

Prompt engineering

How to think like an AI

INCOMPLETE
GUIDEBOOK
V.2

Titayna Kauffmann Will

titayna.kauffmann@uni.lu

C²DH

March 2025

OVERVIEW

1. Definitions

2. Limitations

3. Best practices

Steps of prompt engineering

Clarity

Context

Precision

Examples



4. Learning NLP techniques

Role prompting

Multishot prompting

Typology of LLM reasoning

Chain-of-Thought (CoT)

Tree-of-Thought (ToT)

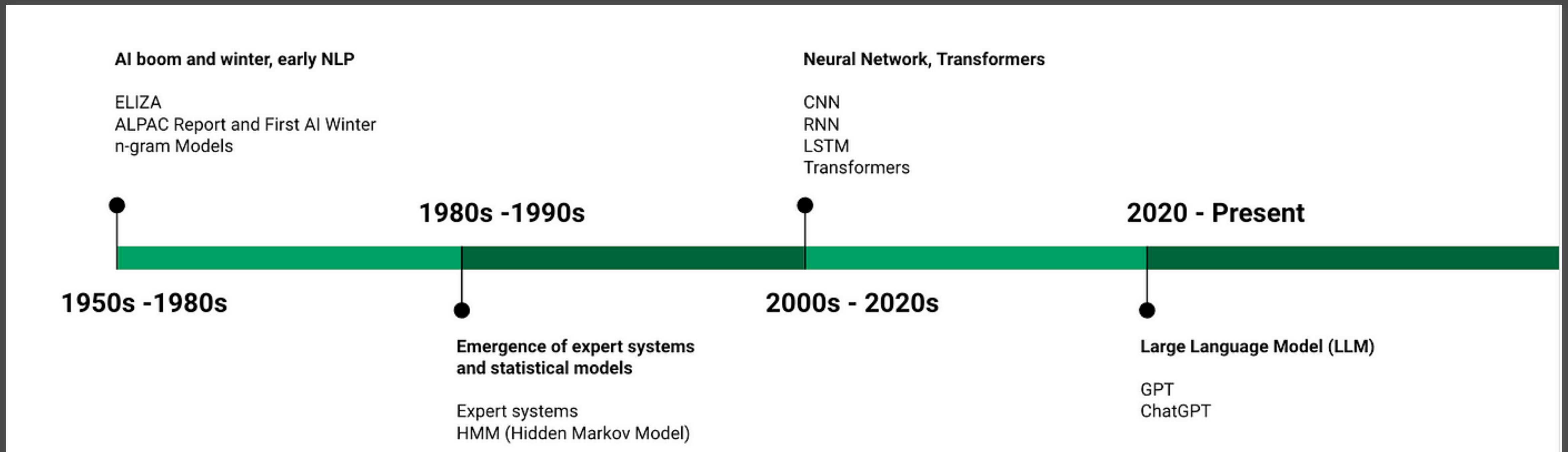
Graph-of-Thought (GoT)



Definitions

WHAT'S A LARGE LANGUAGE MODEL ?

A mathematical model of human language ...



HMM (Hidden Markov Model)
Expert systems

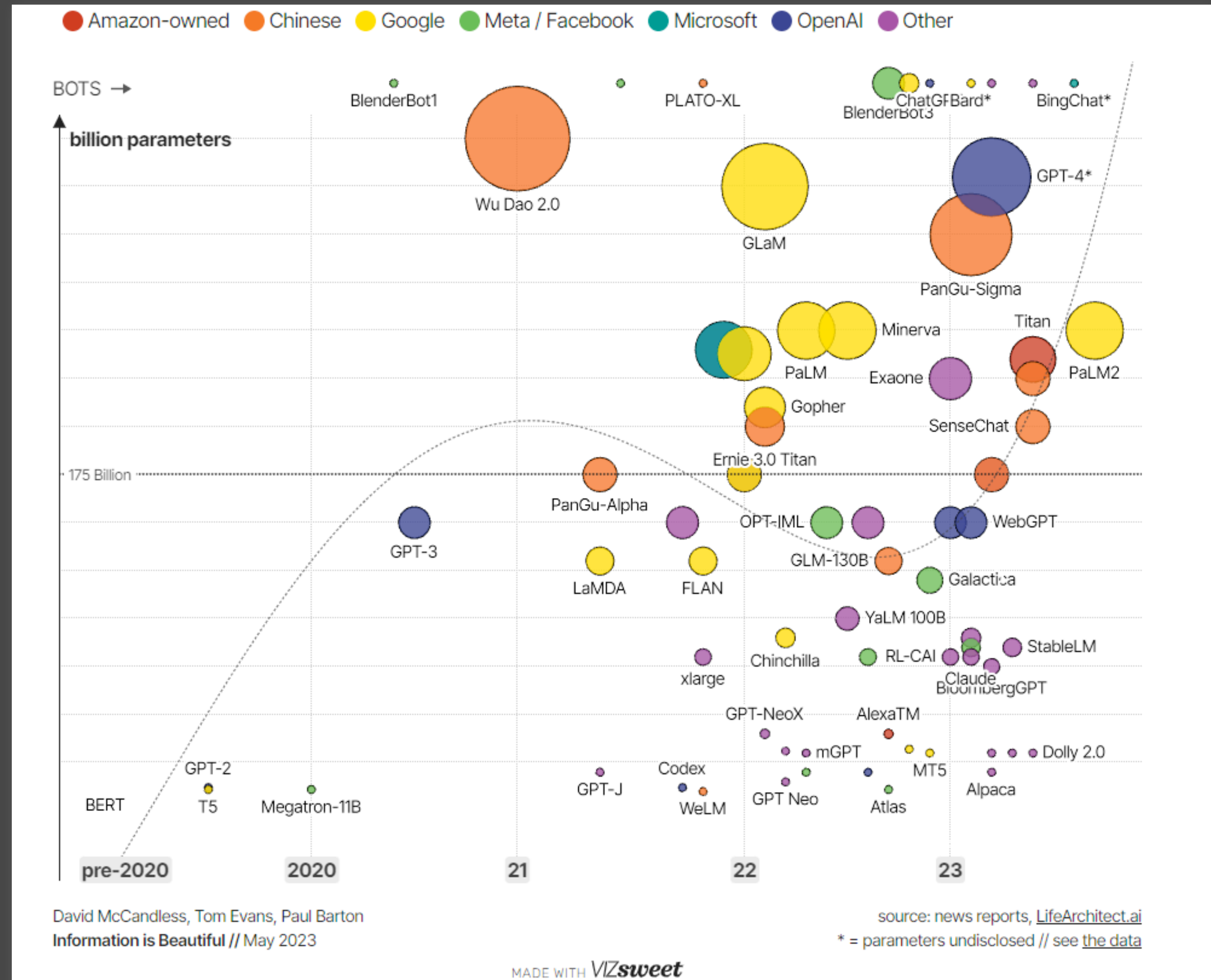
and statistical models

ChatGPT
GPT

medium.com

WHAT'S A LARGE LANGUAGE MODEL?

... with billions
of parameters



WHAT IS PROMPT ENGINEERING?

“Prompt engineering is a relatively new discipline for developing and optimizing prompts to **efficiently use language models** (LMs) for a wide variety of applications and research topics. Prompt engineering skills help to **better understand the capabilities and limitations** of large language models (LLMs).”

WHAT IS A PROMPT ?

“A prompt, sometimes referred to as **context**, is the text provided to a model before it begins generating output.”

(Brexhq/Prompt-Engineering 2023)

“A prompt is a **input** that produces the desired **output**.”

(IBM class on EdX)

“Based on **natural language** requests submitted as prompts, generative AI models collect informations, derive inferences and provide creative solutions.

(IBM class on EdX)

Limitations

RESULTS HIGHLY DEPENDENT ON TRAINING DATA

Essentially, their core principle is still
"using existing information to predict other information".

- If the **training data contains biases** (such as racist or sexist content) or incorrect information, the large language model will generate similarly flawed results.
- When **data is limited in a particular domain**, the model's results become significantly less accurate. In such cases, the model may even generate hallucinations.



LLM

26.02.2025 -
ChatGPT4o :

18 men,
2 women
(Mother Teresa
& Marie Curie)

List examples of great leaders in
history.

Alexander the Great, Winston
Churchill, George Washington...

Cleopatra, Nefertiti, Elizabeth I,
Joan of Arc, Harriet Tubman, Susan B.
Anthony, Emmeline Pankhurst, Rosa
Parks, Queen Nzinga, ...



USER



BOOKS

HALLUCINATIONS

LLM are mathematical models. Their interest is for the **most probable answer**, not the truth.

Type of hallucinations :

- Factual inaccuracies
- Generated quotations or sources
- Logical inconsistencies



learningprompt.wiki

Can you provide me a scientific paper that discusses the effects of social media on teenage mental health?



USER

Sure! A great paper on this topic is:

Smith, J., & Patel, R. (2021). The Psychological Impact of Social Media on Adolescents: A Longitudinal Study. *Journal of Adolescent Psychology*, 35(4), 112-129.



LLM

PROMPT ENGINEERING TO MITIGATE THE RISKS

Prompt engineering tools offer features that **help mitigate bias in generative AI** model responses, provide guidance for crafting prompts that reduce biased or inappropriate outputs, and assist in creating domain-specific prompts.

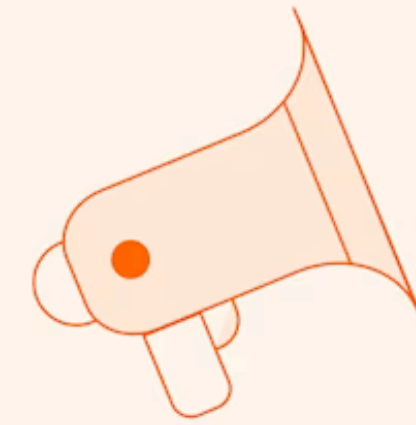
Best practices

NON-TECH PROMPT ENGINEERING SKILLS

5 non-tech prompt engineering skills



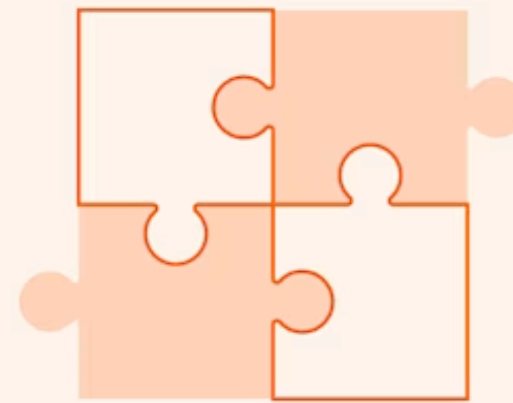
Language



Communication



Creativity



Critical thinking



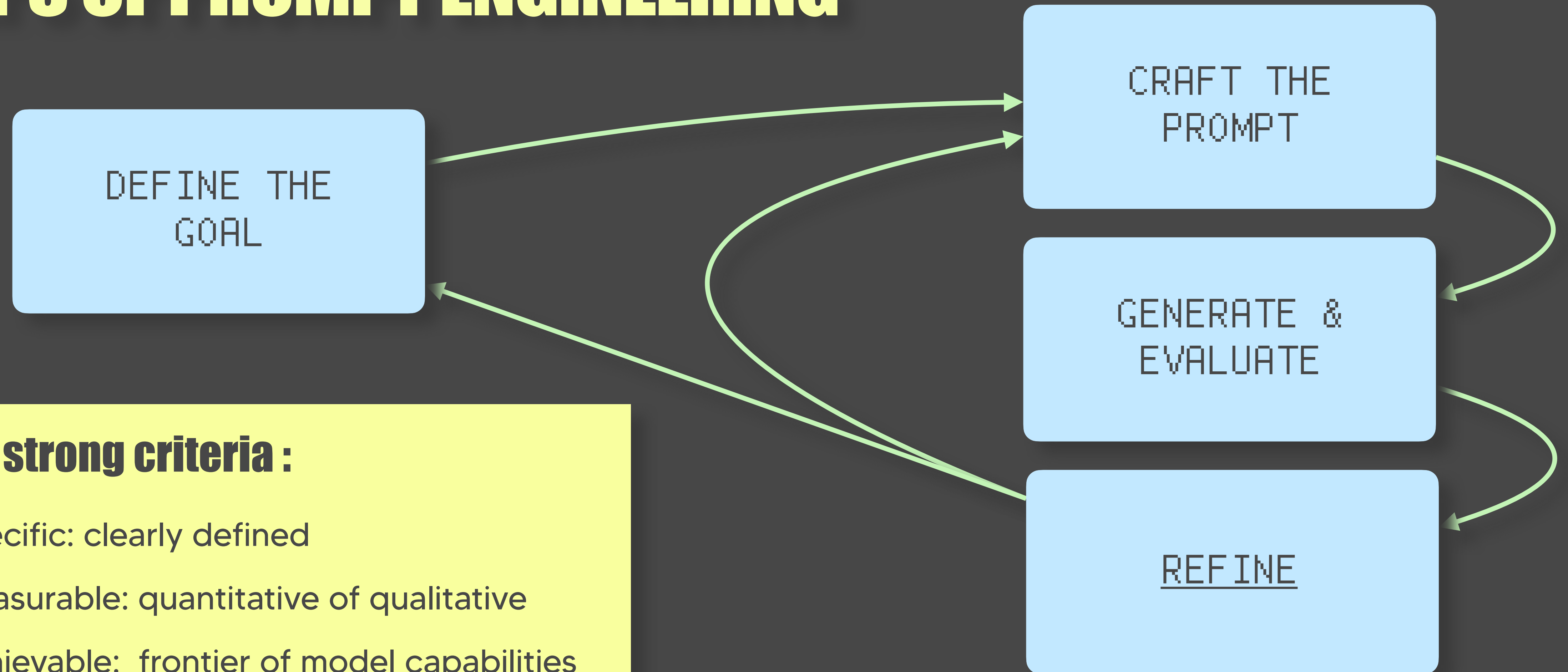
Subject matter expertise

 **zapier**

 **zapier**

zapier.com

STEPS OF PROMPT ENGINEERING



Build strong criteria :

- Specific: clearly defined
- Measurable: quantitative or qualitative
- Achievable: frontier of model capabilities
- Relevant: for your purposes

Anthropic Guidelines

Inspired by Cobus Greyling

4 CENTRAL ASPECTS

Clarity

Context

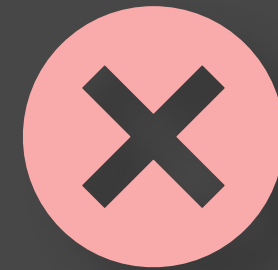
Precision

Example

CLARITY

- Use simple, **straightforward** language to convey instructions clearly. Your prompts should be unambiguous and easy to understand.
- **Set constraints.** When prompts are vague, they often lead to responses that miss your intended goal.
- Always provide **clear, specific** descriptions of the tasks you want the model to perform.
- Clear does **not mean short.**

Write a Python function for text processing.



USER



LLM

```
def process_text(text):  
    return text.split()
```

Write a Python function that tokenizes an input text by splitting on spaces and removing punctuation. The function should return a list of words.



USER



LLM

```
import string  
  
def tokenize(text):  
    text = text.lower()  
    text = text.translate(str.maketrans("", "",  
    string.punctuation))  
    return text.split()
```

CONTEXT

Providing context **helps the model** understand the situation or subject.

Examples of contextual information :

- What the task results will be used for.
- What audience the output is meant for.
- What workflow the task is a part of, and where this task belongs in that workflow.
- The end goal of the task, or what a successful task completion looks like.

Anthropic Guidelines

PRECISION (INPUT)

Delimitation between different part of the instructions

Quotation marks

USER:

Summarize the text
delimited by triple quote.

```
"""insert text here"""
```

XML tags

USER

Summarize the arguments of
these two article.

```
<article> insert first article  
here </article>
```

```
<article> insert second article  
here </article>
```

Section titles

USER:

You will be provided with
an abstract and an
article. Find possible
titles.

```
Abstract: insert abstract here
```

```
Article: insert article here
```

“The more complex a task is the more important it is to disambiguate task details. Don’t make the model work to understand exactly what you are asking of them.”

PRECISION (OUTPUT)

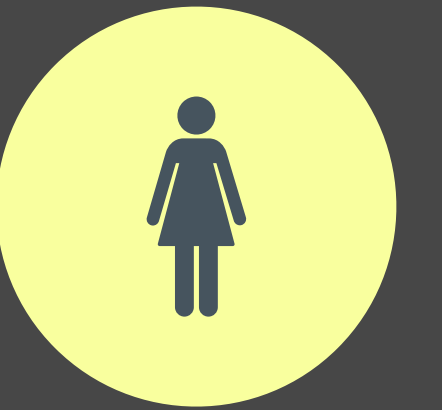
- **Format:** Define how the model should answer (JSON, table, bullet point, etc.)
- **Specify level of detail:** Whether it should be concise, detailed, or summarised.
- **Word or sentence limits:** Ask for responses of a specific length
- **Example Output Templates:** Demonstrate the desired response format by providing an example.



LLM

Analyze the following literary work and provide a structured response. Your answer should be organized into the **following sections**:

1. **Title and Author** - Provide the full title and author's name.
2. **Summary** - A **concise**, five-sentence summary of the main plot or argument.
3. **Themes** - A bullet-point list of the key themes explored in the work.
4. **Literary Devices** - Identify and explain three key literary devices used by the author.
5. **Historical Context** - Provide a brief, two-sentence explanation of the historical or cultural background relevant to the text.



USER

Title and Author:

Pride and Prejudice by Jane Austen

Summary:

Elizabeth Bennet, a strong-willed young woman, navigates issues of class and marriage in 19th-century England. She initially dislikes Mr. Darcy, whom she finds arrogant, but later discovers his true character. Misunderstandings and social expectations challenge their relationship. Eventually, they recognize their mutual respect and love. The novel

EXAMPLES

Provide reference text

“If you intend for the model to copy a particular style of responding to user queries which is difficult to describe explicitly. This is known as **"few-shot" prompting.**”

“If we can provide a model with **trusted information** that is relevant to the current query, then we can instruct the model to use the provided information to compose its answer. Given that all models have **limited context windows**, we need some way to dynamically lookup information that is relevant to the question being asked. **Embeddings** can be used to implement efficient knowledge retrieval.”

SYSTEM SETTINGS :

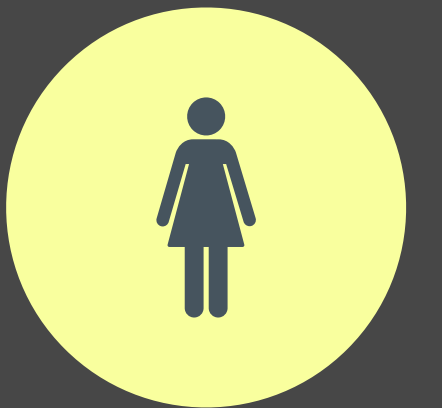
Answer in the style of Ernest Hemingway—concise, direct,
and powerful.

Tell me about courage.



LLM

Courage is grace under pressure.
It's standing when your legs
shake, speaking when your voice
trembles, and moving forward when
fear grips your heart



USER

SYSTEM SETTINGS :

Use the provided articles delimited by triple quotes to answer questions. If the answer cannot be found in the articles, write "I could not find an answer."

<insert articles, each delimited by triple
quotes>

Question: <insert question here>



USER

EXAMPLES

- **Accuracy:** Examples reduce misinterpretation of instructions.
- **Consistency:** Examples enforce uniform structure and style.
- **Performance:** Well-chosen examples boost Claude's ability to handle complex tasks.
- **Relevant:** Your examples mirror your actual use case.
- **Diverse:** Your examples cover edge cases and potential challenges, and vary enough that Claude doesn't inadvertently pick up on unintended patterns.
- **Clear:** Your examples are wrapped in `<example>` tags (if multiple, nested within `<examples>` tags) for structure.

Our peer review process needs better feedback organization. Categorize feedback into **Clarity, Methodology, Sources, Argumentation, Structure, Other**. Also, assess **Sentiment** (Positive/Neutral/Negative) and **Priority** (High/Medium/Low). Here's an example:

Example:

Feedback: The thesis is well-argued, but the conclusion lacks depth.

Category: Argumentation

Sentiment: Neutral

Priority: Medium

Now, analyze this feedback: The paper's use of outdated sources weakens its claims, and the introduction is too vague.



USER



LLM

Category: Sources, Clarity

Sentiment: Negative

Priority: High

NLP techniques

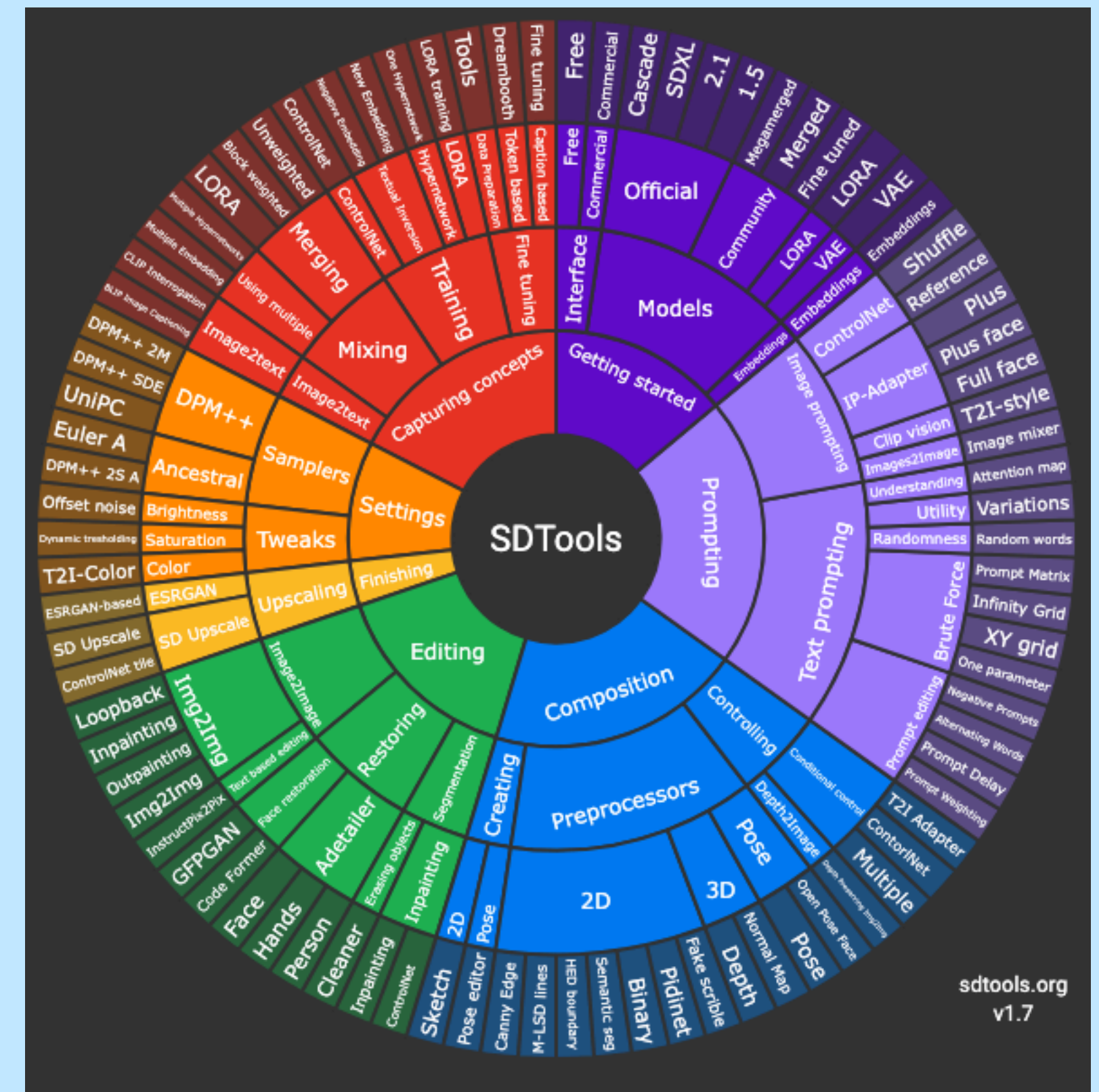
WHAT DOES IT HAVE TO DO WITH NLP ?

Using generative AI involves using Natural Language Processing (NLP), not human language. This means that you'll need to **think about how to communicate with a computer**, rather than attempting to converse with a human. It's crucial to keep in mind that you're **interacting with a machine**, not a human being. Consider it as programming, but with a language that closely resembles human language.

TEXT-TO-TEXT PROMPT TECHNIQUES

Text-to-text prompting is a technique in prompt engineering where both the **input** (prompt) and **output** (response) are structured as **natural language text**.

Text-to-Image prompting



SDTools.org

ROLE PROMPTING

“The Persona Pattern is a prompt engineering technique where the model is instructed to adopt a **specific role, character, or perspective** when generating a response. By framing the prompt in this way, the model tailors its output to align with the expectations, expertise, or tone associated with the given persona.”

[OpenAI Guidelines](#)

“When using Claude, you can **dramatically improve its performance** by using the system parameter to give it a role. This technique, known as role prompting, is the most powerful way to use system prompts with Claude.”

[Anthropic Guidelines](#)

Describe the impact of the French Revolution.



USER



LLM

The French Revolution led to the overthrow of the monarchy, the rise of democracy in France, and widespread social and political changes.

You are a historian specializing in European history. Analyze the impact of the French Revolution on modern democratic movements.



USER



LLM

The French Revolution (1789–1799) marked a turning point in political history, introducing principles of liberty, equality, and fraternity that shaped democratic movements worldwide. The abolition of feudal privileges and the establishment of the Declaration of the Rights of Man laid the foundation for modern constitutional democracies. The revolution also inspired later movements, including the 1848 Revolutions across Europe and the rise of republicanism in the Americas. However, its radical phase, including the Reign of Terror, demonstrated the challenges of rapid political change and the dangers of extremist governance.

THE IMPACT OF FLATTERY ?

Rudeness in prompts tends to yield suboptimal results, which makes sense given that LLMs operate as mathematical models based on probability distributions.

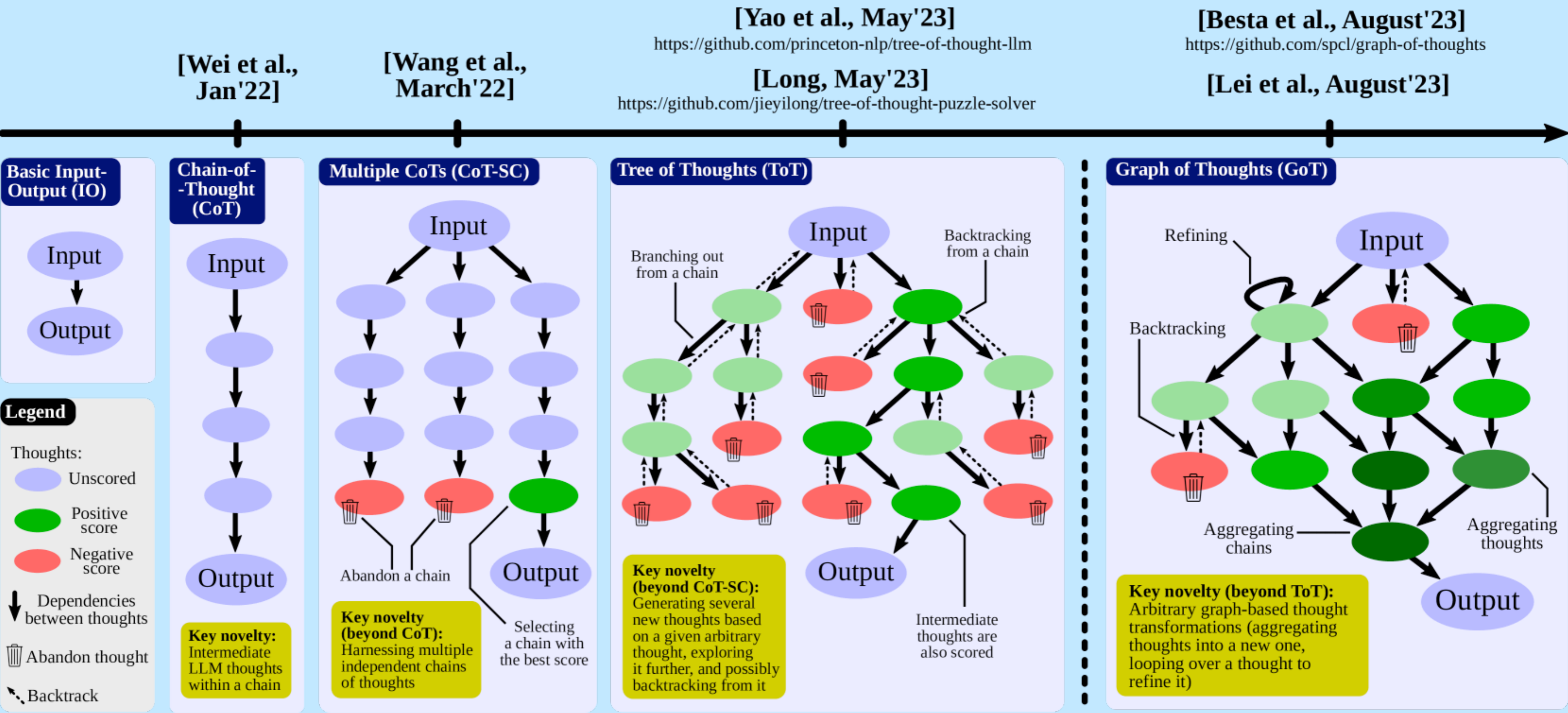
Viswanath 2024

OpenAI's Instructions: “Based on internal evals, the gpt-4.5-preview model has a particular system message that results in better performance. Add your own system message contents after this:”

```
You are a highly capable, thoughtful, and precise assistant. Your goal is to deeply understand the user's intent, ask clarifying questions when needed, think step-by-step through complex problems, provide clear and accurate answers, and proactively anticipate helpful follow-up information. Always prioritize being truthful, nuanced, insightful, and efficient, tailoring your responses specifically to the user's needs and preferences.
```

OpenAI Guidelines

OVERVIEW



MULTISHOT PROMPTING

ZERO-SHOT

Classify the sentiment in this review (positive/negative/neutral):

Review: I loved this movie

The model performs the tasks **without prior example**

ONE-SHOT

Classify the sentiment in this review (positive/negative/neutral):

Review: I hated this movie

Here's an example :

Review: I loved this movie

Sentiment: Positive

The model performs the tasks with **one prior example**

FEW-SHOT

Classify the sentiment in this review (positive/negative/neutral):

Review: I neither loved or hated this movie

Here's some examples :

Review: I loved this movie

Sentiment: Positive

Review: I hated this movie

Sentiment: Negative

The model performs the tasks with **multiple prior examples**

TYPOLGY OF LLM REASONING

CHAIN-BASED

CLASS : Linear sequence.

SCOPE : Guides step-by-step reasoning.

REPRESENTATION : Each step builds on the last, forming a **deterministic** flow.

Fast, simple, but limited to linear logic.

TREE-BASED

CLASS : Hierarchical structure.

SCOPE : Explores multiple solutions before deciding.

REPRESENTATION : Each branch represents an alternative path in reasoning.

Better exploration, but branching increases cost.

GRAPH BASED

CLASS : Network of interconnected nodes.

SCOPE : Flexible, allowing loops, revisits, and bidirectional relationships.

REPRESENTATION : Nodes represent thoughts, edges define relationships.

Maximum flexibility, but highest computational cost.

CHAIN-BASED REASONING

Chain-of-thought (CoT) prompting allows for complex reasoning by breaking down problems into **step-by-step solutions**. When combined with few-shot prompting, this approach **improves performance** on complex reasoning tasks.

WHY CHAIN PROMPTS

Accuracy:

Each subtask gets full attention, reducing errors.

Clarity:

Simpler subtasks mean clearer instructions and outputs.

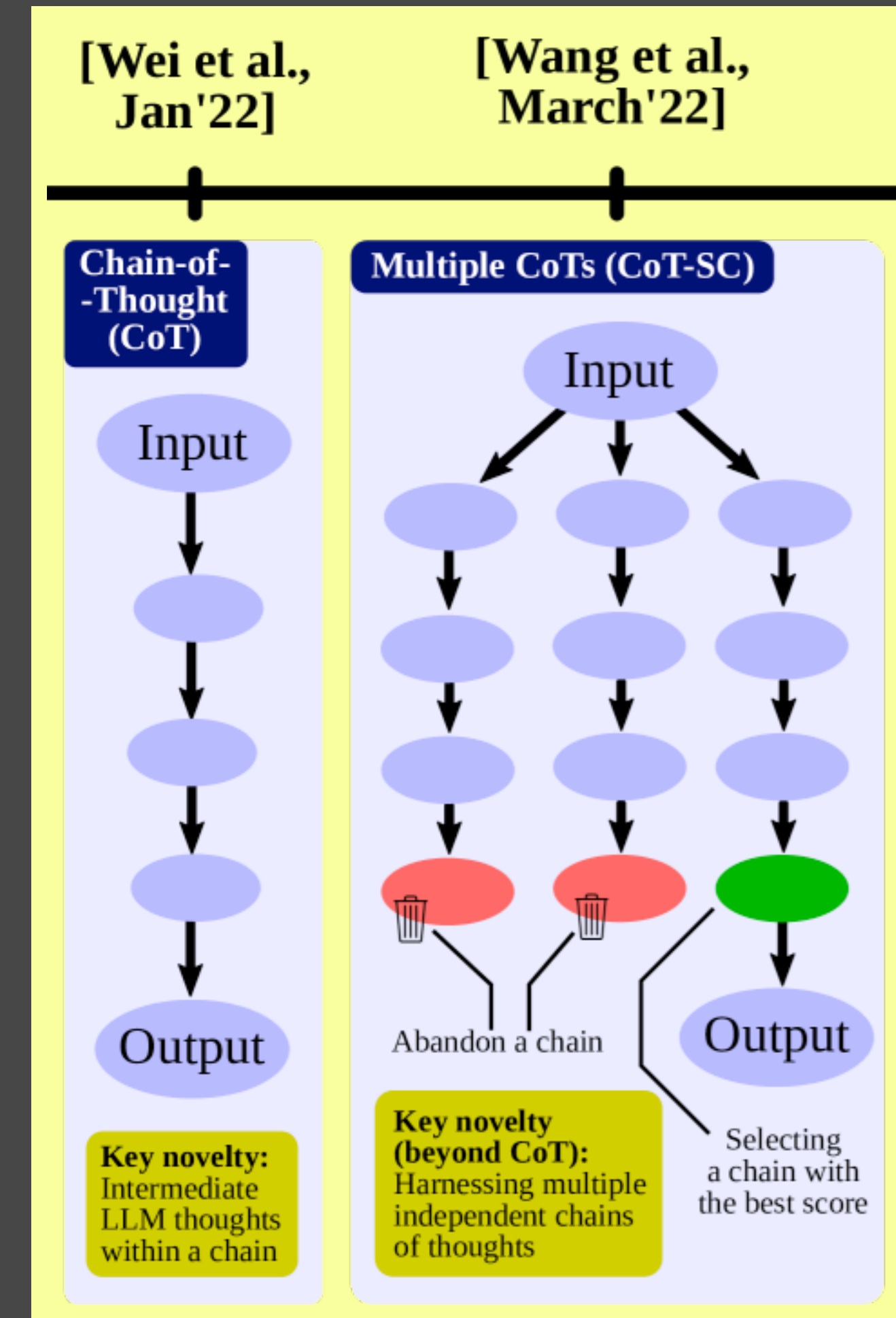
Traceability:

Easily pinpoint and fix issues in your prompt chain.

CHAIN-OF-THOUGHT WITH SELF CONSISTENCY (CoT-SC)

Key Features of Chains

- **Simplicity:** Easy to implement and understand.
- **Single Path:** Limited exploration.
- **Useful for:** Arithmetic, logic, multi-step problems.



Example CoT-SC

Draft personalized email invitations for a conference.

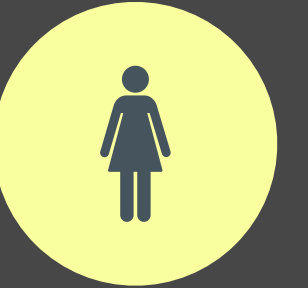
Conference information:

{{CONFERENCE_DETAILS}}

Recipient information:

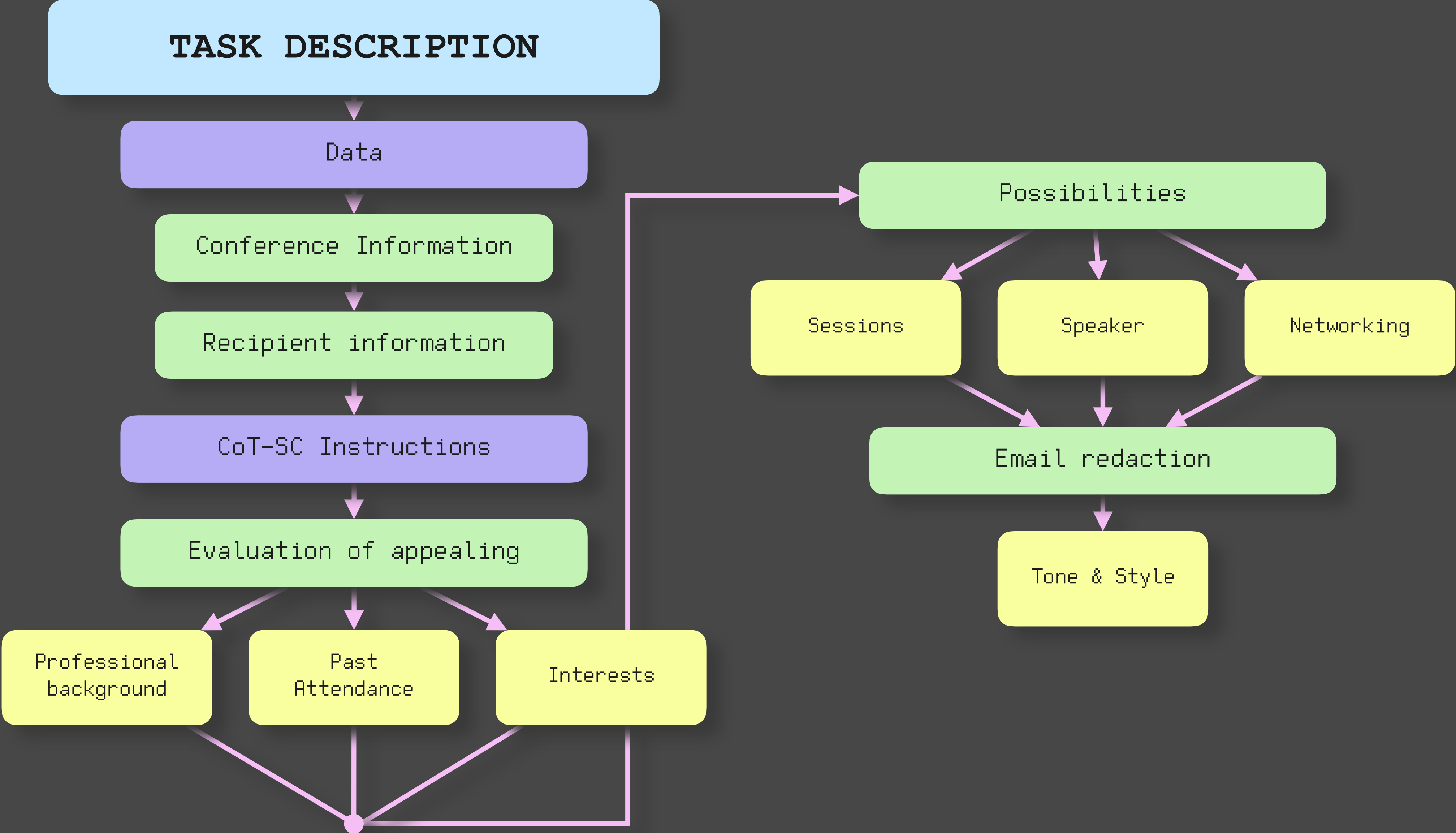
{{RECIPIENT_DETAILS}}

Think before you write the email. First, consider what aspects of the conference would be most appealing to this recipient based on their professional background, past attendance, or interests. Then, determine how to personalize the invitation by highlighting specific sessions, speakers, or networking opportunities that align with their profile. Finally, craft a compelling and engaging email that encourages them to attend while maintaining a warm and professional tone.



USER

Example CoT-SC



CoT Prompting

- **Multi-Step Reasoning** → Splits problems into intermediate steps.
- **Zero-Shot Reasoning Instructions** → Encourages step-wise thinking without examples.
- **Planning & Task Decomposition** → The model explicitly plans before execution.
- **Least-to-Most Prompting** → Solves **simpler problems first**, then generalizes.
- **Decomposed Prompting** → Breaks a complex question into multiple prompts.
- **Task Preprocessing** → Structures input data before model reasoning.
- **Selection-Inference (SI)** → Selects **relevant** facts **before** inference.
- **Chain-of-Symbol (CoS)** → Uses symbolic representations **within chains**.
- **Iterative Refinement** → Revisits & **improves** answers based on feedback.
- **Chain-of-Thought (CoT)** → Generates explicit reasoning steps.
- **Chain-of-Thought with Self-Consistency (CoT-SC)** → Samples multiple reasoning chains and selects the most common answer.

TREE-OF-THOUGHT (ToT)

Key Features of Trees

- **Allows branching:** Explores multiple reasoning paths.
- **Efficient pruning:** Removes unproductive paths.
- **Useful for:** Creative problem-solving, decision-making, code synthesis.

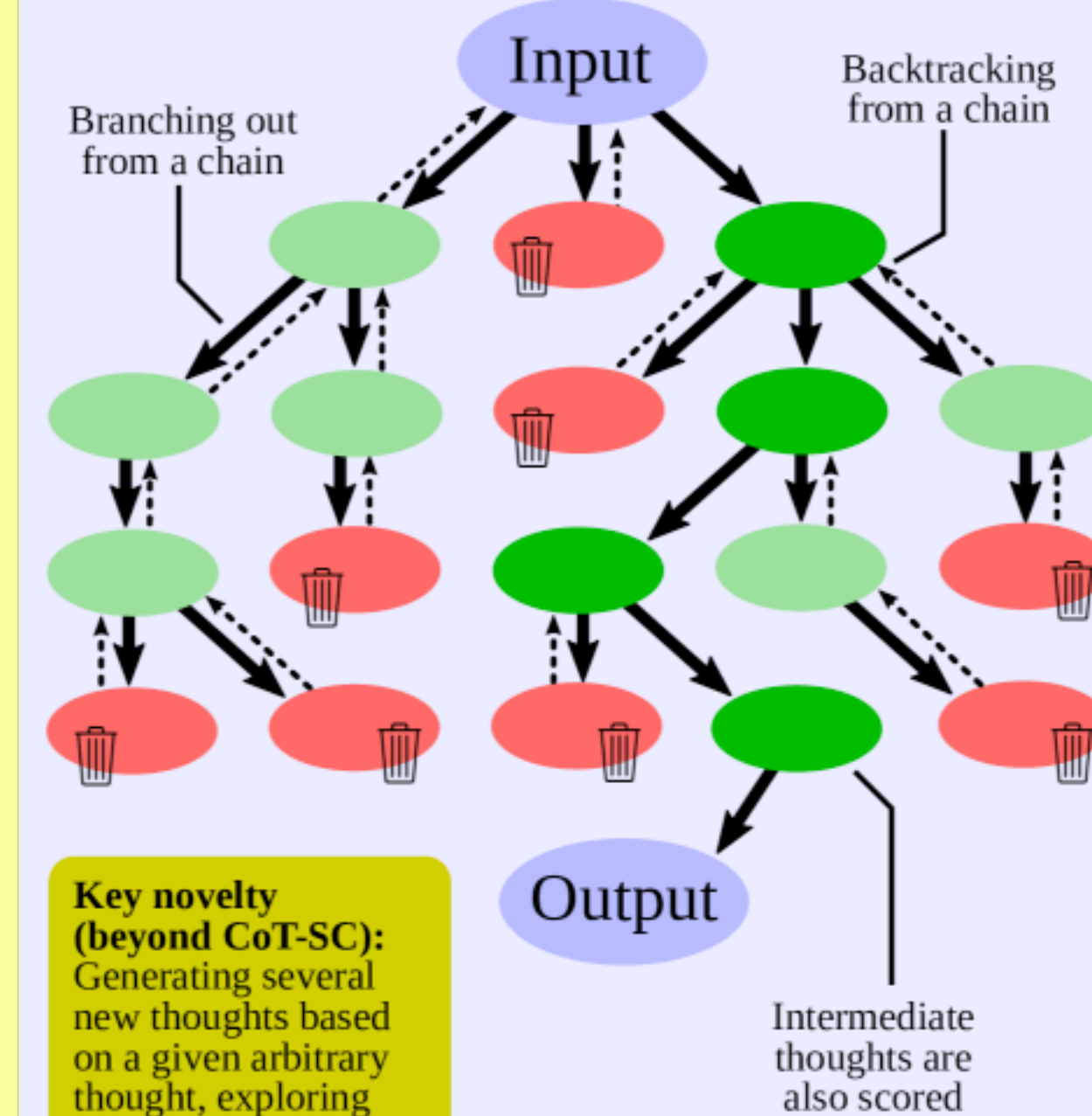
[Yao et al., May'23]

<https://github.com/princeton-nlp/tree-of-thought-llm>

[Long, May'23]

<https://github.com/jieyilong/tree-of-thought-puzzle-solver>

Tree of Thoughts (ToT)



Example ToT

Draft personalized email invitations for a conference.

Conference information:

{{CONFERENCE_DETAILS}}

Recipient information:

{{RECIPIENT_DETAILS}}

Step 1: Three Parallel Streams of Thought (Before Writing the Email)

1. Evaluation of Appeal (What makes the event relevant to the recipient?)

Assess what aspects of the conference would be most appealing based on:

- Their academic background (field of study, research interests, teaching focus)
- Their past attendance at similar academic conferences (new vs. returning attendee)
- Their specific research interests or career stage (early-career researcher, established scholar, graduate student)

2. Identifying Key Personalization Elements (What to highlight?)

Select the most relevant features of the conference:

- Sessions related to their research field, methodology, or interdisciplinary interests
- Speakers who are influential in their domain (e.g., keynote academics, renowned scholars)

- Networking opportunities such as research collaboration forums, mentorship sessions, or funding/grant discussions

3. Strategic & Academic Appeal (How to persuade them?)

Adjust the tone based on their academic role:

- Formal and research-focused for senior professors and tenured faculty
- Engaging and opportunity-driven for early-career researchers and postdocs
- Encouraging and aspirational for graduate students or first-time attendees

Create a sense of urgency (e.g., abstract submission deadlines, limited registration slots for workshops).

Step 2: Generate the Personalized Email

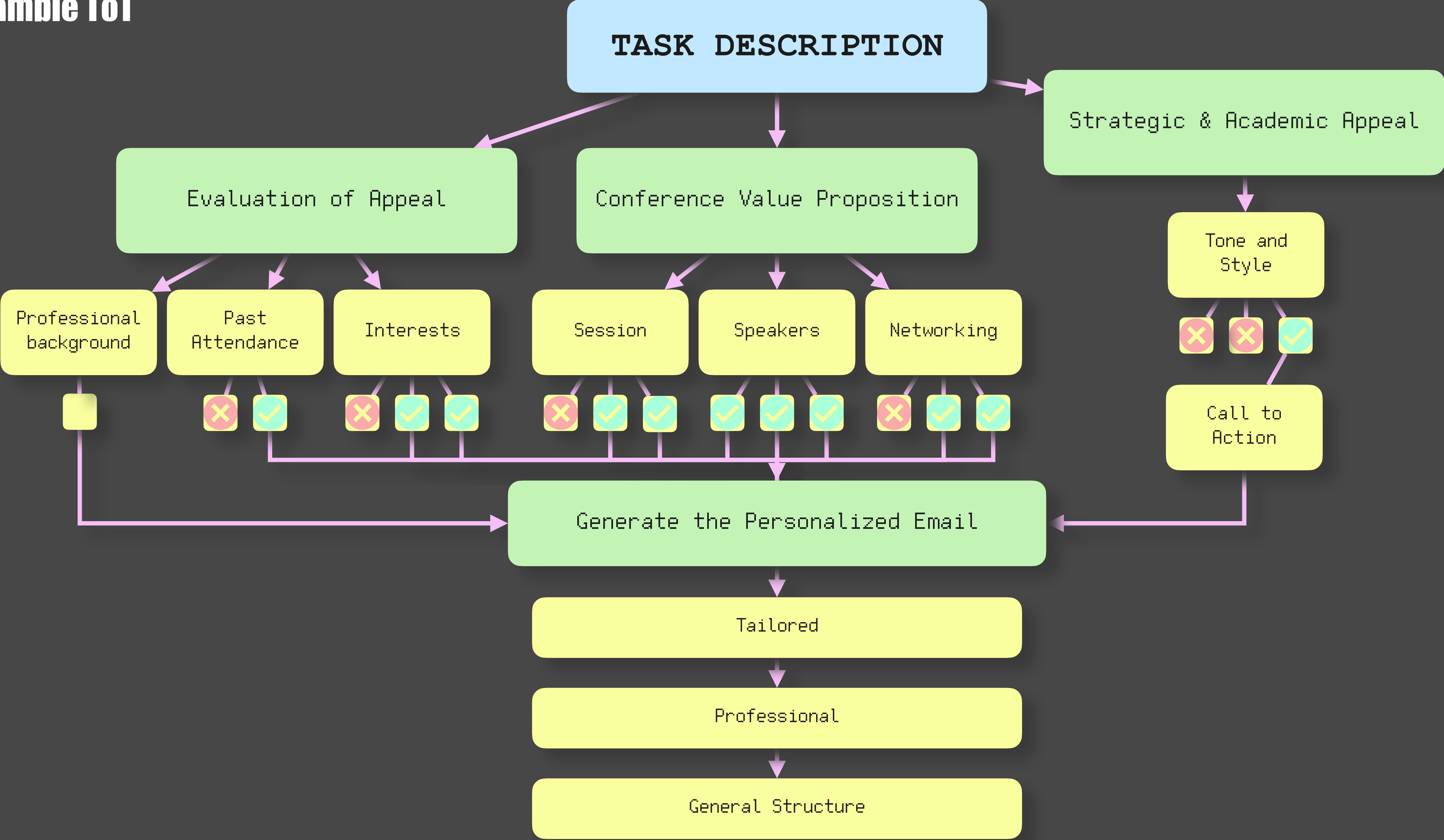
Using insights from the three thought branches, construct an email that is:

- Tailored – Feels relevant and specific to the recipient’s academic profile.
- Professional – Uses an appropriate scholarly tone.
- Engaging – Clearly conveys the benefits of attending.
- Open with an attention-grabbing introduction that connects to their academic work.
- Clearly outline why attending is valuable for their research or professional growth.
- Conclude with a strong and motivating call to action, such as submitting an abstract, registering early, or joining a key discussion.



USER

Example ToT



ToT Prompting

- **Trees of Chains / Chain-of-Thought with Self-Consistency (CoT-SC)** → Extends CoT into multiple chains, selecting the most reliable one.
- **Single-Level Trees / Skeleton-of-Thought (SoT)** → Forms a coarse outline before full reasoning.
- **k-Ary Trees** → Nodes have k children, controlling tree depth.
- **Tree-of-Thought (ToT) by Long** → The model explores and prunes weak paths dynamically.
- **Tree of Thoughts (ToT) by Yao et al.** → Evaluates different reasoning paths before selecting.
- **Thought Decomposition** → Breaks down complex tasks into sub-thoughts.
- **Dynamic Least-to-Most Prompting** → Adapts difficulty dynamically.
- **Algorithm of Thoughts (AoT)** → Formalizes reasoning like an algorithm.
- **Tree of Uncertain Thought (TouT)** → Handles uncertain reasoning paths.
- **Tree-of-Mixed-Thought (TomT)** → Mixes multiple prompting strategies in a tree.
- **Tree of Clarifications (ToC)** → Generates clarifying questions at each branch.

GRAPH-OF-THOUGHT (GOT)

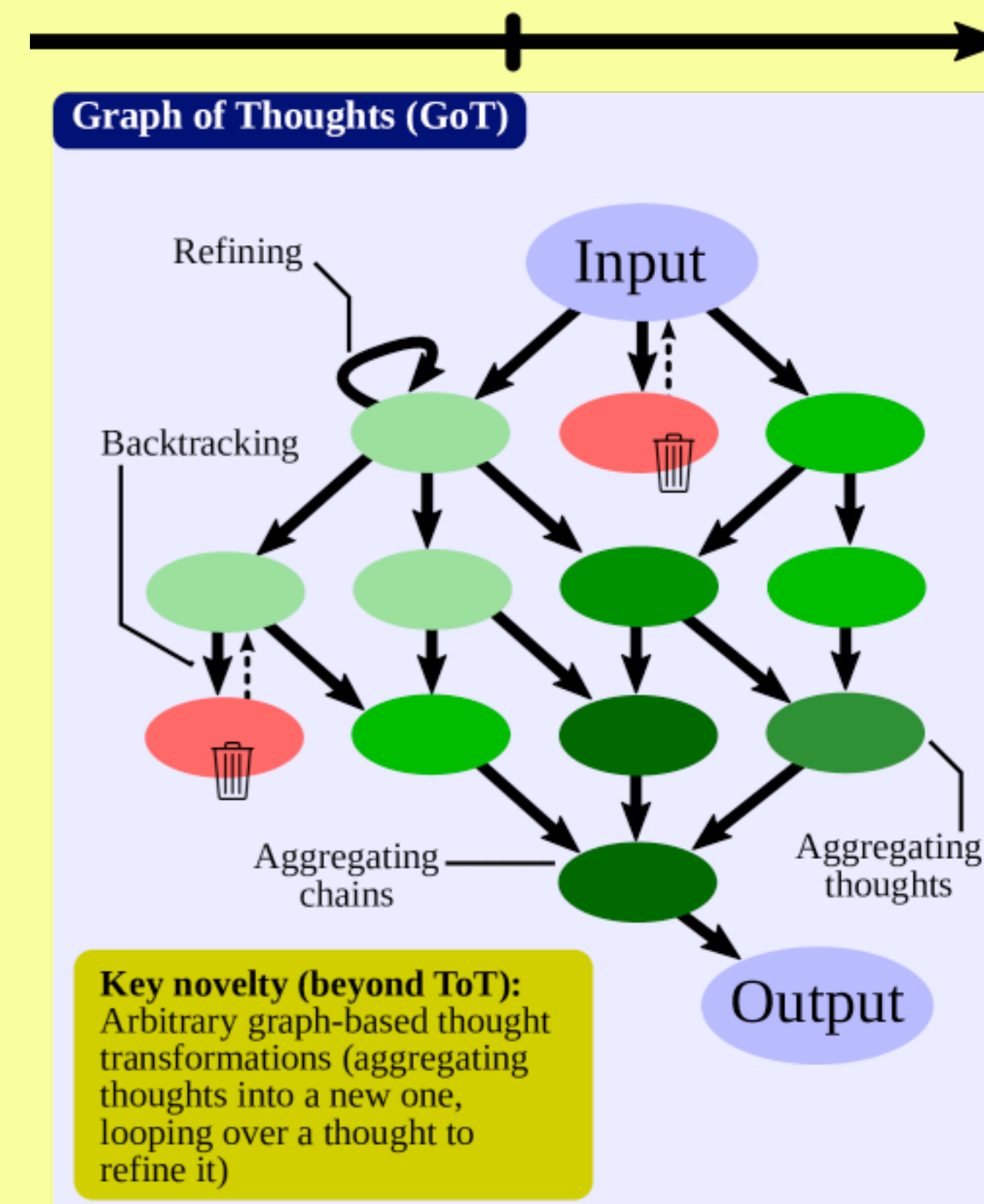
Key Features of Graphs

- **Highly flexible:** Unlike trees, graphs allow cycles and dependencies.
- **Handles ambiguity:** Great for tasks with uncertain or conflicting inputs.
- **Useful for:** Complex decision-making, research synthesis, knowledge retrieval.

[Besta et al., August'23]

<https://github.com/spcl/graph-of-thoughts>

[Lei et al., August'23]



Math Reasoning

Propose Prompt

Suppose you are one of the greatest AI scientists, logicians and mathematicians. Let us think step by step. **Please use First-Order Logic (FOL) to deduce a "Proposition" from two given "Premises". Please make sure that the "Proposition" is logically correct. Please make sure that the "Proposition" is not a duplicate of the "Premises". Please make sure your reasoning is directly deduced from the "Premises" and "Propositions" other than introducing unsourced common knowledge and unsourced information by common sense reasoning.**
[in-context examples]

Premises: [...]
Question: [...]

He had \$110 in total before buying the shoes.

Verifier Prompts

Validate Step

Suppose you are one of the greatest AI scientists, logicians and mathematicians. Let us think step by step. **Please determine whether the "Proposition" is directly deduced from the "Premises" with certainty other than introducing unsourced information by common sense reasoning, reply with True or False.**
[in-context examples]

Premises: [...]
Proposition: He had \$95 in total before buying the shoes.

Judgment:
False

Verify Step

Suppose you are one of the greatest AI scientists, logicians and mathematicians. Let us think step by step. **Please use First-Order Logic (FOL) to determine whether the deduction of two given "Premises" to a "Proposition" is valid or not, reply with True or False.**
[in-context examples]

Premises: [...]
Proposition: He had \$95 in total before buying the shoes.

Judgment:
False

The proposition is not added to the valid node list.

Each premise is a valid node at the beginning. In each iteration five valid nodes are selected at random to generate a proposition. Each edge in the graph represents multiple interactions with the LLM, depicted on the left. Each proposition is validated and verified before adding it to the valid node list, judging if this is deduced from valid nodes and verified based on FOL or if it is a dead end (dashed nodes). After having a predefined number of propositions, the Reporter is invoked to answer the question with all valid nodes.

User Input

Premises:
1. Tobias is buying a new pair of shoes that costs \$95.
[...]
Question: How many driveways did he shovel?

He had \$110 in total
before buying the shoes.

He had \$75 from saving
the last three months and
mowing lawns.

He had \$95 in total
before buying the shoes.

...

He earned \$35 from
shoveling driveways.

We know that he earned \$35 from shoveling, and
he charges \$7 for shoveling. Therefore, the
number of driveways he shoveled is $35 / \$7 = 5$.

Reporter Prompt

Suppose you are one of the greatest AI scientists, logicians and mathematicians. Let us think step by step. **Read and analyze the "Premises" first, then using First-Order Logic (FOL) to answer the "Question". Please make sure your reasoning is directly deduced from the "Premises" and "Propositions" other than introducing unsourced common knowledge and unsourced information by common sense reasoning.**
[in-context examples]

Premises:
1. Tobias is buying a new pair of shoes that costs \$95.
2. He has been saving up his money each month
[...]
Question: How many driveways did he shovel?
Generated Propositions:
1. He had \$110 in total before buying the shoes.
2. He had \$75 from saving the last three months and mowing lawns.
3. Using 1 and 2, we know that he earned \$35 from shoveling driveways.
Reasoning:

We know that he earned \$35 from shoveling, and he charges \$7 for shoveling. Therefore, the number of driveways he shoveled is $35 / \$7 = 5$.

GoT PROMPTING

- **Graph of Thoughts (GoT)** → Models reasoning as a graph, allowing exploration of interdependent ideas.
- **Cumulative Reasoning** → The model builds on past reasoning to improve output.
- **Hypergraph-of-Thought (HoT)** → Edges connect multiple nodes (instead of just pairs), handling multi-modal reasoning.
- **Graph of Deliberation (GoD)** → Weighs different thoughts before choosing.
- **Graph-of-Multiple-Strategies (GoMS)** → Simultaneously tests different strategies.
- **Graph of Uncertain Thoughts (GoUT)** → Deals with conflicting or uncertain reasoning.
- **Graph of Revisable Thought (GoRT)** → Revises previous errors dynamically.
- **Graph-Based Least-to-Most (GB-LTM)** → Extends Least-to-Most prompting to a graph format.
- **Bayesian Graph-of-Thoughts (BGoT)** → Uses Bayesian models to weigh beliefs at each step.
- **Graph of Contradictions (GoC)** → Resolves contradictory information across multiple paths.