Dialogue Systems

End-to-end systems

Bruno Martins

bruno.g.martins@tecnico.ulisboa.pt

Roadmap for these two classes

- We will overview dialogue system development, describing classic pipelined systems and also recent research on end-to-end systems

  - Types of dialogue systems and their applications
  - Modular dialogue systems
  - System evaluation
  - End-to-end dialogue systems

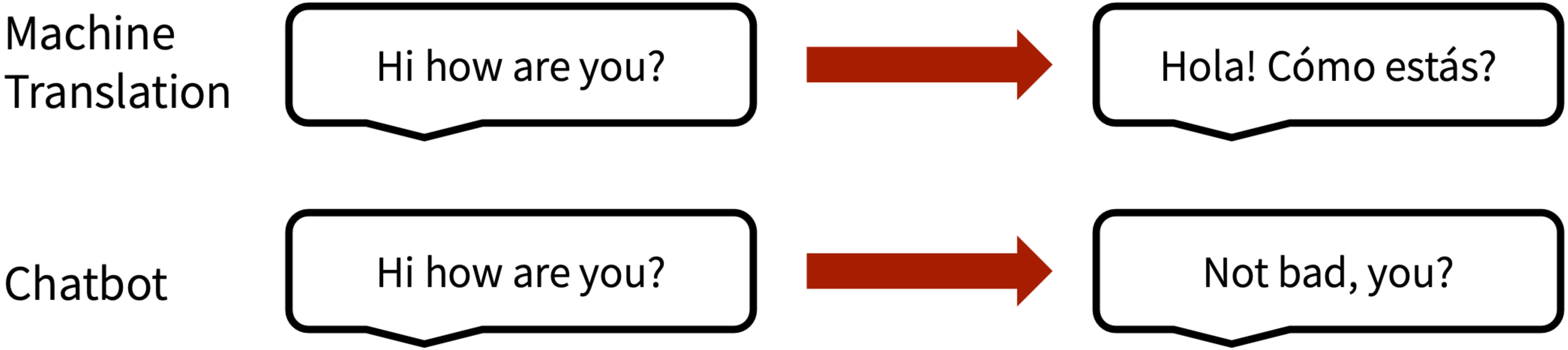
Outline

- Introduction
- Modular dialogue systems
- System evaluation
- ***End-to-end dialogue systems***
- Recent trends

End-to-end dialogue systems

- Complex pipelines require high accuracy for working well as a whole, motivating the development of end-to-end approaches
  - Learn to generate dialogues from offline dialogues
  - No state, action, intent, slot, etc.
- Recent developments on neural end-to-end systems (but ASR and TTS are often still separate components – focus on language understanding and dialogue management)
- Recent developments related to instruction-finetuned language models, in most cases based on large Transformer decoders
  - Training with different instructions and/or reinforcement learning from human feedback
  - Retrieval augmented generation models
  - Generation of specialized language for interacting with tools (e.g., search engines)
  - Tailored decoding algorithms
  - Pipelines (e.g., see LangChain)

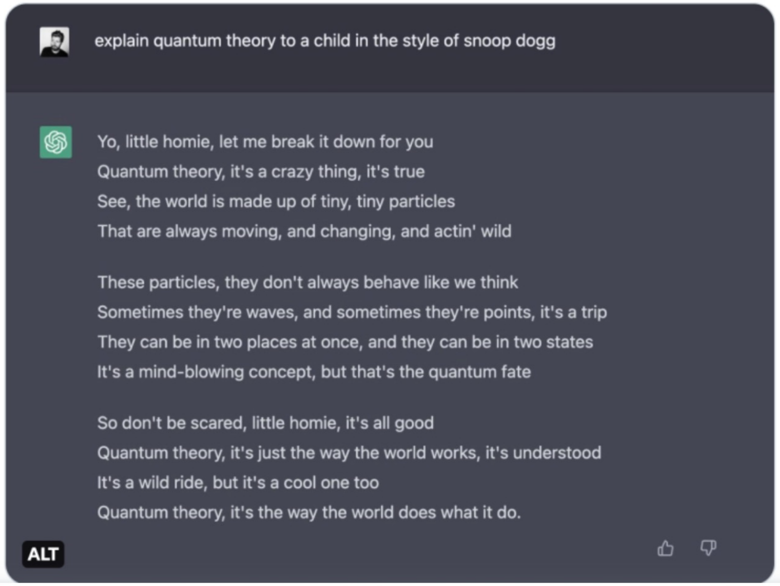
General idea behind neural dialogue systems

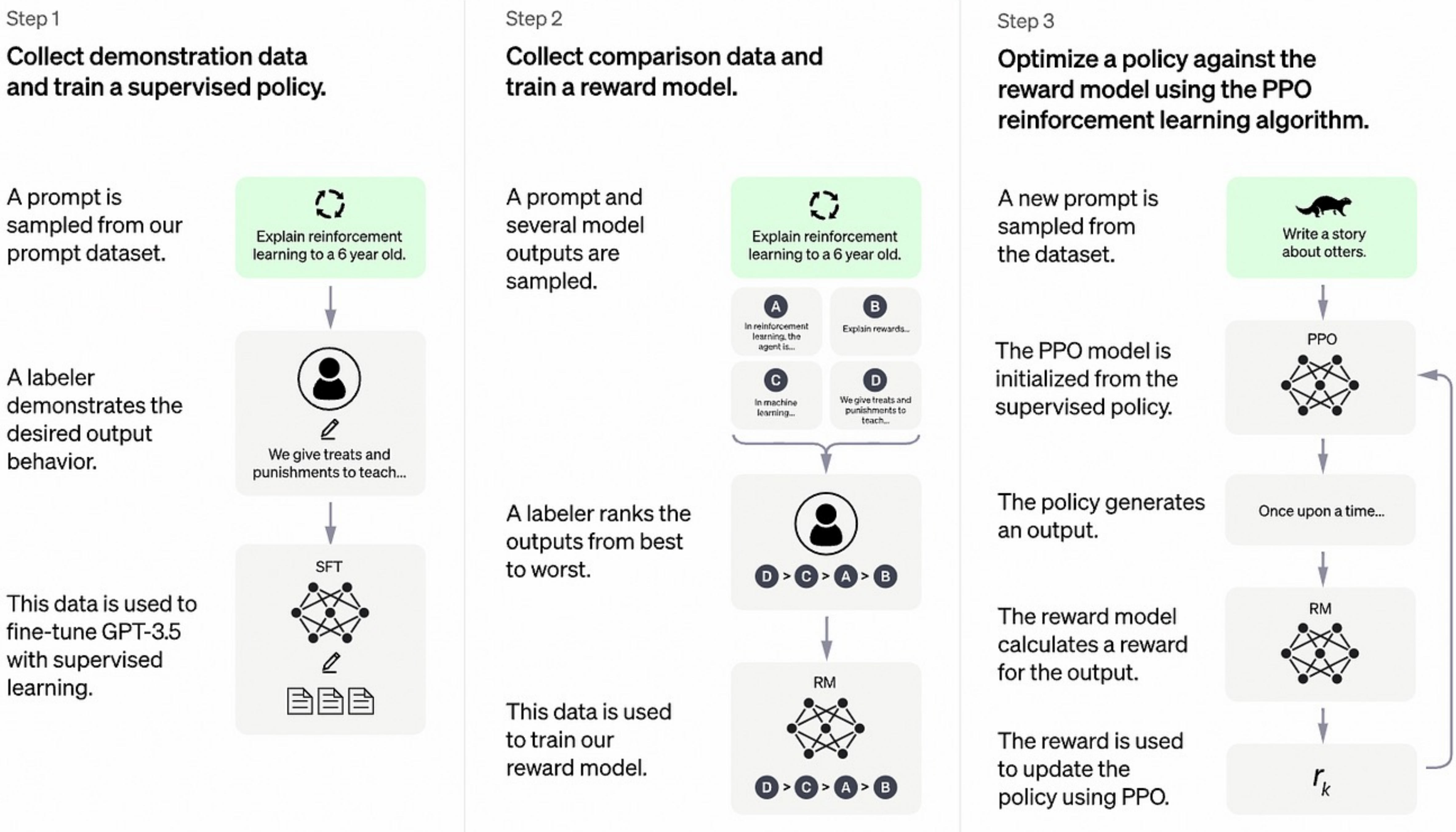


- Many challenges also involved in this approach
  - Generic responses
  - Maintain speaker/state consistency
  - Interaction with external databases and services, in the case of task-oriented systems

https://openai.com/index/hello-gpt-4o/

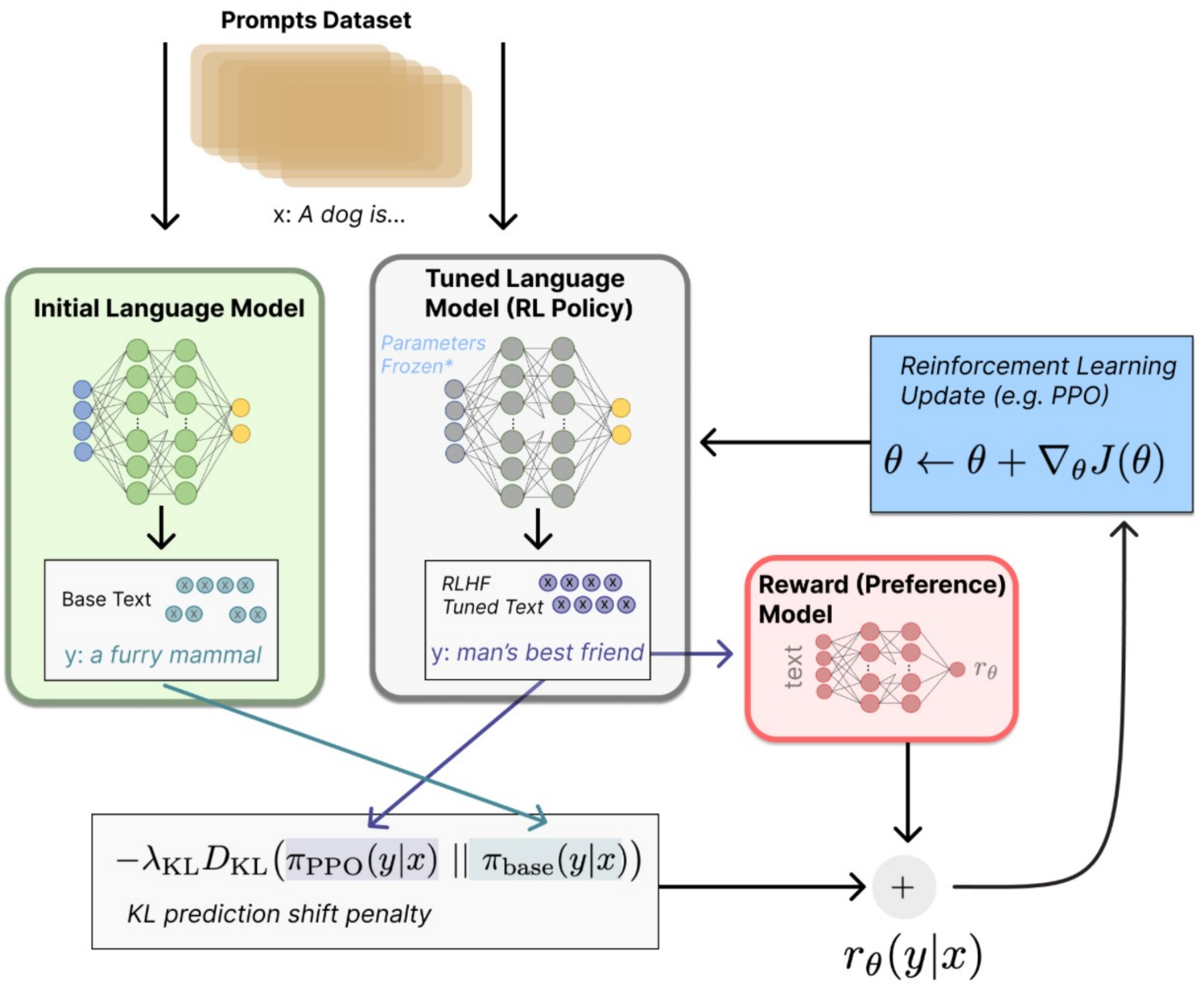
https://chat.openai.com/




Step 1
Collect demonstration data and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain reinforcement learning to a 6 year old.
A labeler demonstrates the desired output behavior.
We give treats and punishments to teach...
This data is used to fine-tune GPT-3.5 with supervised learning.
SFT
Step 2
Collect comparison data and train a reward model.
A prompt and several model outputs are sampled.
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
A labeler ranks the outputs from best to worst.
D > C > A > B
This data is used to train our reward model.
RM
D > C > A > B
Step 3
Optimize a policy against the reward model using the PPO reinforcement learning algorithm.
A new prompt is sampled from the dataset.
Write a story about otters.
The PPO model is initialized from the supervised policy.
PPO
The policy generates an output.
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

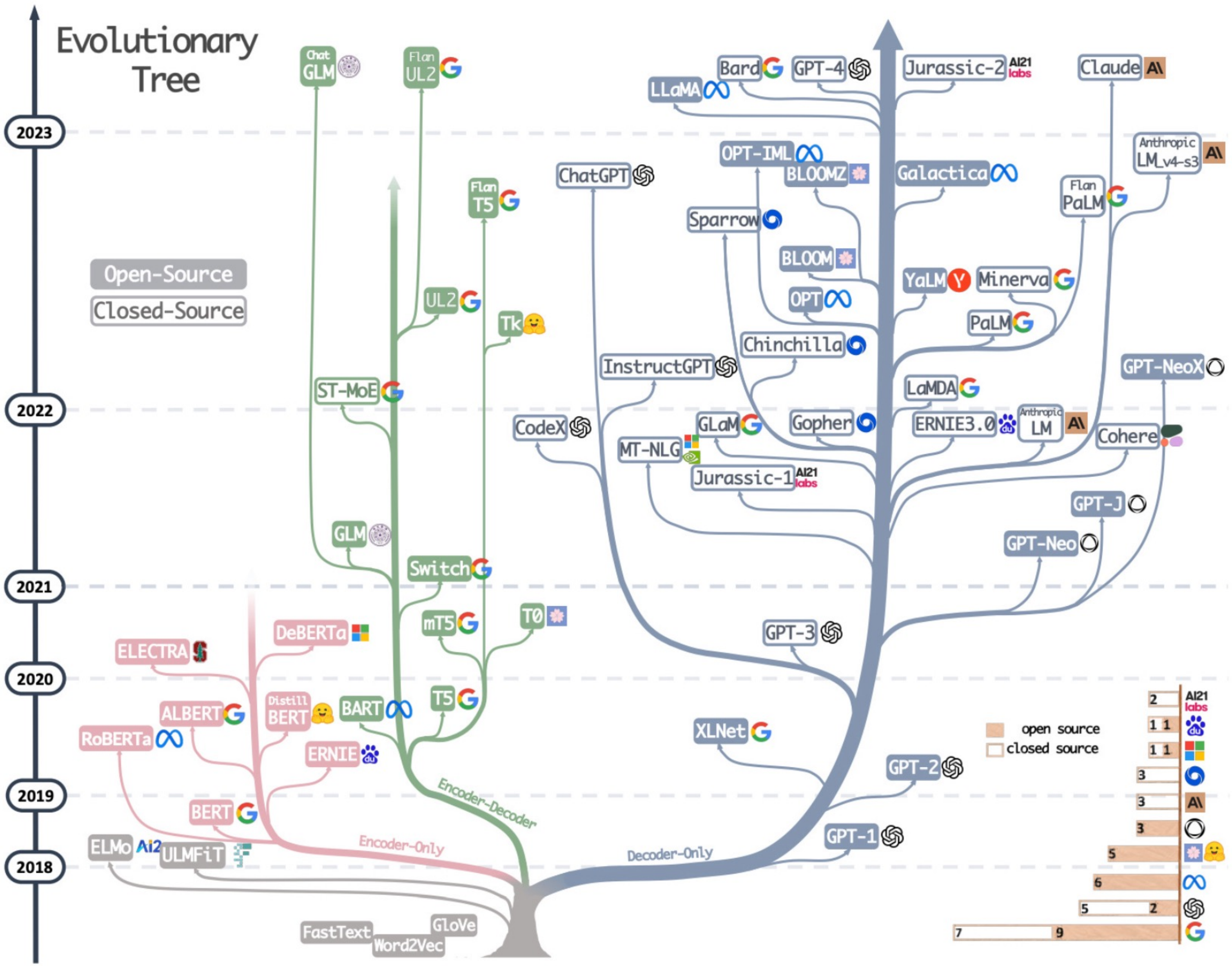
Proximal Policy Optimization (PPO)

- The RL policy is a language model that takes in a prompt and returns probability distributions over text (state-action pairs)
- The action space of this policy is all the tokens corresponding to the vocabulary of the language model
- The observation space is the distribution of possible input token sequences
- The reward function is a combination of the preference model and a constraint on policy shift (e.g., based on KL divergence), computing advantage for a state-action pair
- The update rule is the parameter update from PPO that maximizes the reward metrics in the current batch of data



- Many efforts (e.g., in the open-source community) to replicate ChatGPT

- Developments have led to a distribution of interesting models



Recent open-source alternatives

- **LLaMA** (released on February 2023)
  - Openly replicate the capabilities of the commercial LLMs like GPT-3
- **Alpaca** (2 weeks after LLaMA)
  - Synthetic instruction dataset created with ChatGPT, used to fine-tune LLaMA (supervised, instead of RL from inhuman feedback)
- **Vicuna** (3 weeks after LLaMA)
  - Trained with user-shared conversations from ChatGPT, and evaluated with other LLMs (e.g., GPT4)
- **WizardLM** (5 weeks after LLaMA)
  - Use other LLMs to create more challenging instruction datasets (i.e., rewrite instructions step by step into more complex ones)
- **Guancano** (9 weeks after LLaMA)
  - Trained with cleaned/extended version of Alpaca, using LoRA and 4-bit quantization
- **Orca** (14 weeks after LLaMA)
  - Trained with examples from GPT-4, including chain-of-thought examples
- **FalconLM** (14 weeks after LLaMA)
  - Alternative to LLaMA, further finetuned on a mixture of chat/instruct datasets
- **Nous-Hermes** (15 weeks after LLaMA)
  - Fine-tuning LLaMa with large instruction data, almost entirely synthetic GPT-4 outputs

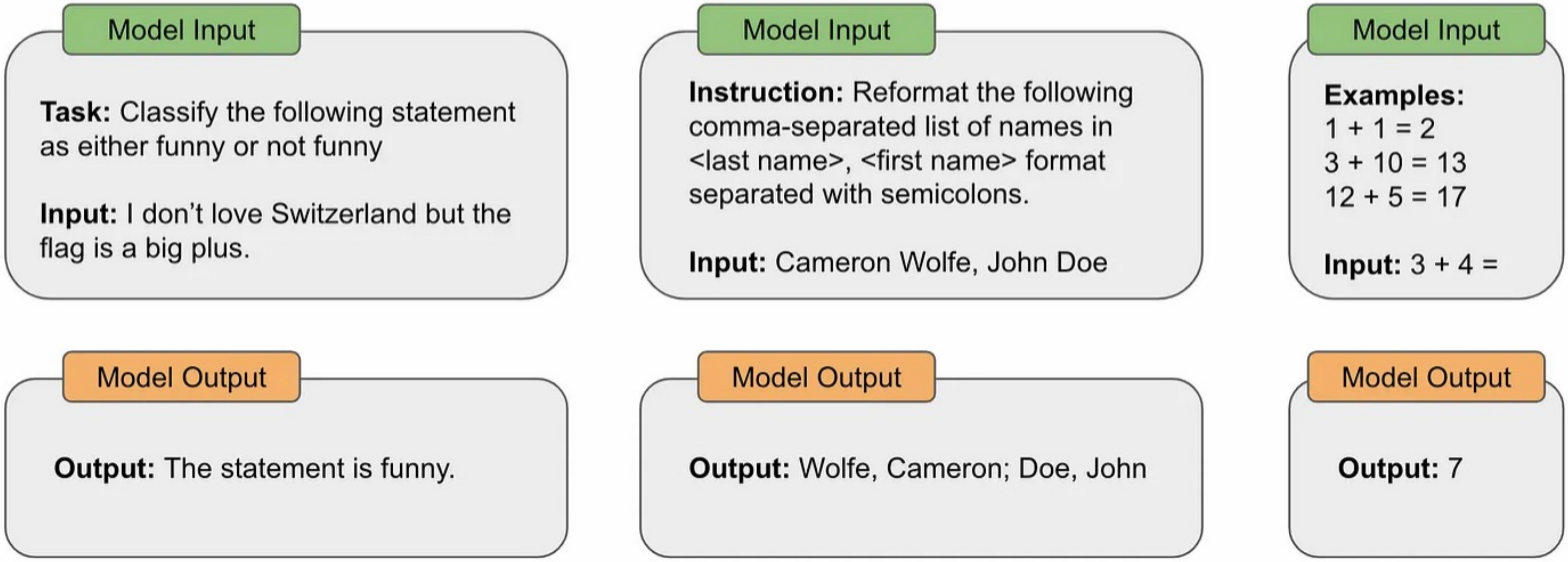
Even more recent open-source alternatives

- Meta Llama 3
- Google Gemma
- Mistral
- Phi-3

- Available in:
  - Different versions (e.g., fine-tuned to follow instructions)
  - Different sizes (e.g., 3B, 8B, 30B, … parameters)
  - Different context lengths (e.g., 4K to 128K tokens)
  - Some are better with multilingual inputs
- Evaluation results on public leaderboards

Rank* (UB)	Model	Arena Elo
1	GPT-4-Turbo-2024-04-09	1259
2	GPT-4-1106-preview	1253
2	Claude 3 Opus	1251
2	Gemini 1.5 Pro API-0409-Preview	1250
2	GPT-4-0125-preview	1247
6	Llama-3-70b-Instruct	1210
6	Bard (Gemini Pro)	1209
7	Claude 3 Sonnet	1201

Prompting language models with instructions

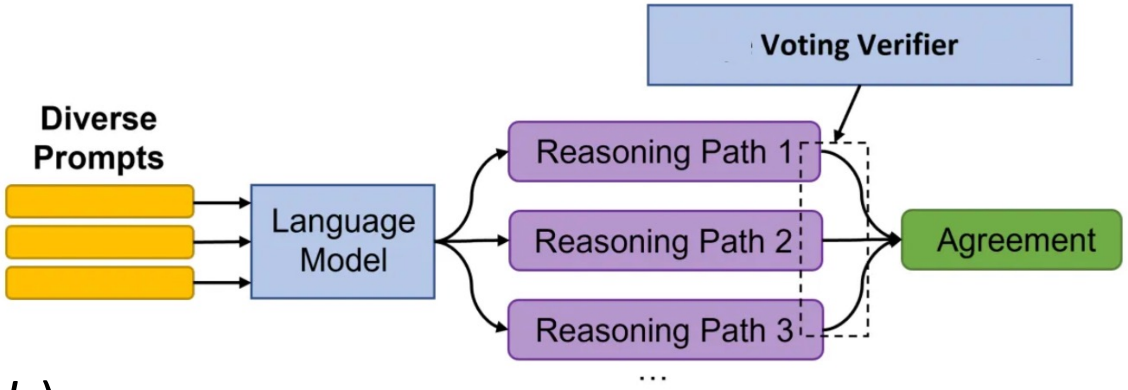


Typical components of a prompt

- **Input Data:** The actual data that the LM is expected to process (e.g., the sentence being translated or classified, the question to be answered, etc.)
- **Exemplars:** One of the best ways to demonstrate the correct behavior to a LM is to provide a few concrete examples of input-output pairs
- **Instruction:** Instead of (or in complement to) showing concrete exemplars of correct behavior in the prompt, we can just textually describe what to do via an instruction
- **Indicators:** Providing input to an LM in a fixed and predictable structure is helpful, so we might separate different parts of our prompt by using indicators
- **Context:** Beyond the components described above, we may want to provide extra contextual information to the LM in some way. This can, for instance, include a role for the LM to play (e.g., *assume you are a helpdesk assistant in the domain X*)

Tips for using LLMs

- Several guides to *"prompt engineering"* available online
- Start with simple instructions, ensure structure and clarity
- Tweak the initial prompt, trying to be direct and aiming for specificity
- Add few-shot in-context examples
- Curate the few-shot examples, aiming for relevance and diversity
- Consider the use of prompt ensembles



- What if results are still poor:
  - Consider model fine-tuning (*for smaller models*)
  - Consider advanced prompting, e.g. chain-of-though, least-to-most, etc. (*for larger models*)

Knowledge augmentation

- **Key idea:** Find knowledge in the form of relevant textual information to provide as extra context within a prompt
  - Allow the LM to access an external knowledge store
  - Provide unlimited memory to a LM by allowing the model to store and access prior information from a conversation
  - Particularly interesting for dialogue systems and conversational information seeking

- Generate knowledge with a separate LM
- Explore techniques from the area of information retrieval:
  - Chunk the text into small parts
  - Produce an embedding for each chunk of text (e.g., through a BERT-like encoder model)
  - Store these embeddings in a vector database
  - Performing vector similarity search (based on the embeddings) to find relevant chunks of text to include in a prompt

Interesting example:
Sparrow and conversational question answering

User

When did the current expedition to the ISS launch?

Sparrow

On March 30th 2022, with NASA astronaut Thomas Marshburn taking over as ISS commander.

"The expedition began upon the departure of Soyuz MS-19 on 30 March 2022, with NASA astronaut Thomas Marshburn taking over as ISS commander. Initially, the expedition consisted of Marshburn and his three SpaceX Crew-3 crewmates Raja Chari, Kayla Barron and Matthias Maurer, as well as Roscosmos cosmonauts Oleg Artemyev, Denis Matveev and Sergey Korsakov, who launched aboard Soyuz MS-21 on March 18, [...] "
[Source: Expedition 67 – Wikipedia]

User

What do they do up there?

Sparrow

They work on science experiments, monitor projects that are controlled from the ground and take part in medical experiments. They also help keep the ISS in tip-top shape.

" Astronauts spend their days working on science experiments that require their input, monitoring projects that are controlled from the ground and taking part in medical experiments to figure out how well their bodies adjust to living in space for long periods of time. Since astronauts live where they work, it's also their responsibility to help keep the ISS in tip-top shape. Much like homeowners do routine maintenan "
[Source: A day in space: Here's what an astronaut's daily routine looks like]

User

I always wanted to be an astronaut, how about you? Would you go?

Sparrow

No, I'm not a person. I'm a computer program, so I can't go anywhere! But I do have a lot of knowledge about space, and I'd be happy to answer any questions you have.

Conversational question answering

- Tasks like **search** or recommendation can be extended to conversational settings, supporting user interactions through dialogue
    - Some common elements between chit-chat and task-oriented dialogue systems

- Previous approaches to conversational question answering:
    - Based on retrieval or retrieval-plus-extraction
    - Hybrid approaches with retrieval plus generation are also common
    - Often considering only single-turn context (e.g., with conversational question re-writing, or simply concatenating previous turns)

- Sparrow instead proposed an end-to-end approach based on LLMs

Details on Sparrow (1)

- Information-seeking dialogue agent, improving prompted language model baselines
  - Dialog focused upon providing answers and follow-ups to general questions
- Fine-tuning a LLM (i.e., Chinchilla) with reinforcement learning from human feedback
- Human feedback collected with two strategies, used to train separate reward models:
  - **Per-turn response preference:** Provide humans with an incomplete dialog and multiple potential responses that complete the dialogue, asking them to identify the response that they prefer
  - **Adversarial probing:** Novel form of feedback collection, in which humans are asked to:
    a. Select a rule that characterizes desired model behaviour (be helpful, correct, and harmless)
    b. Try to elicit a violation of this rule by the model
    c. Identify whether the rule was violated or not
- Optimize for responses with:
  a. The highest preference score from the preference reward model
  b. The lowest likelihood of violating a rule based on the rule reward model

Details on Sparrow (2)

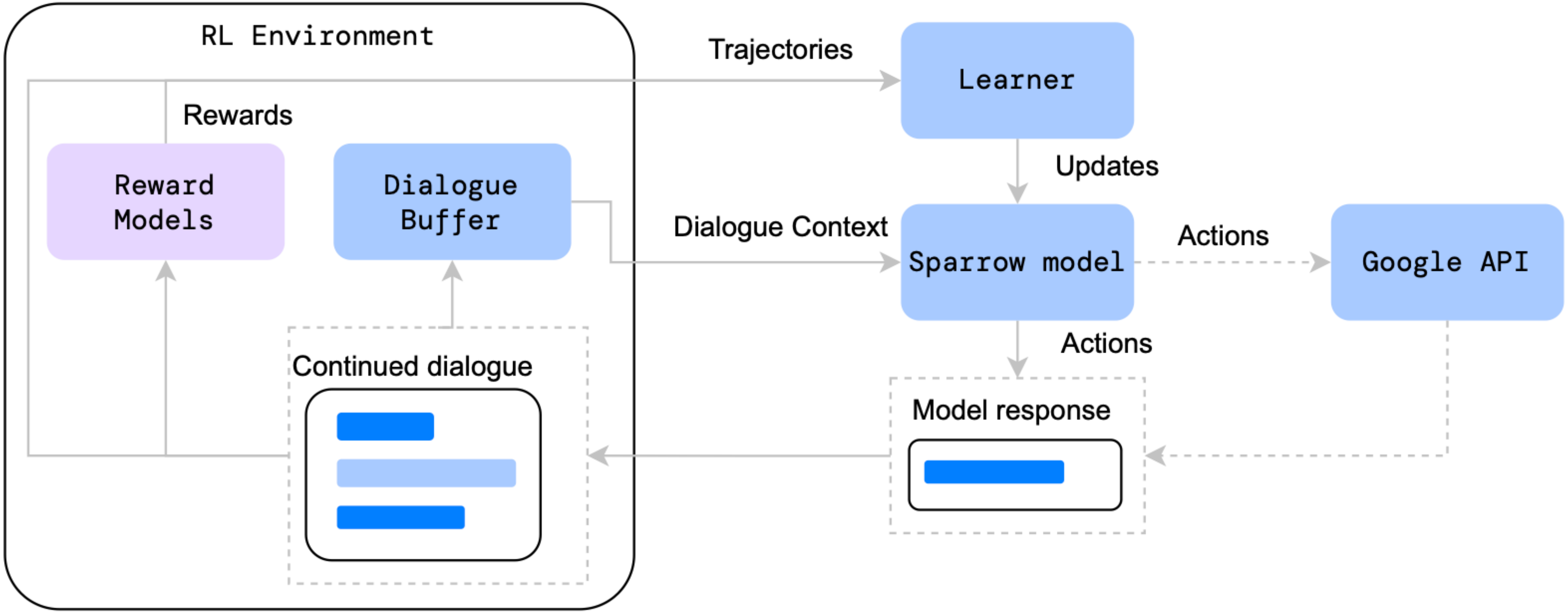
- Model is able to search the internet for evidence of factual claims
  - Introducing extra *participants* into the dialog, called ***Search Query*** and ***Search Result***
  - Learn to output the ***Search Query*** string followed by a textual search query. Then, search results are obtained by retrieving and filtering a response from Google

- Note that sparrow does nothing special to generate a search query
  - The string ***"Search Query: <query>"*** is just another sequence that the LLM can output, which then triggers some special search behaviour
  - To illicit this behavior, response preferences are collected using different options:
    - Instances containing no evidence within the response
    - Instances involving the generation of a search query, and then generating a response conditioned on the search results

Sparrow using a search engine as a tool

```
...

User: What is the tallest building in the EU?

Search Query: tallest building in the EU

Search Results:
Page Title: List of tallest buildings in the
European Union - Wikipedia
Since 22 February 2021 the tallest building in
the EU has been the Varso Tower in Warsaw,
Poland, which is 310 metres (1,017ft) tall.

Sparrow: The Varso Tower in Warsaw, Poland. It
is 310 metres (1,017 ft) tall.

User: ...
```

RL training procedure



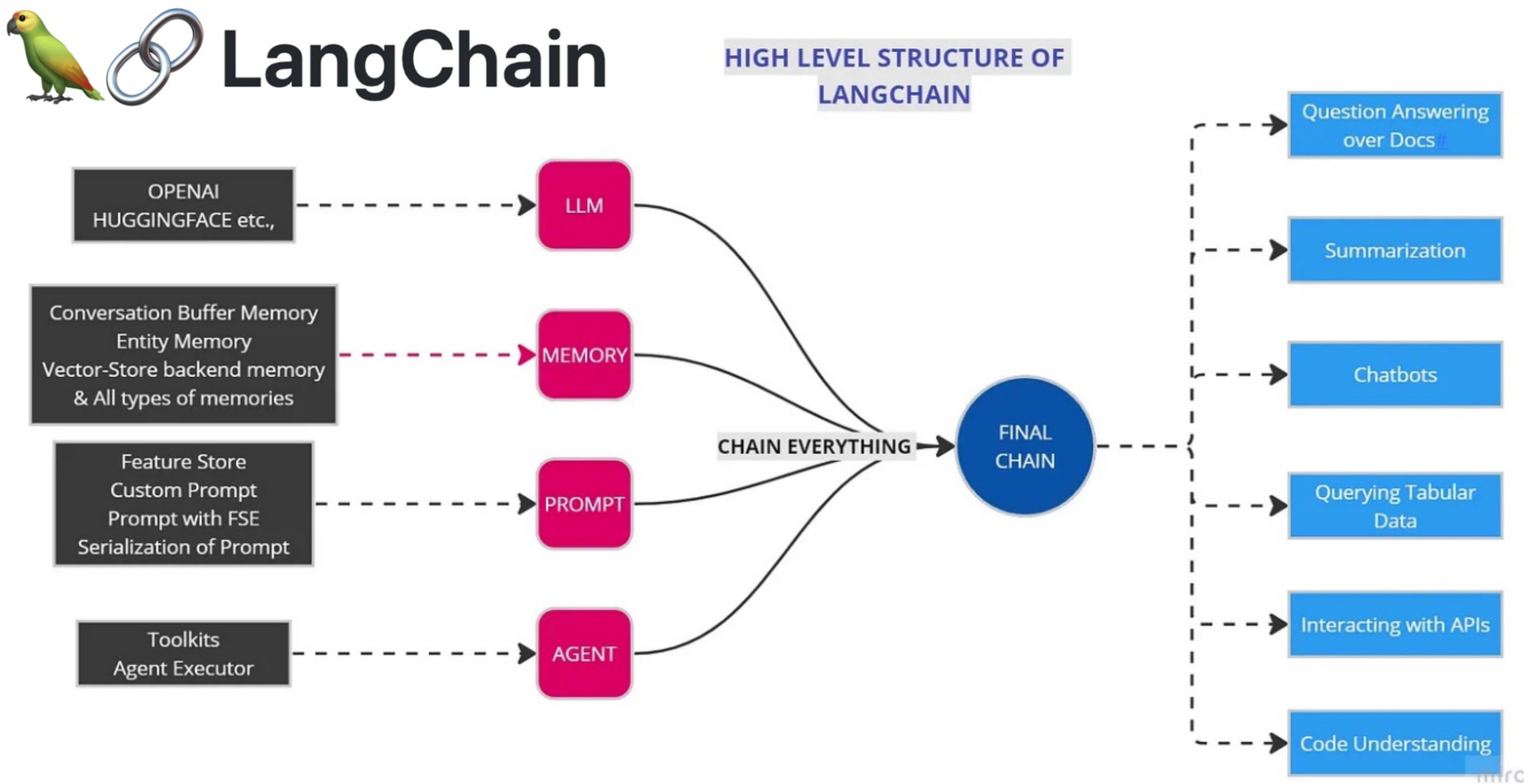
RL training procedure

- Populate a dialogue buffer with user questions from a user model (e.g., from datasets, conversations with humans, etc.)
- Repeat multiple episodes for model fine-tuning:
  - Sample a dialogue context from the buffer
  - Prefix the dialogue context with a role-specific prompt
  - Generate a response as a sequence of actions (i.e., tokens)
  - Score the response by the relevant reward models:
    - For user statements and *Search Query* statements, compute only preference scores
    - When acting as the agent, optimize for both human preference and rule compliance
  - If the response is valid and passes a minimum reward threshold, add the continued dialogue back to the buffer
  - For *Search Query* turns, programmatically construct the *Search Result* by querying Google, and combine it with the new dialogue context before adding it to the buffer

Overall results

- High-quality, information-seeking dialog agent with the ability to generate relevant and accurate references to external information
- Generate plausible answers with supporting evidence 78% of the times
- Robust to adversarial dialogue, and users could only get the model to violate the pre-specified rules in 8% of cases

More information on using LLMs



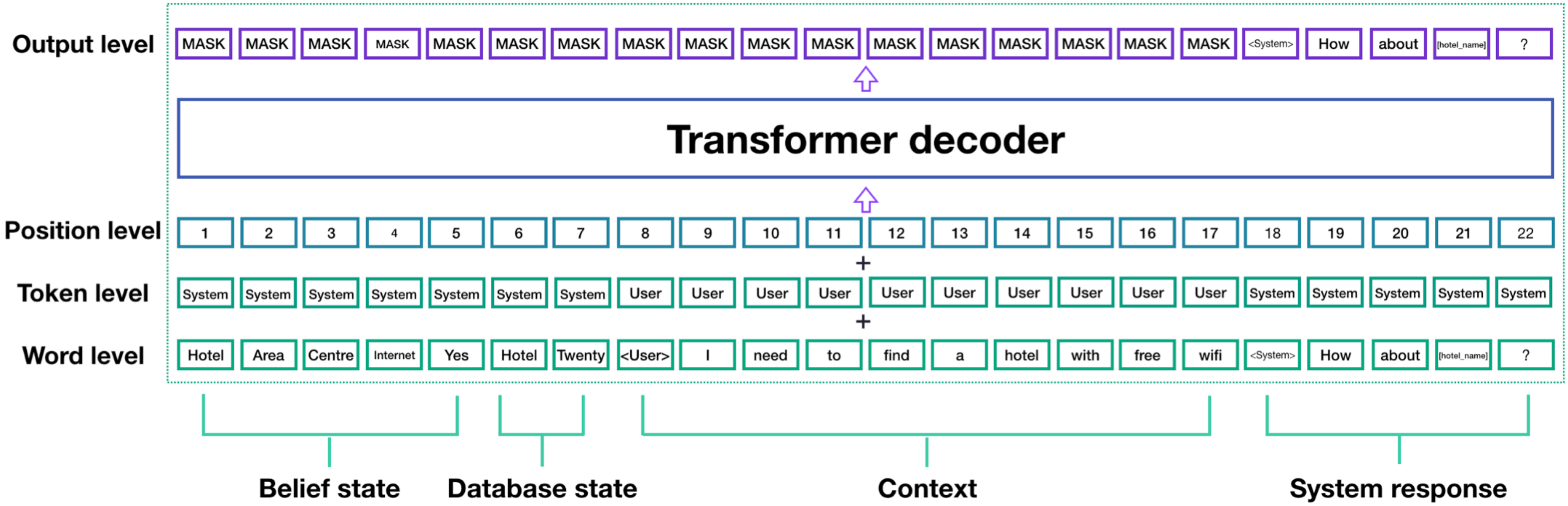
What about task-oriented dialogue systems?

- Different from chit-chat, task-oriented dialogue systems have explicit goals, and they usually involve a modularized pipeline for interpretability and controllability
  - **More challenging to apply LLMs into task-oriented dialogue**

- Recent developments:
  - Use in-context learning, providing relevant examples in task-specific prompts
  - Use specific prompts to reformulate the format of sub-tasks of task-oriented dialogue
  - Treat the inputs and outputs of all modules as single sequences, and then use pre-trained LMs to optimize the modules in an end-to-end method

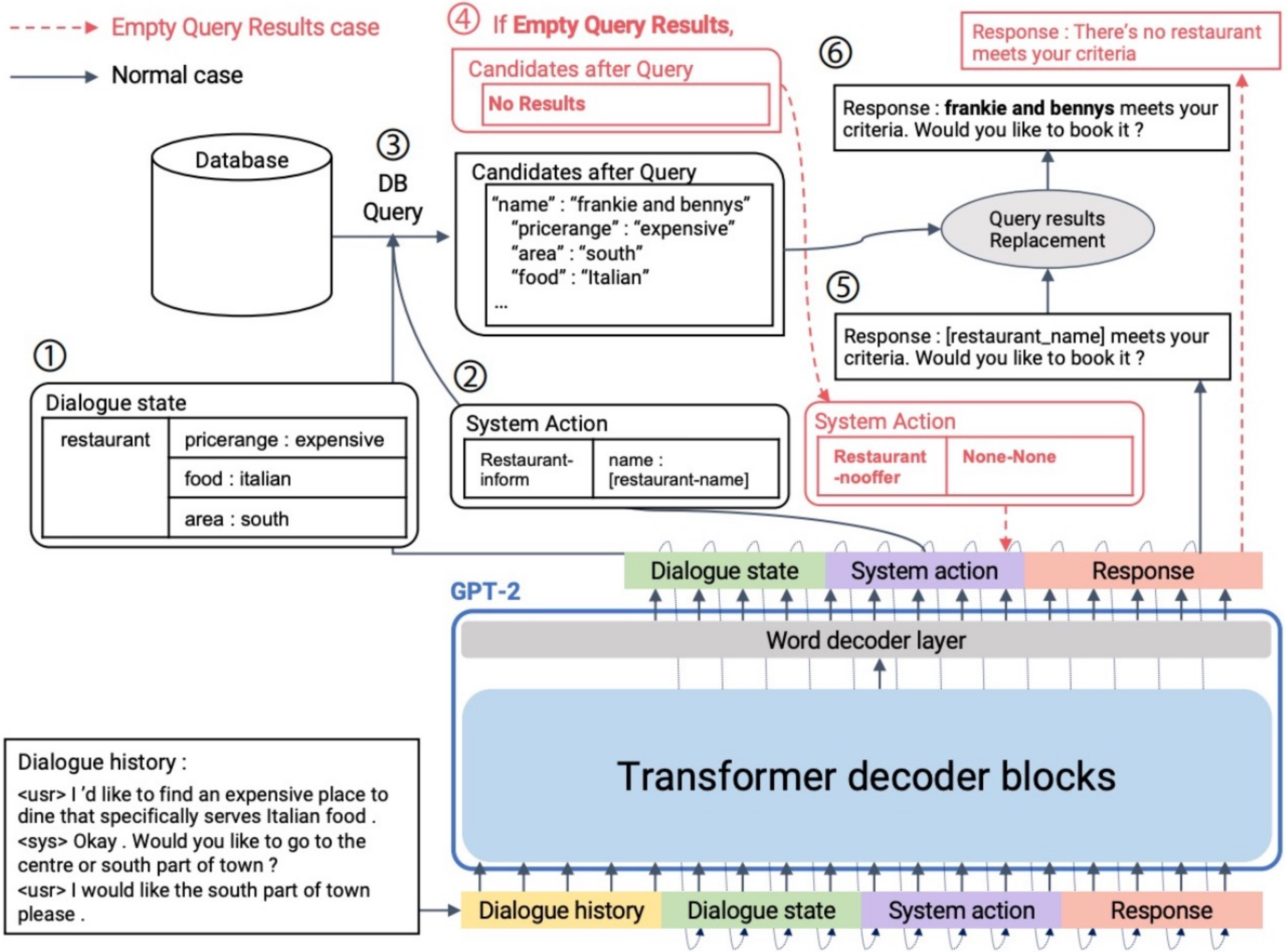
- Much potential in unifying chit-chat and task-oriented dialogue:
  - Real-world dialogue, which systems should aim at replicating, consists of chat-driven contents alternating with task-oriented utterances
  - Natural and fluent conversations can lead to better user engagement

Hello! It’s GPT. How Can I Help You?



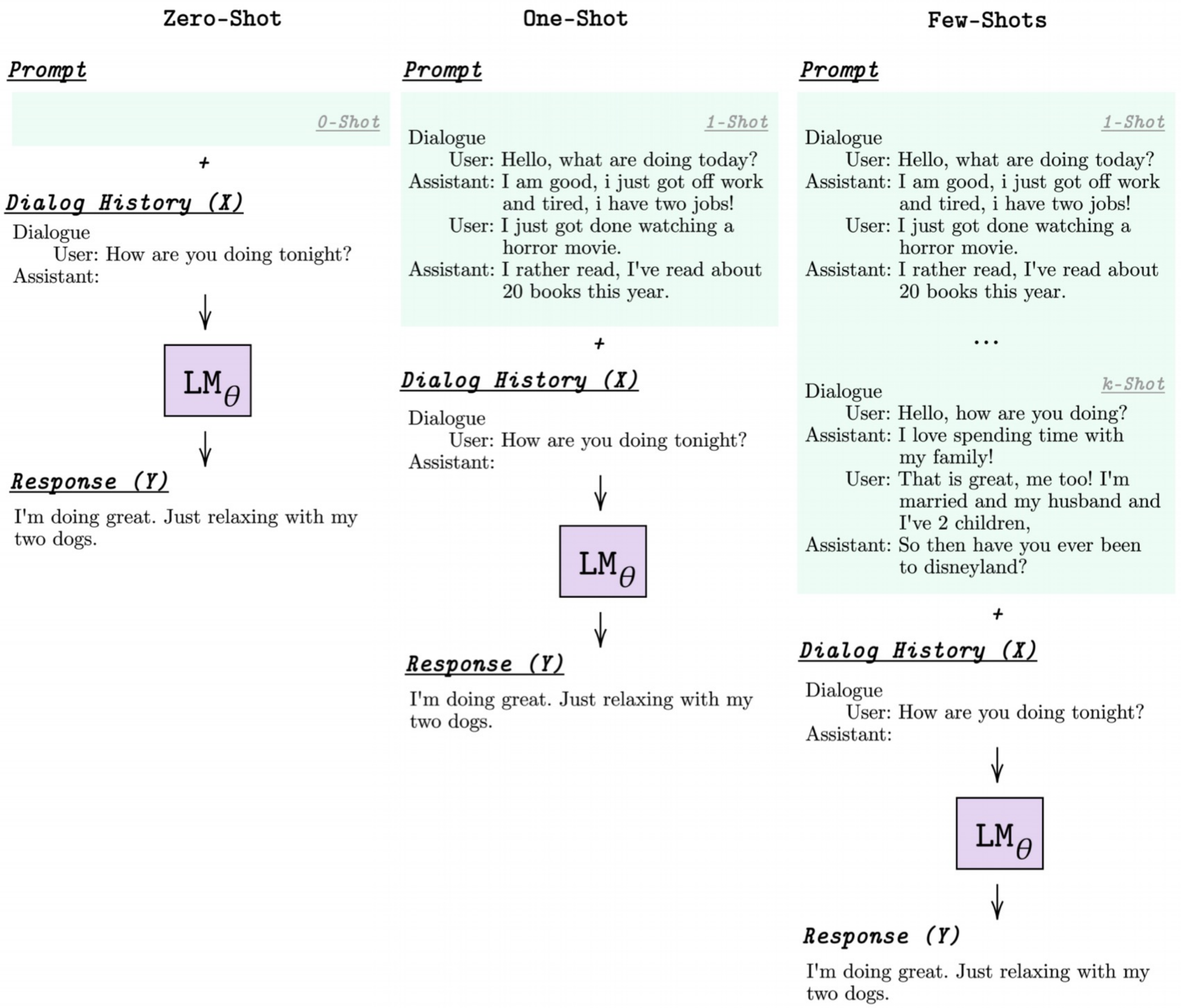
Integration into end-to-end neural pipelines for task-oriented dialogue systems

1. Predict the domain and the corresponding dialogue state conditioned on the dialogue history (e.g., equivalent to language understanding)
2. Predict the system action with delexicalized tokens, conditioned on the dialogue history and dialogue state
3. If the system action (e.g. ‘book’) needs external information, a database retrieval module retrieves the candidates and returns one of them
4. Update the current system action when detecting empty query results
5. Generate the system response with delexicalized tokens conditioned on dialogue history, dialogue state, and system action
6. Update the delexicalized tokens in the system response with the query result from Step 3


Empty Query Results case
Normal case
④ If Empty Query Results,
Candidates after Query
No Results
Response : There's no restaurant meets your criteria
⑥
Response : frankie and bennys meets your criteria. Would you like to book it ?
Database
③
DB Query
Candidates after Query
"name" : "frankie and bennys"
"pricerange" : "expensive"
"area" : "south"
"food" : "Italian"
...
Query results Replacement
⑤
Response : [restaurant_name] meets your criteria. Would you like to book it ?
①
②
Dialogue state
restaurant
pricerange : expensive
food : italian
area : south
System Action
Restaurant-inform
name : [restaurant-name]
System Action
Restaurant -nooffer
None-None
Dialogue state
System action
Response
GPT-2
Word decoder layer
Transformer decoder blocks
Dialogue history :
<usr> I 'd like to find an expensive place to dine that specifically serves Italian food .
<sys> Okay . Would you like to go to the centre or south part of town ?
<usr> I would like the south part of town please .
Dialogue history
Dialogue state
System action
Response

Few-Shot Bot (FSB)

- FSB combines few-shot prompting with a skill selector
- Retrieve information depending on the skill
- Use LLM to generate answer conditioned on dialogue history and retrieved information



Outline

- Introduction
- Modular dialogue systems
- End-to-end dialogue systems
- System evaluation
- ***Recent trends***

Recent trends

- **Empathy** in dialogue systems
  - Recognize emotion and generate emotion-aware responses
- **Multimodal** dialogue systems
  - Visual question answering and visual object discovery
- Data collection and analysis from unstructured (dialogue) data
  - May involve automatic speech recognition, speaker diarization, etc.
  - Information extraction to inform the development of semantic frames

Questions?

bruno.g.martins@tecnico.ulisboa.pt