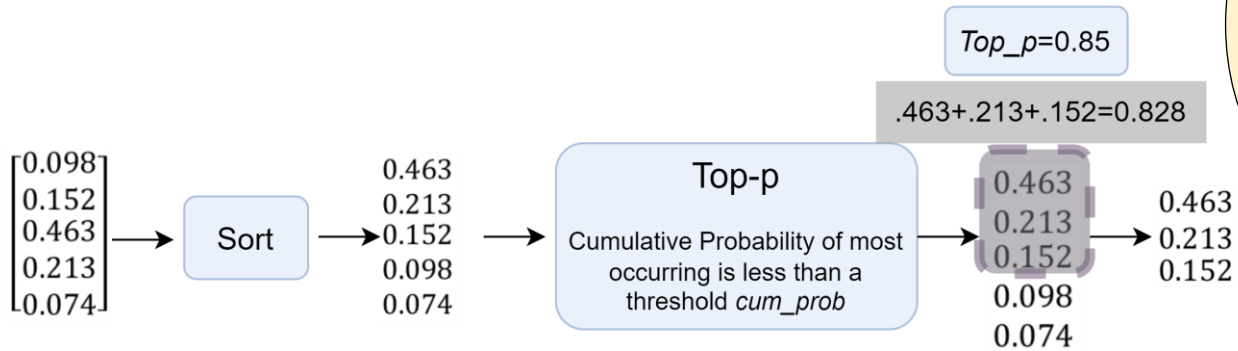
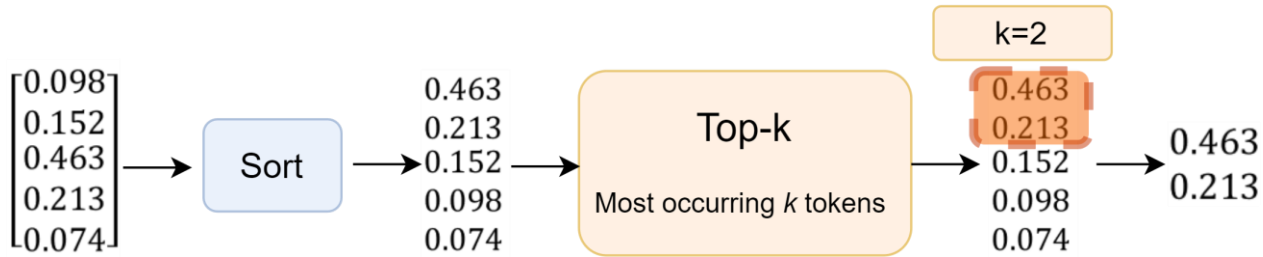
Range [0,2]

Lower value - deterministic tokens
Higher value - creative



Inference Parameter

Token Selection: Top-k, Top-P



Select Tokens from this list

Strategies

- Random selection of finalized tokens
- Weighted selection of finalized tokens

Prompt Engineering

Prompt

Classify the sentiment of the below text.

Text:

Awful experience. I hate new design of the website.

Output

The sentiment of the given text is negative.

Refined Prompt

Classify the sentiment of the below text as very positive, positive, neutral, negative or very negative.

Text:

Awful experience. I hate new design of the website.

Output

The sentiment of the given text is negative.

Prompt Engineering

- Ask query in certain pattern to obtain desired results
- No change in model weights
- Provide examples with different scenarios
- Provide sufficient data to obtain desired results

Prompt Engineering : ICL

Refined Prompt with few sample

Classify the sentiment of the below text as very positive, positive, neutral, negative or very negative.

Text: It's been amazing experience browsing through the website.

Sentiment: Very Positive

Text: Experience while browsing has not been that great.

Sentiment: Negative

Text: Browsing this website is neither good nor bad.

Sentiment: Neutral

Text:

Awful experience. I hate new design of the website.

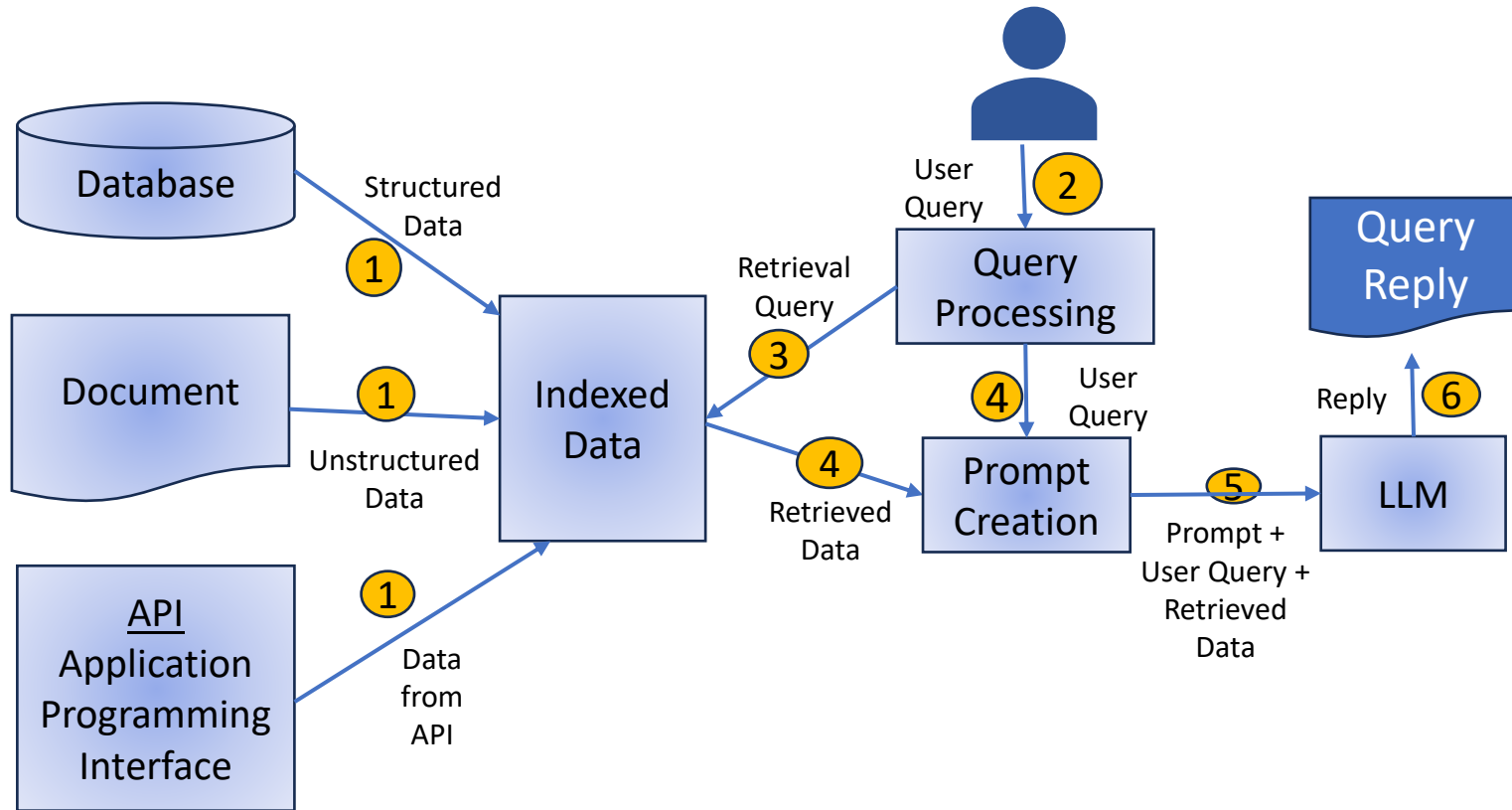
Output

Sentiment: Very Negative

ICL In-Context Learning

- Steering output of LLM towards desired results using examples.
- Also called few-shot learning.
- No change in model weights

Prompt Engineering: RAG



RAG Retrieval Augmented Generation

- Retrieve information relevant to user query.
- Embed retrieved information in prompt template along with user query



Part II – SystemC Code Generation

Challenges Mitigations Demo



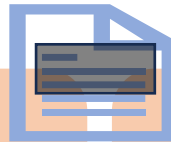
Challenges – SystemC Code Generation



Confidential information about IP peripherals details.



Same IP feature among various vendors but different implementation



Complete explanation of a feature along with ports/registers might overshoot context length of LLM

As input tokens length increases, LLM performance degrades

Mitigations – SystemC Code Generation

Limit Explanation

Use of short hand notation

- Create short hand notation to explain frequently occurring concepts
- Based on concepts used in input query, use short hand notation explanation for only those concepts in prompt creation

RAG

- Retrieve relevant information from documentation for prompt creation
- Summarize or extract relevant info from retrieved data for prompt creation

Few-Shot

- Use relevant or nearest samples from the input query for prompt creation

2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

DEMO

