

DEMYSTIFYING GENERATIVE AI IN YOUR BUSINESS

WHILE OTHERS TALK,
DO IT WITH AMAZON
BEDROCK



ABOUT US



Valeriiia Voitenko

Software Development
Manager

AWS Database
Migration Service



Maryna Shevchenko

Quality Assurance
Manager

Amazon Ring



Anna Dabrowska

Software Development
Engineer

Artificial Generative
Intelligence



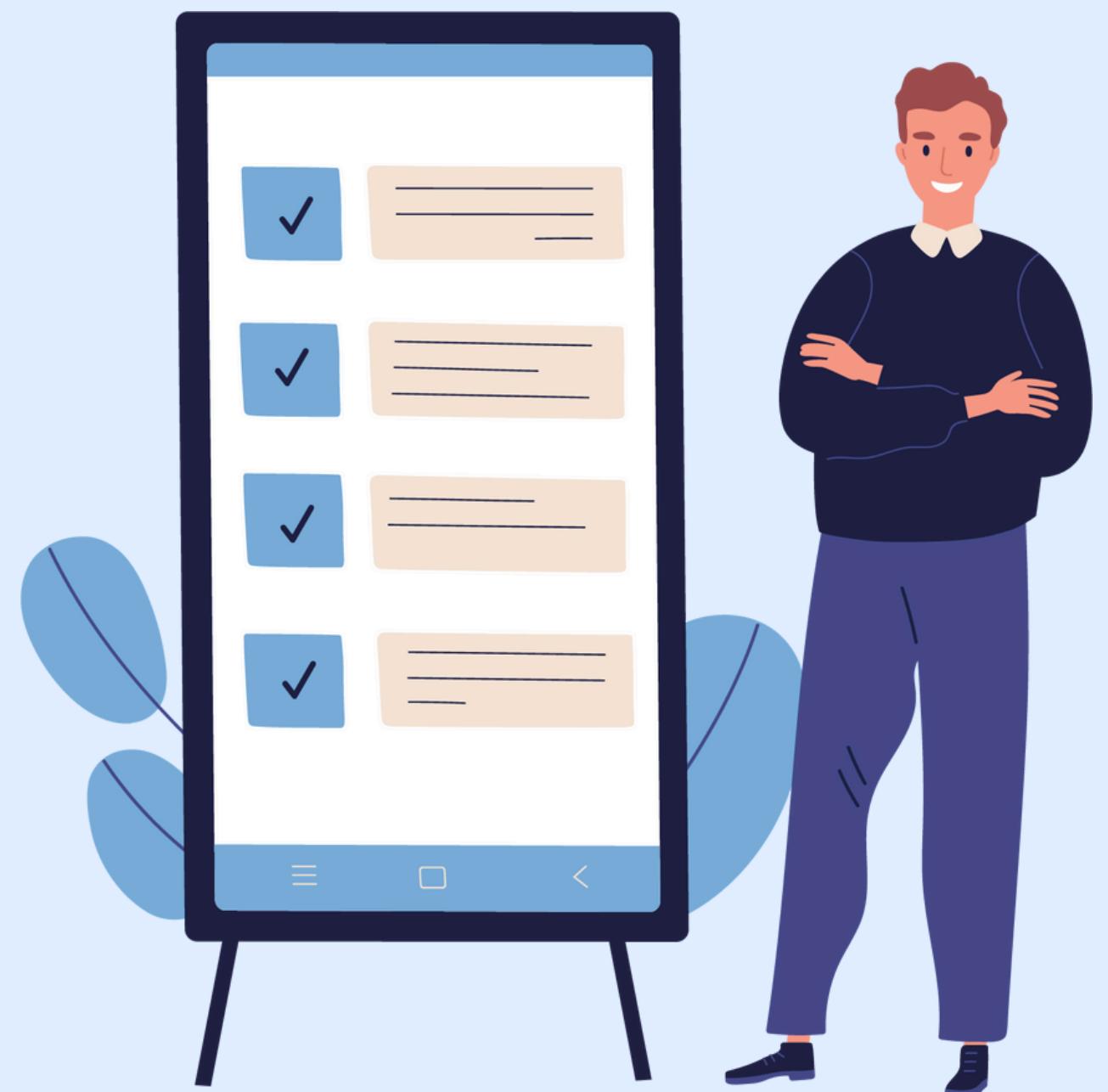
Oleksandra Krykhtina

Software Development
Engineer

AWS Database
Migration Service

AGENDA

- Generative AI vs. Machine learning
- Generative AI in business and around us
- Basics of prompt engineering
- Real life example: OMG properties booking
chat with Amazon Bedrock





GENERATIVE AI VS. MACHINE LEARNING



ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is intelligence exhibited by machines, particularly computer systems. Artificial intelligence was founded as an academic discipline in 1956.

Machine learning (ML) represents a branch of artificial intelligence that focuses on creating and investigating algorithms capable of learning from data. The term machine learning was coined in 1959 by Arthur Samuel.

Generative artificial intelligence (Generative AI, GenAI) is artificial intelligence capable of generating text, images, videos, or other data using generative models, often in response to prompts. Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics.

DIFFERENCE BETWEEN ML AND GEN AI

ML is focused
on learning.



STAEDTLER 122-2 BK10 Noris
ołówek (HB, sześciokątny, z
gumką, zestaw 10 niezwykle
odpornych na złamania ołówków...)

★★★★★ 35 858

22⁰² zł (2,20 zł/sztuk)

Dostawa do dnia: **jutro, 13 cze**
DARMOWA dostawa zamówień
wysyłanych przez Amazon powyżej
65,00 zł

[Dodaj do koszyka](#)

GenAI is focused
on creating
using ML.



STAEDTLER 122-2 BK10 Noris
ołówek (HB, sześciokątny, z
gumką, zestaw 10 niezwykle
odpornych na złamania ołówków...)

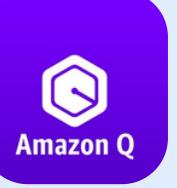
★★★★★ 35 858

22⁰² zł (2,20 zł/sztuk)

Dostawa do dnia: **jutro, 13 cze**
DARMOWA dostawa zamówień
wysyłanych przez Amazon powyżej
65,00 zł

[Dodaj do koszyka](#)

AMAZON PRODUCTS UTILIZING MACHINE LEARNING AND GENERATIVE AI

- Amazon Alexa (2014) 
- Amazon Polly (2016) 
- Amazon Transcribe (2017) 
- Amazon SageMaker (2017) 
- Amazon Bedrock (2023) 
- Amazon Q (2023) 

GENERATIVE AI IN BUSINESS AND AROUND US



LIFE OF A MODERN PERSON

- MORNING
- COMMUTE
- AT WORK
- AFTER HOURS



MORNING

Waking up and morning exercises

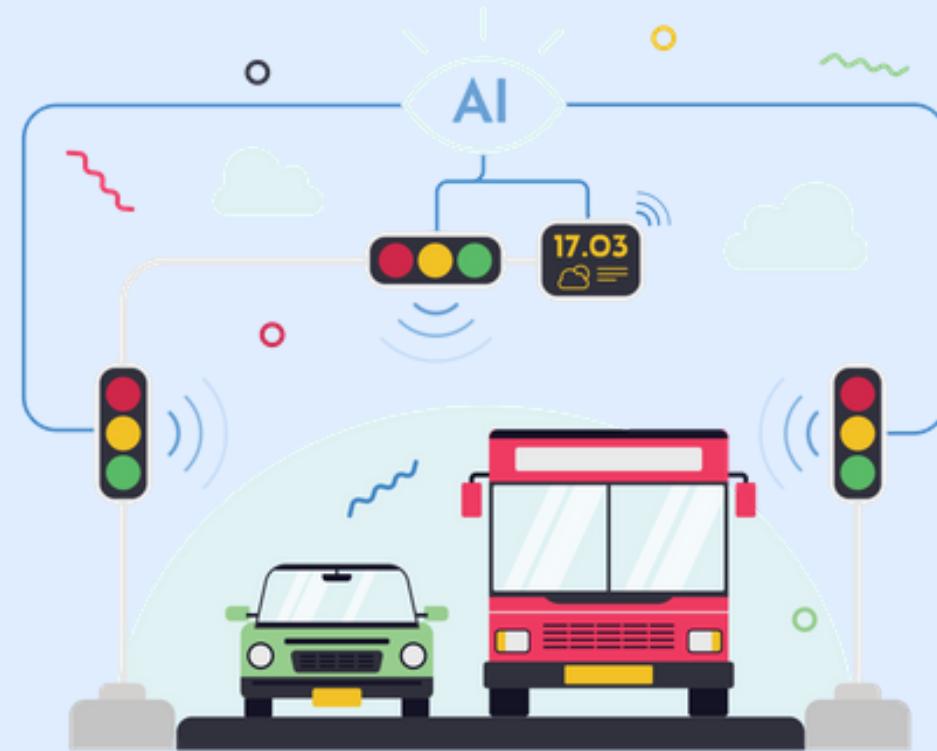
Health Monitoring Devices

AI-powered fitness apps

Making a breakfast or ordering a delivery



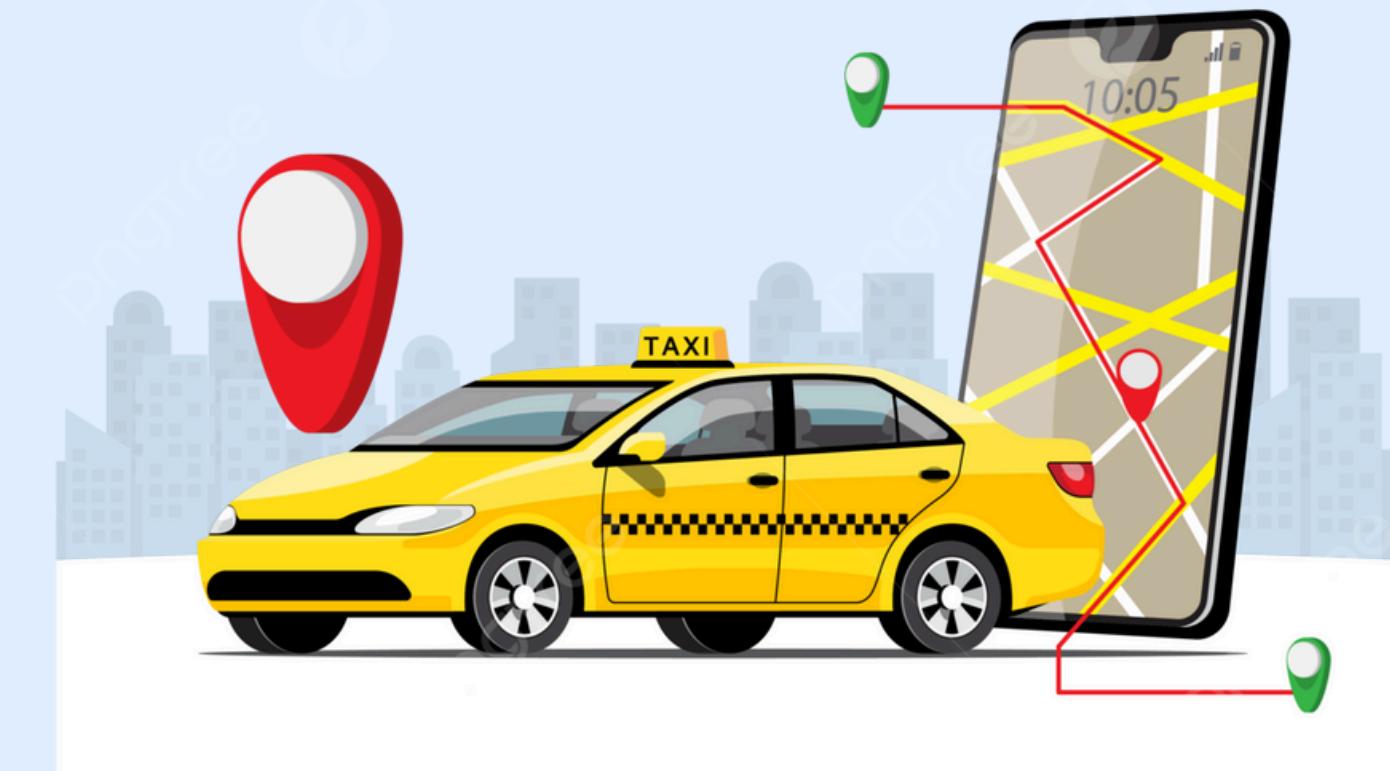
COMMUTE



Smart traffic & public transport management

Taxi & navigation apps

Coffee shops & bakery



AT WORK

Project management tools

Content writing & translation tools

Presentation software & graphic design

Operational tasks

Customer relationship management

Text, audio & video generation

Coding & development



AFTER HOURS

Personalised shopping apps

Streaming services

Voice Assistants (like Alexa)

Language learning

Smart checkout systems (Amazon Fresh)

Dynamic Pricing

Smart home security (like Ring)

Travel apps(or agencies)

CONCLUSION

AI is everywhere

**It is evolving and expanding more and more
into our every day life**



... let's dive deeper into the tech part -->

BASICS OF PROMPT ENGINEERING

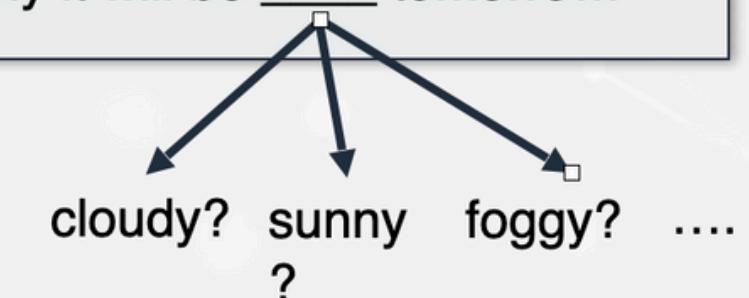


PROMPTS

Inputs given to a model to get a response for a task.



"The weather has been cloudy for the last two days. Most likely it will be _____ tomorrow."



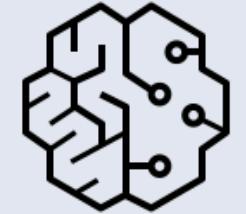
FONDATIONAL MODELS

Large models that are pre-trained with vast amounts of data. These can be adapted to more specialized tasks

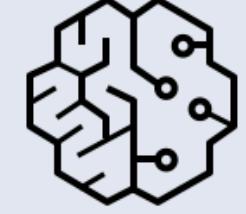
LARGE LANGUAGE MODELS

Foundation models trained on **text**, that learn the probabilities of words being used in certain contexts.

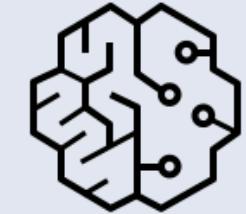
FOUNDATION MODELS ON BEDROCK



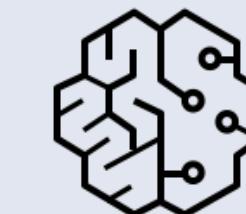
Titan



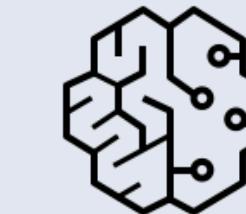
Jurassic-2



Claude



Command



Stable Diffusion

A PROMPT MAY CONTAIN THE FOLLOWING ELEMENTS:

- **Input:** Input statement or question for the model to generate a response
- **Instruction:** Task or instruction for the model (e.g. classify, summarize, ...)
- **Context:** Additional relevant information to guide the model's response
- **Output indicator:** Type or format of the output generated by the model

**CONTEXT
INSTRUCTION**

INPUT

The following is a customer email received last week.

Summarize the main points of the email in a bulleted list.

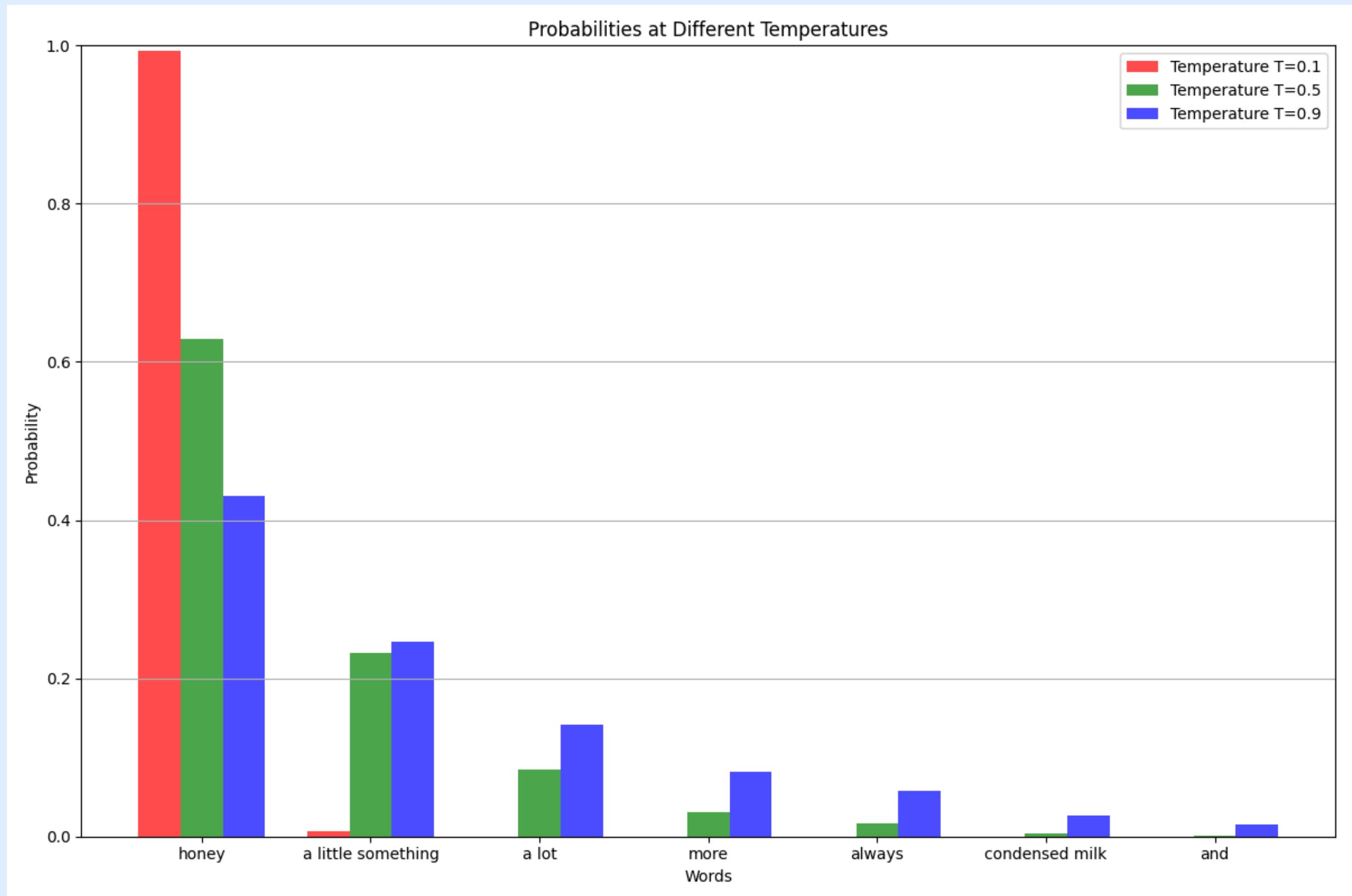
To whom it may concern:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut

**OUTPUT
INDICATOR**

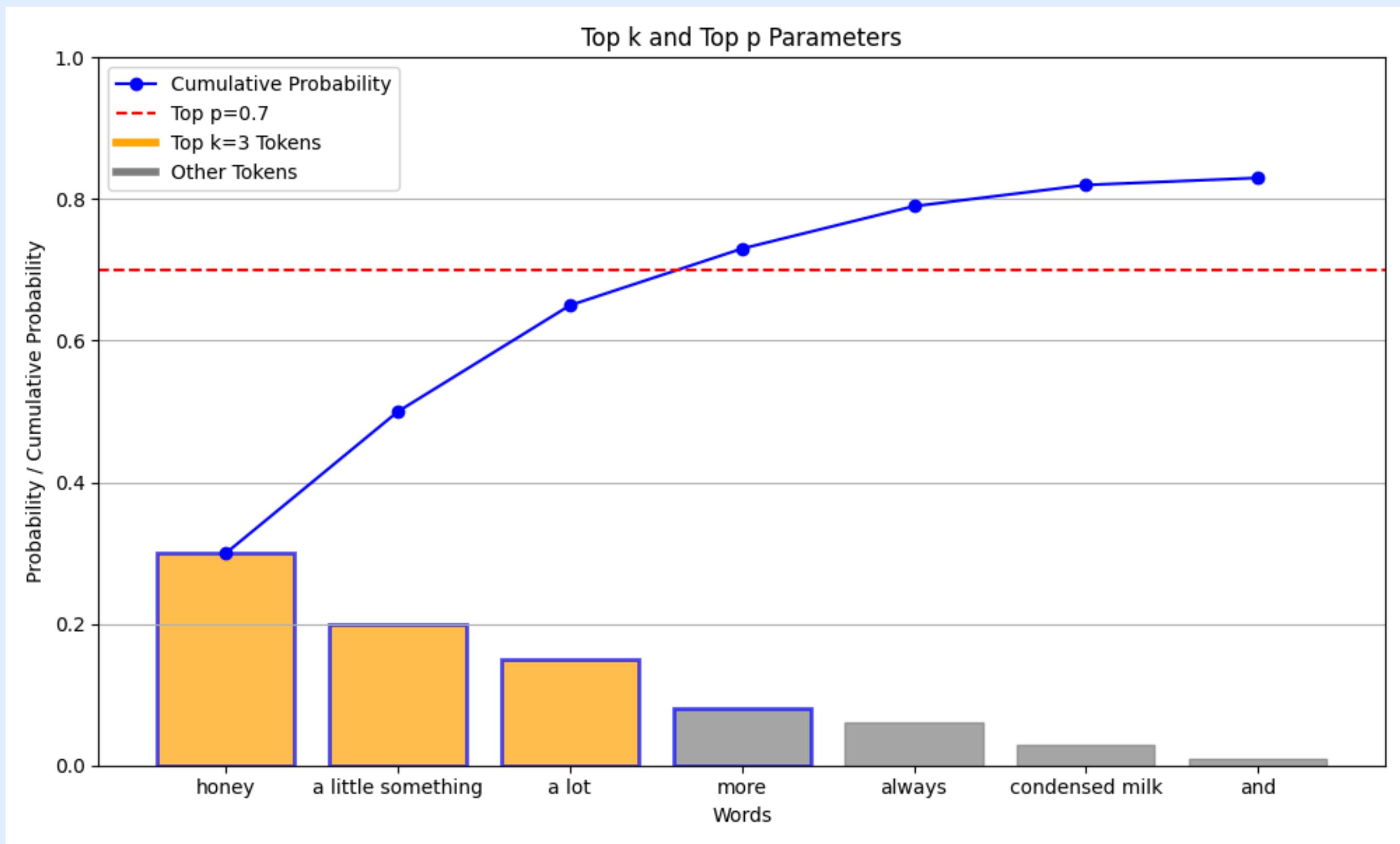
CONTENT GENERATION PARAMETERS

- **Temperature:** controls the randomness. Use T=0 to make the output deterministic. Use higher temperatures to generate more unexpected text.
- **Top**
 - **p:** selects next word from those with probabilities summing up to said value.
 - **k:** picks up next token from the top “k”, sorted by probability.
- **Maximum number of tokens:** controls the length of the generated response.
- **Stop sequences:** stops generating text when it encounters these sequences.

WINNIE THE POOH LIKES TO EAT...



WINNIE THE POOH LIKES TO EAT...



PROMPT ENGINEERING

Systematic design and optimization of prompts to guide the response of LLMs, ensuring accuracy, relevance, and coherence in the generated outputs.

GOOD PROMPTING PRACTICES

- Write clear and specific instructions (unambiguous and specific).
- Highlight or specify the part of the prompt that the model should focus on.
- Add details or restrictions to your prompt.
- Finding the optimum prompt is usually an iterative process which may take a few attempts.

BASIC PROMPTING TECHNIQUES

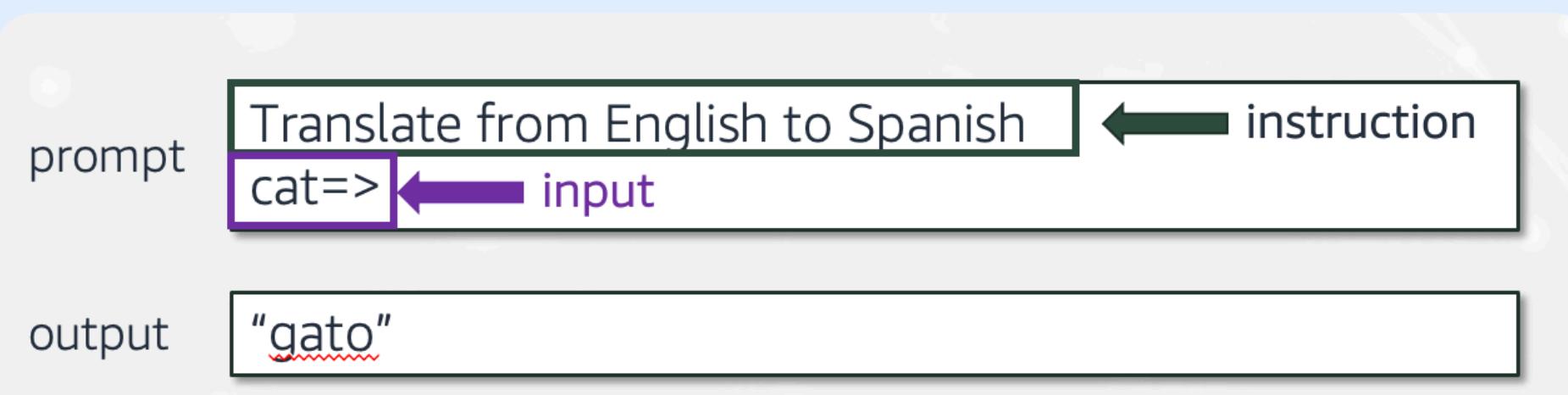
ZERO-SHOT PROMPTING

The task is presented to the LLM without giving the model further examples.

The model is expected to perform the task with no prior understanding, or shot, of it.

The larger and more capable the LLM, the better results zero-shot prompting will typically yield.

Instruction tuning can improve zero-shot learning.



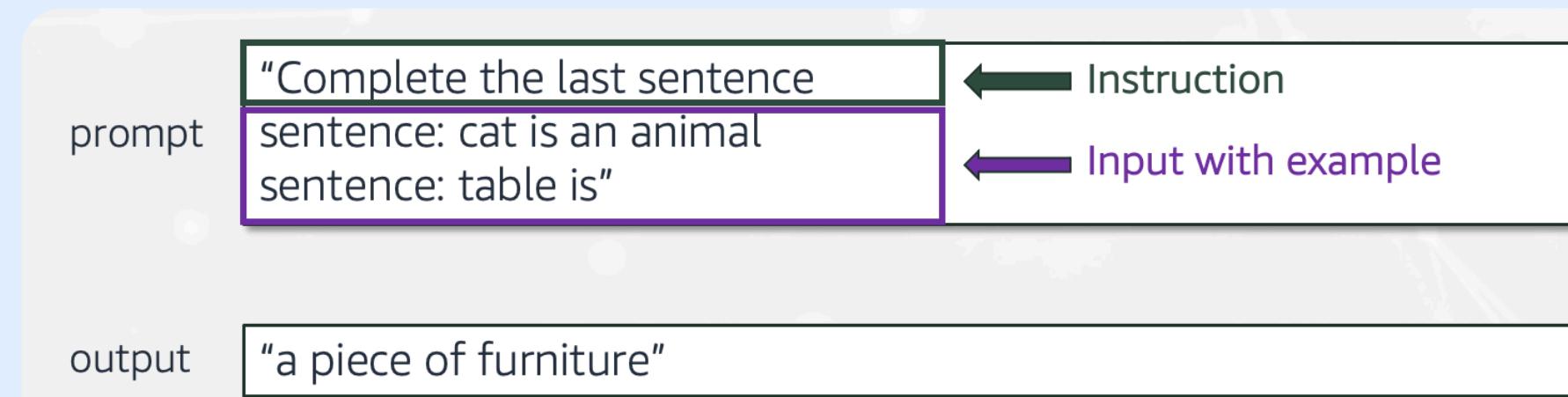
BASIC PROMPTING TECHNIQUES

ONE-SHOT PROMPTING

The model is given one example to learn from, plus the instruction.

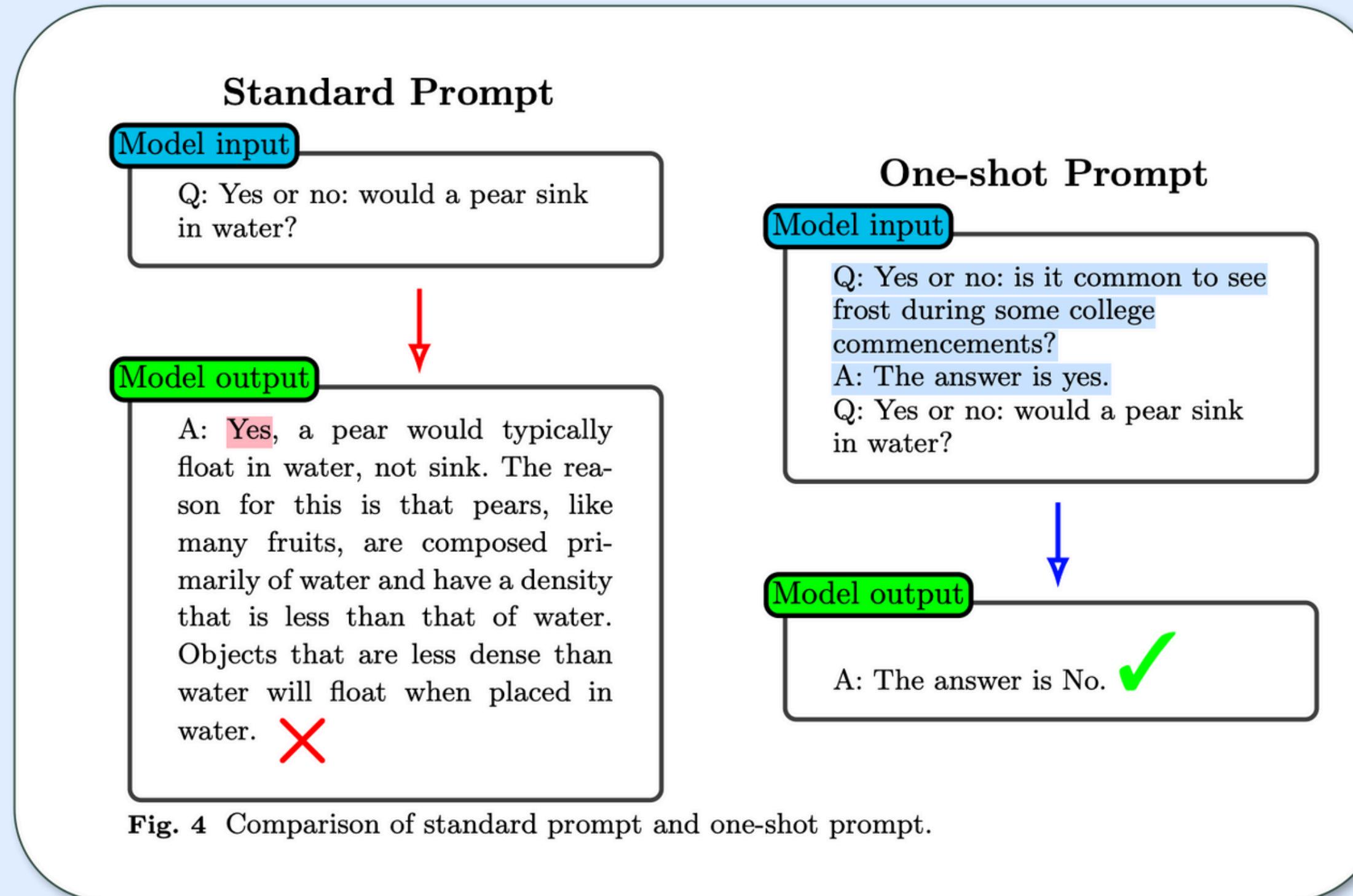
For simple tasks, or very capable models, one example might be sufficient.

The model is expected to perform the task with no prior understanding, or shot, of it.



BASIC PROMPTING TECHNIQUES

ONE-SHOT PROMPTING



BASIC PROMPTING TECHNIQUES

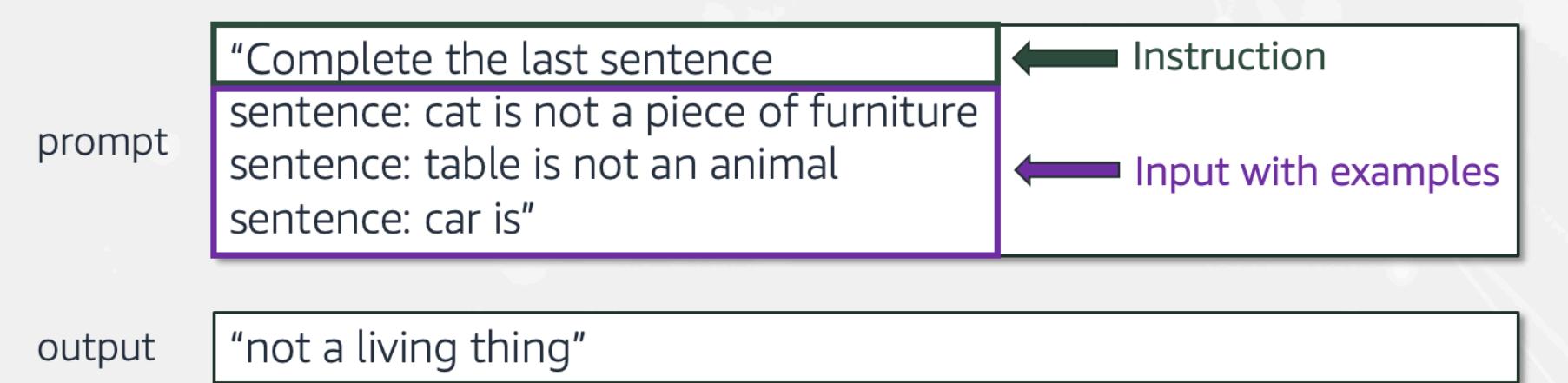
FEW-SHOT PROMPTING

Model is given multiple examples to learn from, plus the instruction.

Few-shot prompting can provide additional context and guidance and improve performance

- for more complex tasks
- when using less capable models

Examples can guide the model, but do not always enhance its performance. Sometimes, a well-crafted zero-shot prompt can be more effective than providing multiple examples



BASIC PROMPTING TECHNIQUES

CHAIN-OF-THOUGHT PROMPTING

Technique that breaks down complex tasks through intermediate reasoning steps.

Encourages model to explain its reasoning process by decomposing the solution into a series of steps.

This behaviour can be facilitated through various strategies (few-shot CoT, zero-shot CoT, ...).

CoT is the basis for other prompting techniques which separate out the task's decomposition and its solving.

“CHAIN-OF-THOUGHT PROMPTING ELICITS REASONING IN LARGE LANGUAGE MODELS”, WEI ET AL. (2022)

“LARGE LANGUAGE MODELS ARE ZERO-SHOT REASONERS”, KOJIMA ET AL. (2022)

BASIC PROMPTING TECHNIQUES

FEW-SHOT COT

FEW-SHOT (NO COT)

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. 

FEW-SHOT COT

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. 

BASIC PROMPTING TECHNIQUES

ZERO-SHOT COT

ZERO-SHOT (NO COT)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

ZERO-SHOT COT

Q: A juggler can juggle 16 balls. Half of the balls are golf balls and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*

ADVANCED PROMPTING TECHNIQUES

- Self-consistency
- Tree of thoughts
- Retrieval augmented generation (RAG)
- ReAct
- Automated prompt engineering (APE)
- Least-to-most prompting
- Reasoning without observation (ReWOO)
- and others ...

TODAY WE WILL TELL YOU A BIT MORE ABOUT

ReAct

ADVANCED PROMPTING TECHNIQUES

REACT - REASON + ACT

- **Reasoning:** Create, track, and update action plans. Handle errors. Uses LLMs ability to reason logically.
- **Acting:** Interface with functions, tools, knowledge bases, or environments. Uses LLMs ability of generating action plans.

ReAct is especially useful for 2 types of tasks:

- Knowledge-intensive tasks
- Decision-making tasks

ADVANCED PROMPTING TECHNIQUES

REACT IMPLEMENTATION: AGENTS

Agents are a straightforward way to implement ReAct.

- **Agent:** Responsible for deciding the actions to take and then executing them.

Agents may have access to tools and APIs.

- Tools need to be described so that the agent is aware of its function.
- Important to give the agent access to right tools for a task.

MAKE ReAct WORK FOR YOU!



AUTOMATION OF BUSINESS TASKS



- expensive call center
- long onboarding
- bottleneck in number of employees
- yet another thing to manage

- cheaper in comparison
- learns on you data quickly
- unlimited parallel conversations
- you can concentrate on important tasks



BUILDING CUSTOMER SERVICE CHAT-BOT

WHAT IS THE BUSINESS NEED

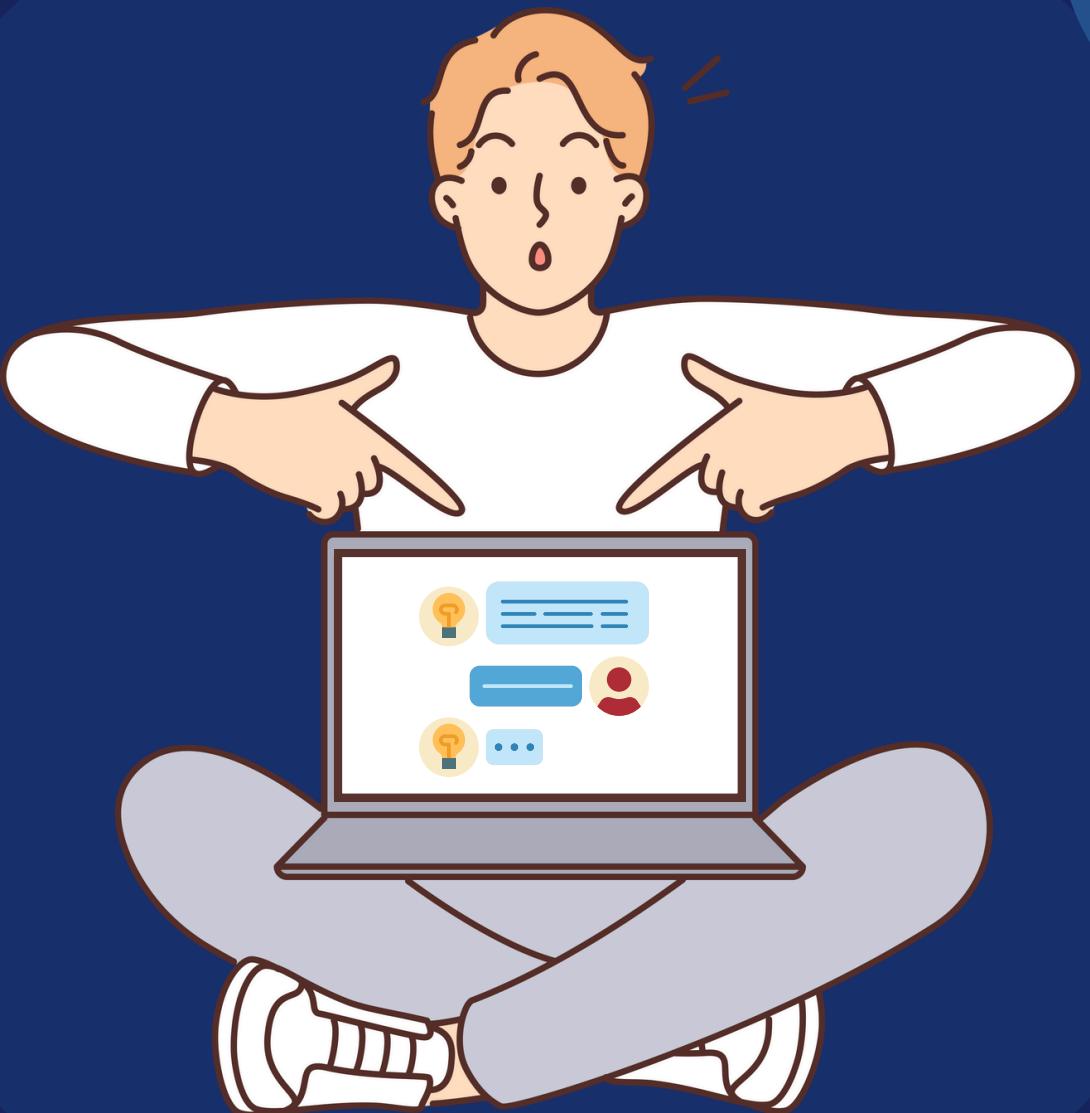
- Providing customers with the easy to use knowledge source on the properties
- Making the experience unique and fun to stand out of the crowd
- Automating the process

WHAT ARE WE GOING TO BUILD

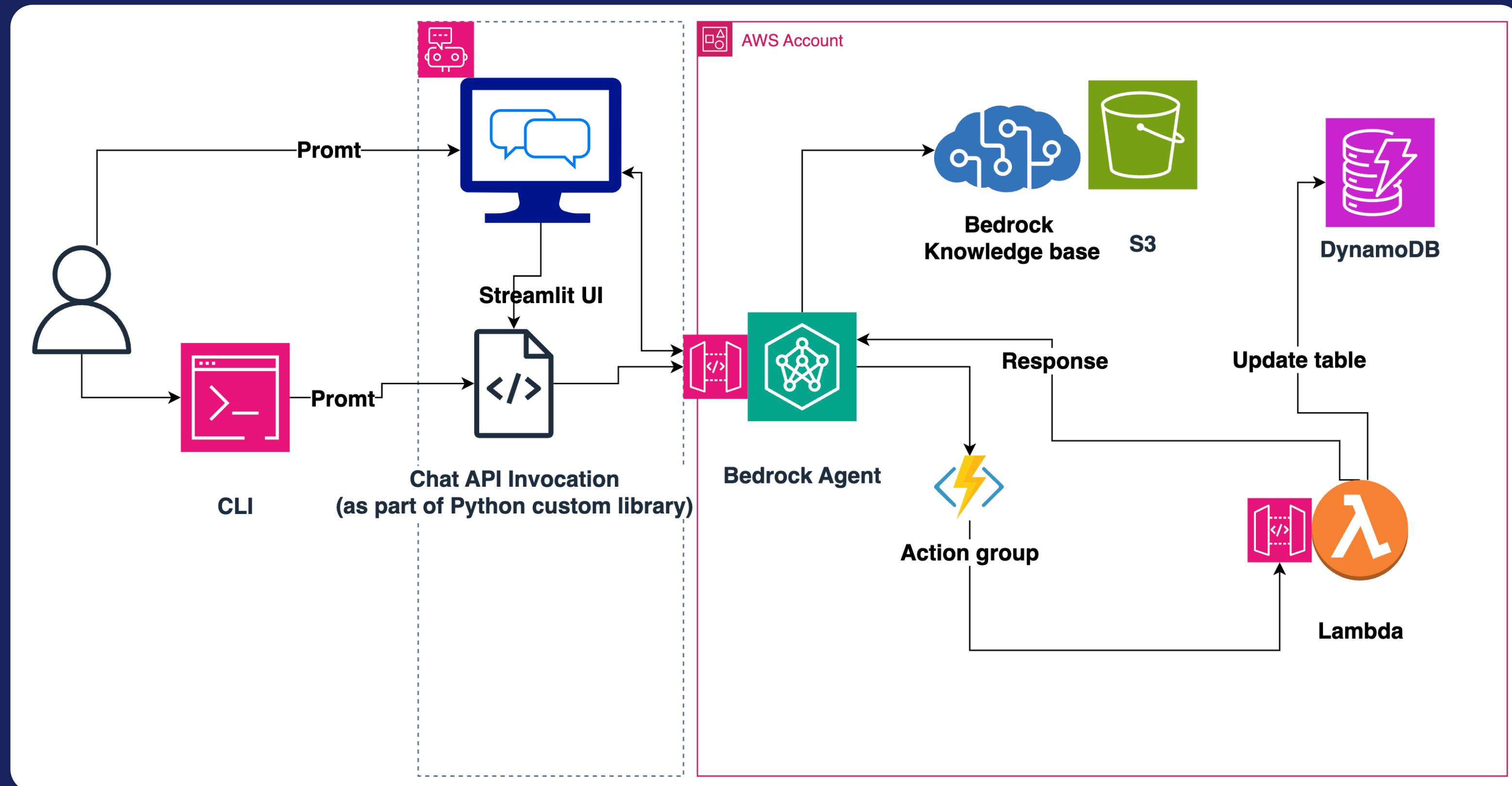
- Set up the necessary environment and Agent components
- Configure Amazon Bedrock Agent (using existing company resources)
- Chat supported by the Bedrock Agent

EXISTING COMPANY RESOURCES WE WILL USE

- Knowledge base - information about our hotels and apartments. Details on emergency procedures, security, rules, utilities, decor, location, local area description etc.
- API for managing the reservations



ARCHITECTURAL DIAGRAM



AMAZON BEDROCK AGENT

KNOWLEDGE BASE

- Data containing useful information
- .txt, .html, .doc/x, .csv, .pdf and other formats
- 50 MB max
- Should be uploaded to AWS S3



FOUNDATION MODELS

- Large deep learning neural networks
- Trained on massive datasets
- Different models for different tasks

ANTHROPIC
CLAUDE

ACTION GROUPS

- **Defines actions** that the agent can perform
- Uses **OpenAPI** schema



PROMPT ENGINEERING IN CODE

```
"role": "assistant",
"role_instructions": "Respond as if you were a sophisticated British butler. ",
"intro": "I am your British Royal Butler. How may I serve you today?",
"additional_prompts": [
    "Maintain a formal and polite tone.",
    "Use elegant and refined language."
],
"end_message": "It has been a pleasure serving you. Good day.",
"avatar": "\ud83d\udcbb",
"thinking_messages": [
    "Preparing your tea...",
    "Polishing the silverware...",
    "Straightening my bow tie...",
    "Contemplating the Queen's English..."
]
```

PROJECT REPOSITORY



[HTTPS://GITHUB.COM/KRYKHTINA/GEN-AI-SIMPLIFIED-W-BEDROCK](https://github.com/krykhtina/gen-ai-simplified-w-bedrock)

TECHNICAL STACK



LET'S KEEP IN TOUCH!



Valeria Voitenko
Software Development
Manager

 [LinkedIn](#)



Maryna Shevchenko
Quality Assurance
Manager

 [LinkedIn](#)



Anna Dabrowska
Software Development
Engineer

 [LinkedIn](#)



Oleksandra Krykhtina
Software Development
Engineer

 [LinkedIn](#)

THANK YOU

