LLMs for Low Resource Languages in Multilingual, Multimodal and Dialectal Settings

https://llm-low-resource-lang.github.io

QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

EACL 2024, 21th March, 2024

Speakers

Firoj Alam
Scientist

Shammur Chowdhury
Scientist

Sabri Boughorbel
Scientist

Maram Hasanain
Post Doctoral
Researcher

Qatar Computing Research Institute

Content

- Introduction **[20 mins]**
- Models and their capabilities for low-resource languages **[70 mins]**
    - NLP models [40 mins]
    - Multimodality [25 mins]
        - Overview
            - Multimodality
            - Speech
    - QA [5 mins]
- Coffee break **[30 mins]**
- Prompting + Benchmarking Tool **[60 mins]**
    - Prompt Engineering [40 mins]
        - Prompting techniques
        - Cross-/multi-lingual prompting
    - Prompt and Benchmarking tools [15 mins]
    - QA: [5 mins]
- Other Related Aspects [**20 mins**]

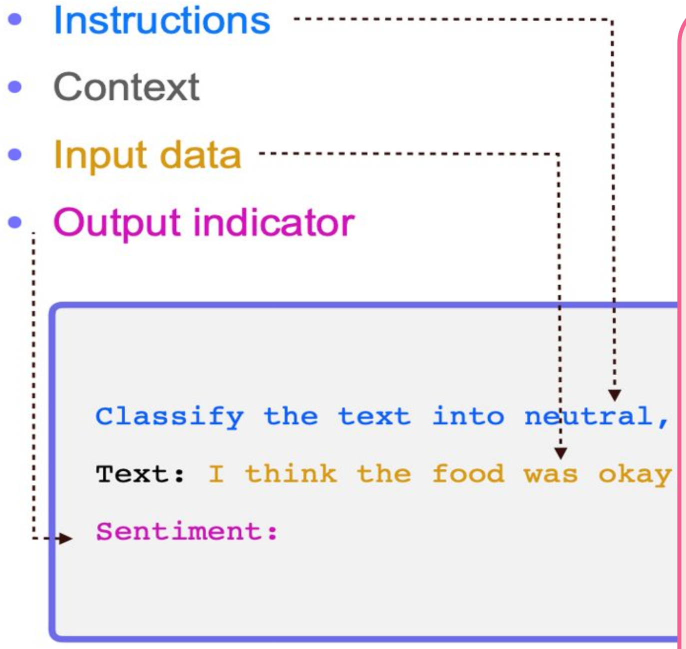
Prompting and Benchmarking Resources

Prompt Engineering

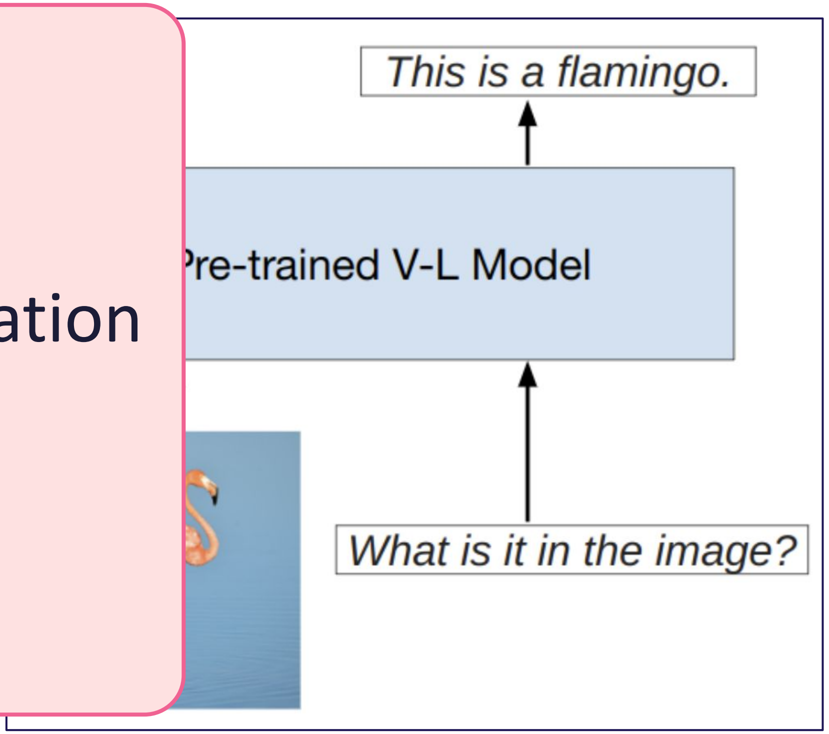
- Prompt Engineering
- Prompting techniques
- Cross-/multi-lingual prompting
- In-Context/Few-shot Learning

What is a “Prompt”?

An instruction given to LLM to guide it on how to perform a user task



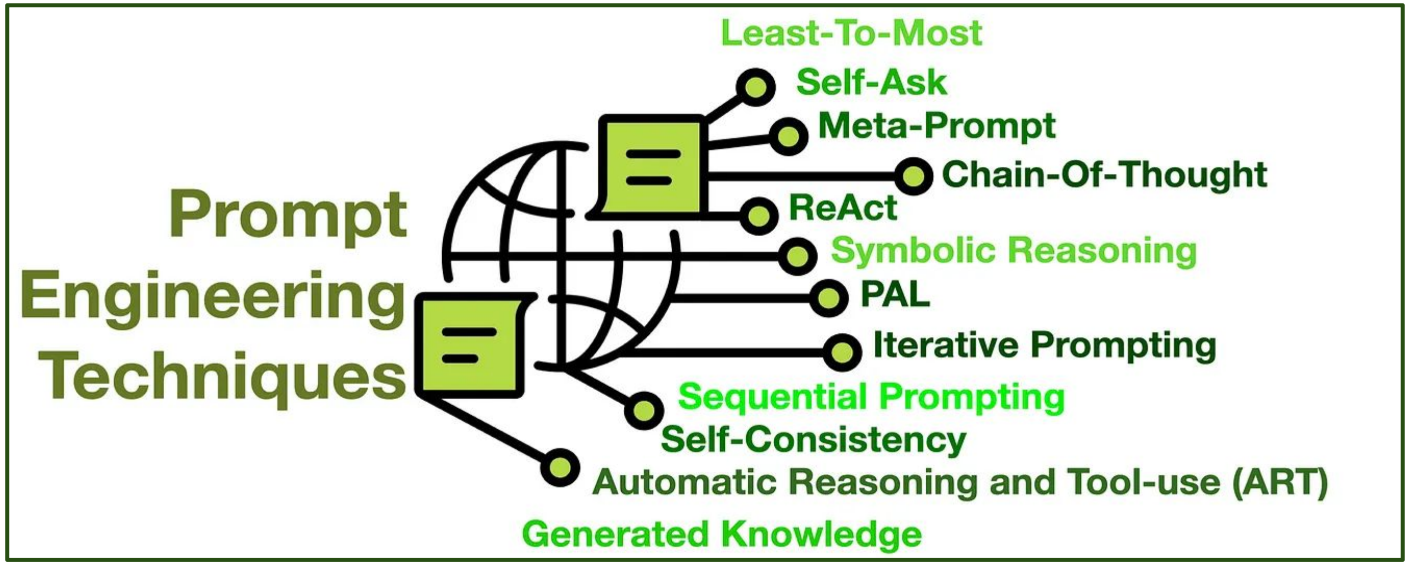
Instructions
Definitions
Background information
Questions
Examples
Images



https://arxiv.org/pdf/2307.12980.pdf

What is Prompt Engineering?

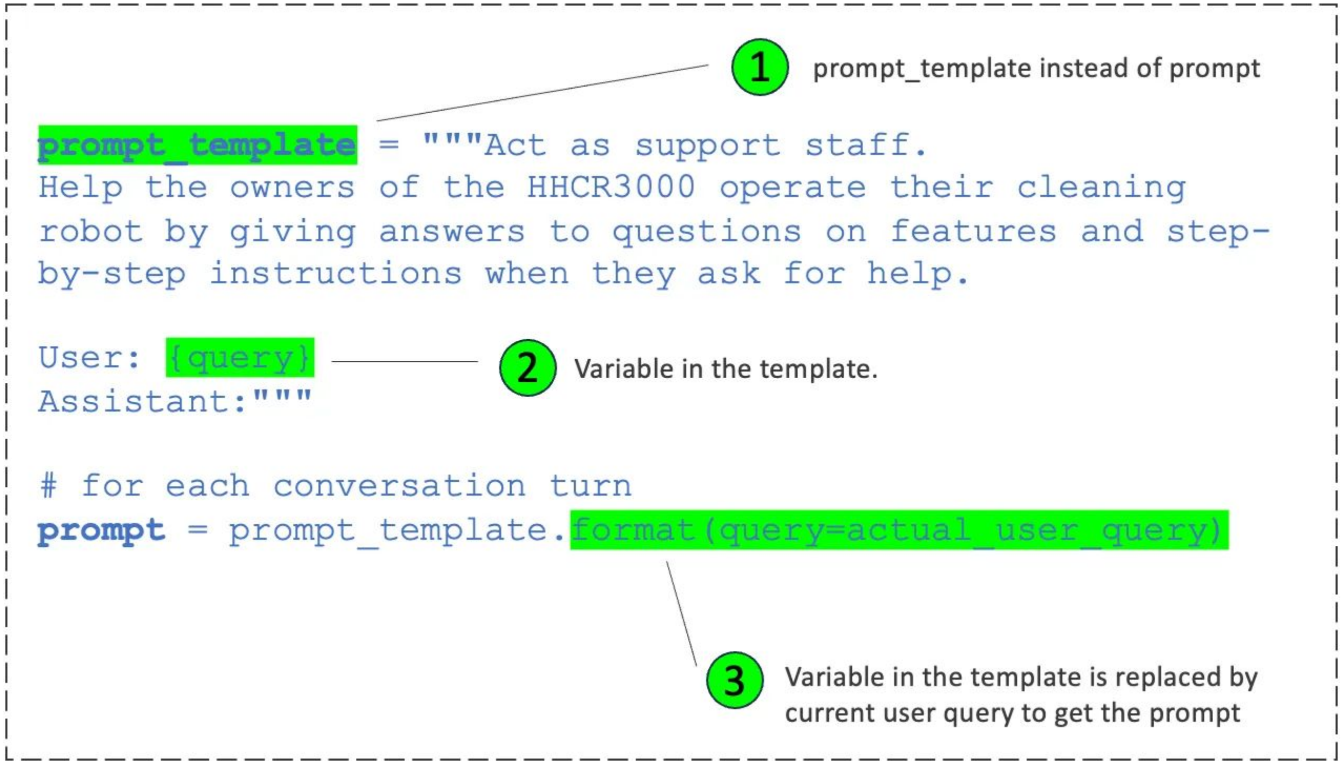
An iterative process of developing and optimizing prompts to efficiently use LLMs for a variety of tasks



https://cobusgreyling.medium.com/12-prompt-engineering-techniques-644481c857aa

Prompt Templates

A prompt is converted into a template with key and values replaced with placeholders. The placeholders are replaced with application values/variables ***at runtime***.



https://link.medium.com/phuemMDlPHb

Types of Prompts

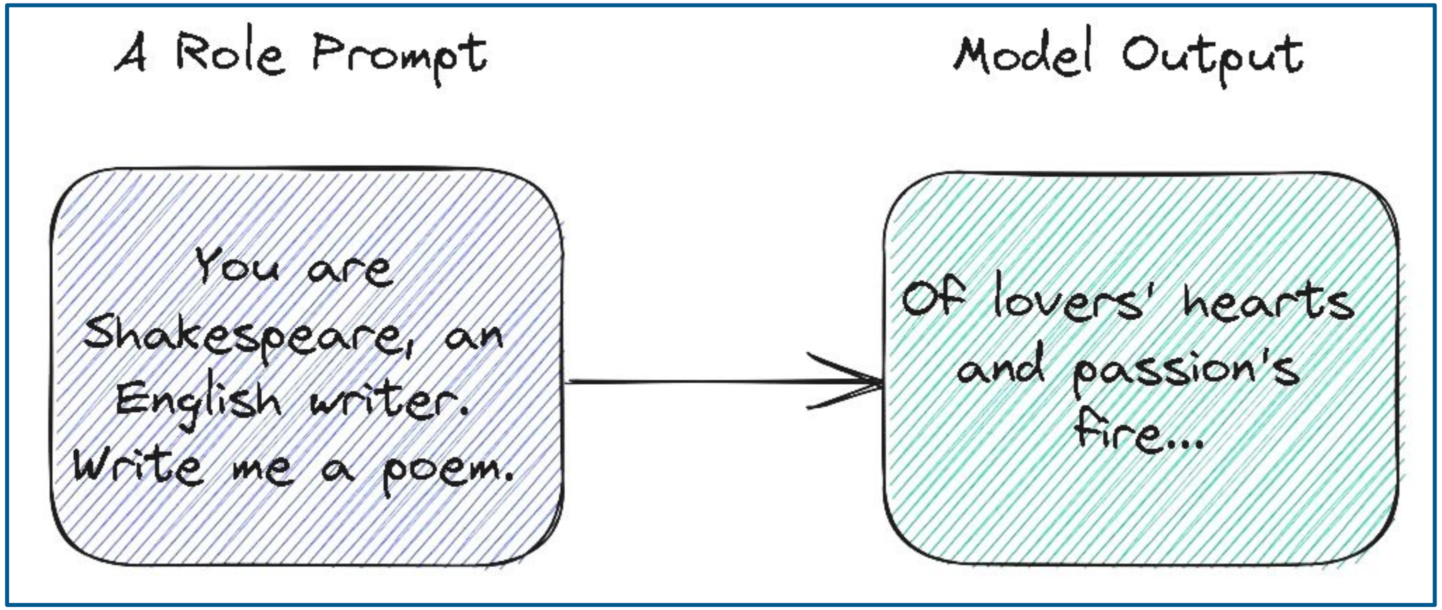
- Role-based Prompts
- Chain-of-Thought (CoT)
- Tree of Thoughts (ToT)
- Graph of Thoughts (GoT)
- Cross-Lingual-Thought Prompting
- Cross-Lingual Tree of Thoughts
- Iterative Prompting

Role Based Prompts

Aim: “set the tone of the conversation”

⇒ Model’s responses more relevant & increases the accuracy.

How: Specify the role the model should play.



https://www.linkedin.com/pulse/role-prompting-aris-ihwan/

Chain-of-Thought (CoT) Prompts

Aim: Improve the ability of LLM to perform complex reasoning

⇒ Instruct the model to "think" in smaller steps.

(Wei et al., 2022)

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?

Provide LLM with examples with a series of intermediate natural language reasoning steps that lead to final output

(Kojima et al., 2022)

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. ✓

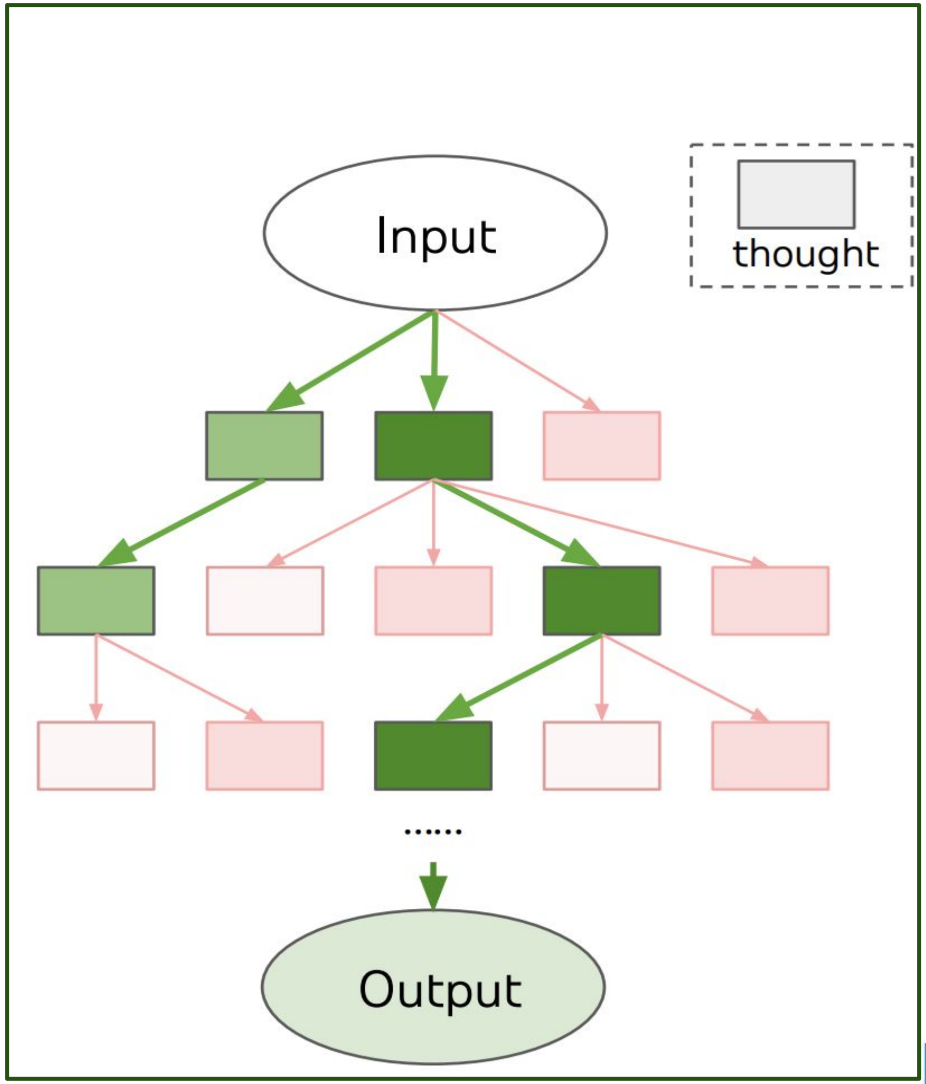
Ask model to "think step by step" without providing examples

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (Wei et al., arXiv 2022)
Large Language Models are Zero-Shot Reasoners (Kojima et al., arXiv 2022)

Tree of Thoughts (ToT) Prompts

Aim: Improve the ability of LLM in deliberate decision making by considering multiple different reasoning paths

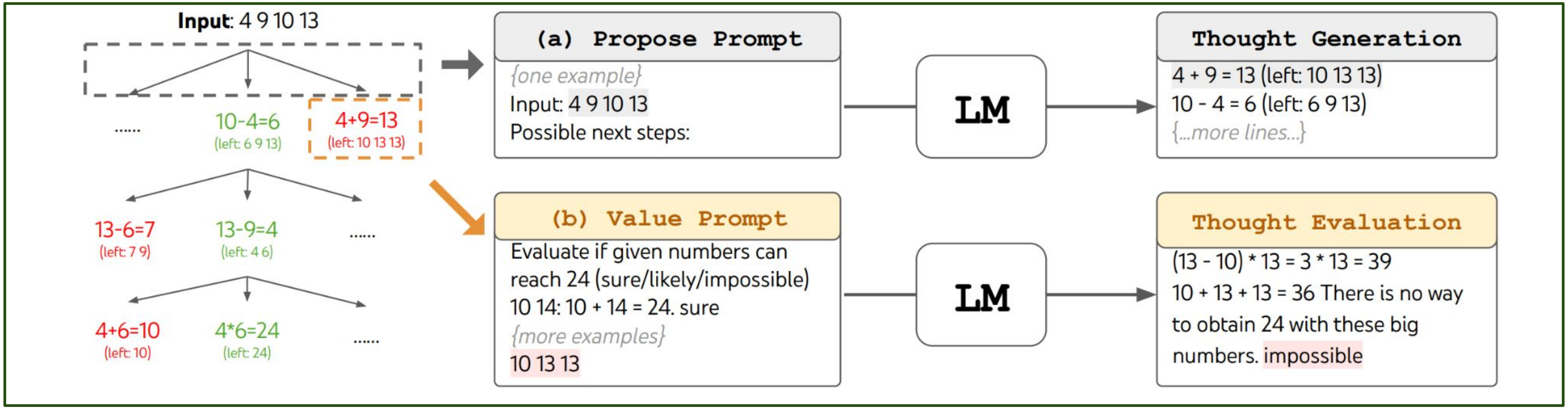
⇒ Model generates and evaluate thoughts, and search algorithms used to explore thoughts with lookahead and backtracking.



Tree of Thoughts: Deliberate Problem Solving with Large Language Models (Yao et al., NeurIPS 2023)

Tree of Thoughts (ToT) Prompts

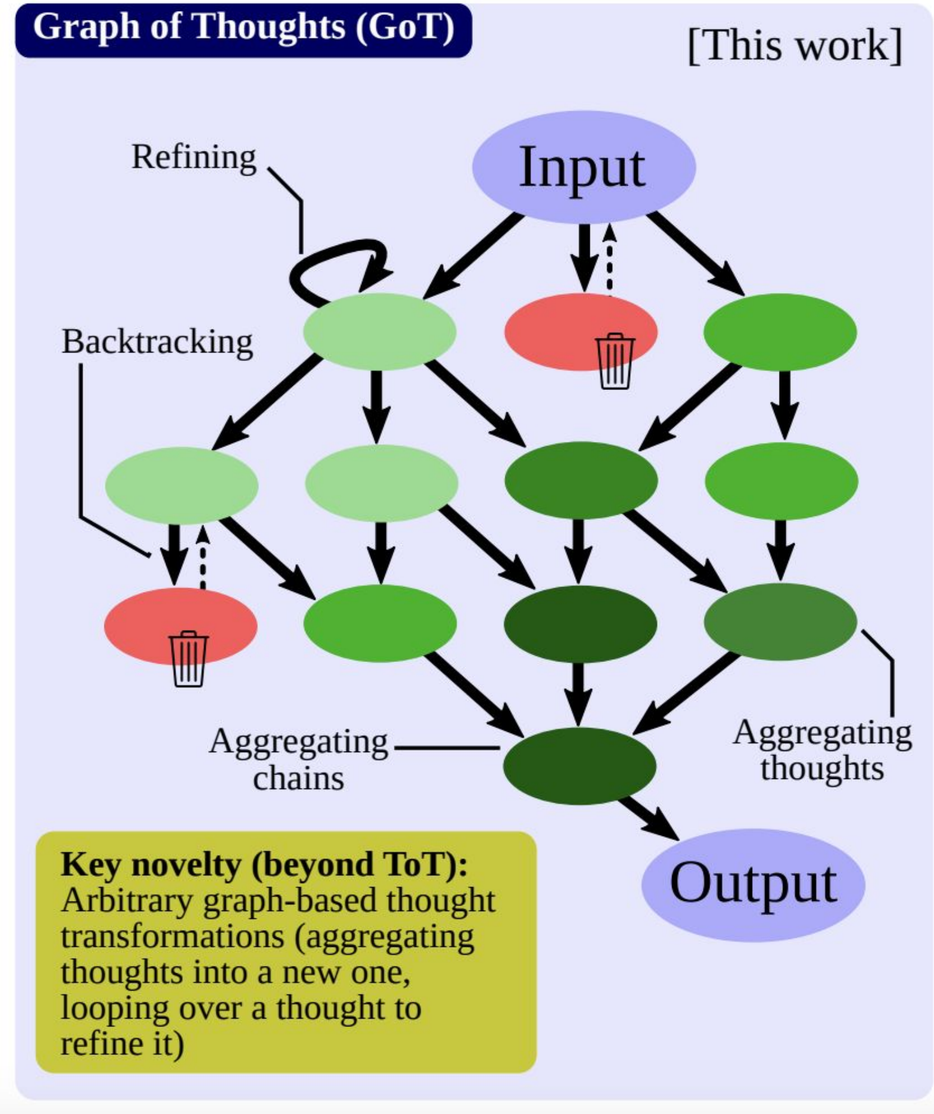
ToT for a game of 24 where the goal is to use 4 numbers and basic arithmetic operations (+-*/) to obtain 24.



Tree of Thoughts: Deliberate Problem Solving with Large Language Models (Yao et al., NeurIPS 2023)

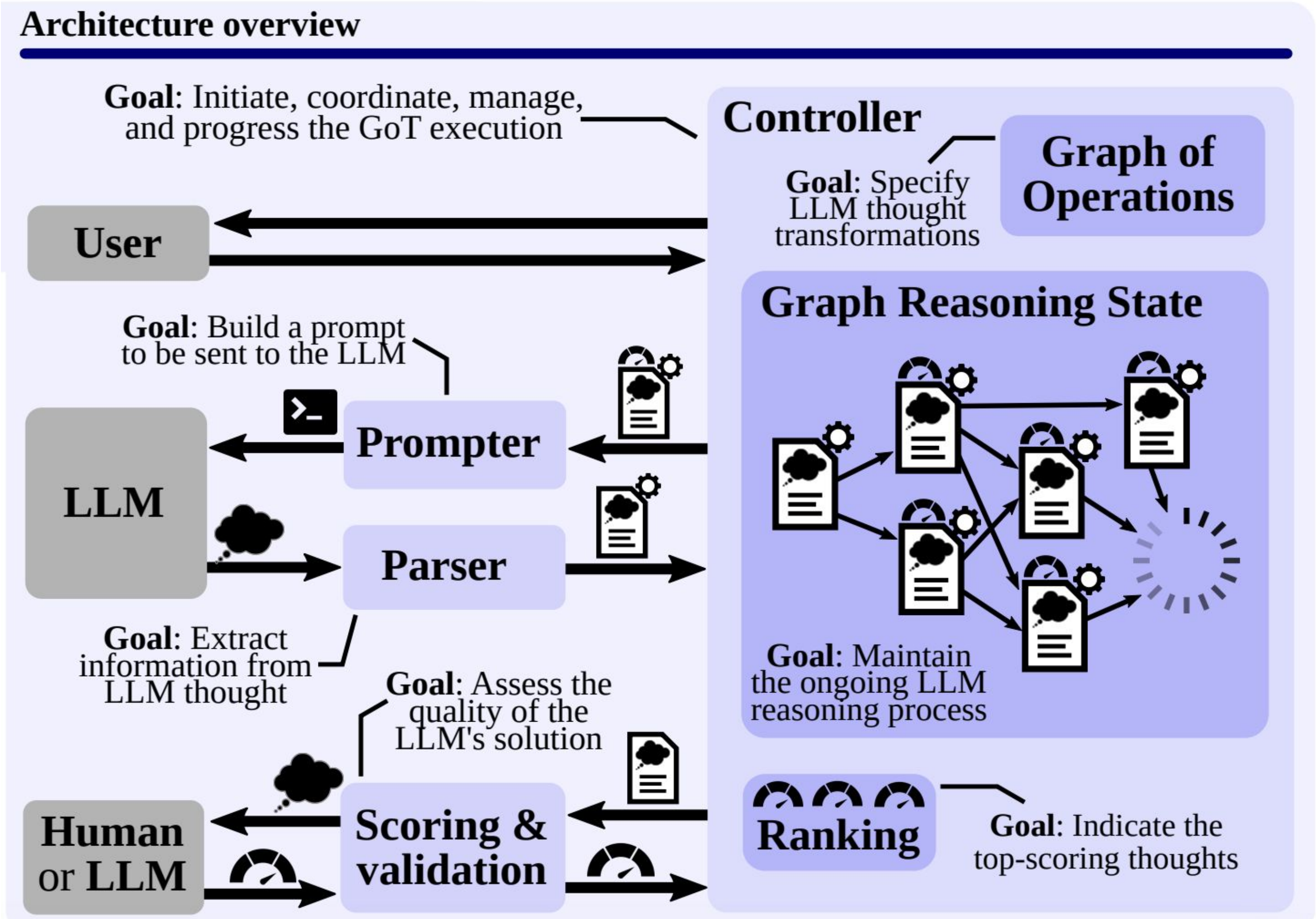
Graph of Thoughts (GoT) Prompts

Aim: Solve complex problems by modeling them as a Graph of Operations (GoO), which is automatically executed with an LLM as the engine



Graph of Thoughts: Solving Elaborate Problems with Large Language Models (Besta et al., arXiv 2024)

Graph of Thoughts (GoT) Prompts



Graph of Thoughts: Solving Elaborate Problems with Large Language Models (Besta et al., arXiv 2024)

Cross-Lingual-Thought Prompting

Aim: Improve the ability of LLM in performing tasks for multilingual inputs.

⇒ Create a prompt that uses both CoT (step-by-step) and asks the model to translate the input instruction/sample to English.

XLT

I want you to act as an arithmetic reasoning **expert for** Chinese.

Request: 詹姆斯决定每周跑 3 次 3 段冲刺，每段冲刺跑 60 米。他每周一共跑多少米?

You should retell the request **in English.**

You should do step-by-step answer to obtain a number answer.

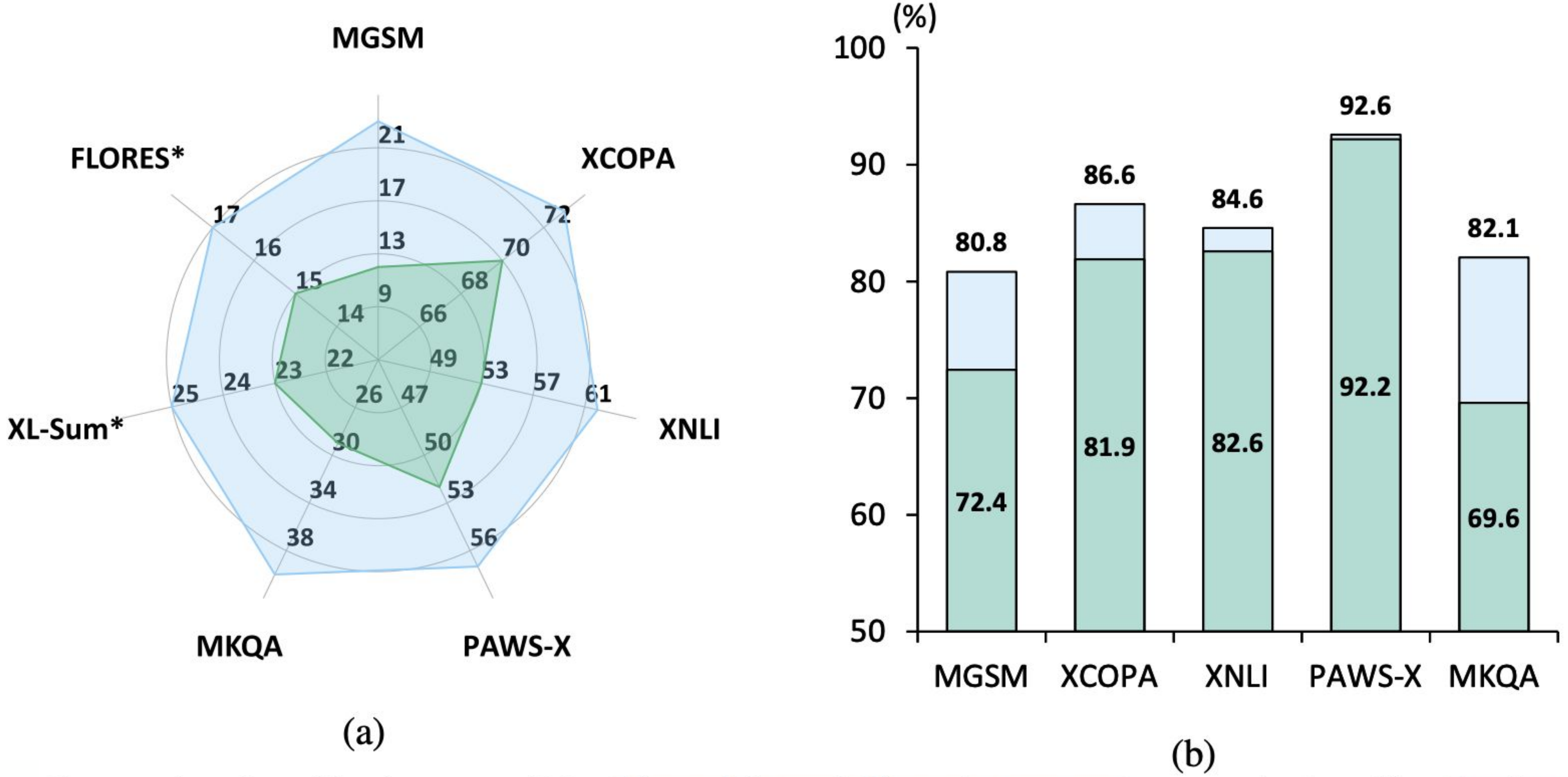
You should step-by-step answer the request.

You should tell me the answer **in this format '**Answer:**'.**

```
I want you to act as a task_name expert for task_language .
task_input
You should retell/repeat the input_tag in English.
You should task_goal .
You should step-by-step answer the request.
You should tell me the output_type ( output_constraint ) in this format ' output_type :'.
```


Not All Languages Are Created Equal in LLMs: Improving Multilingual Capability by Cross-Lingual-Thought Prompting (Huang et al., EMNLP 2023)

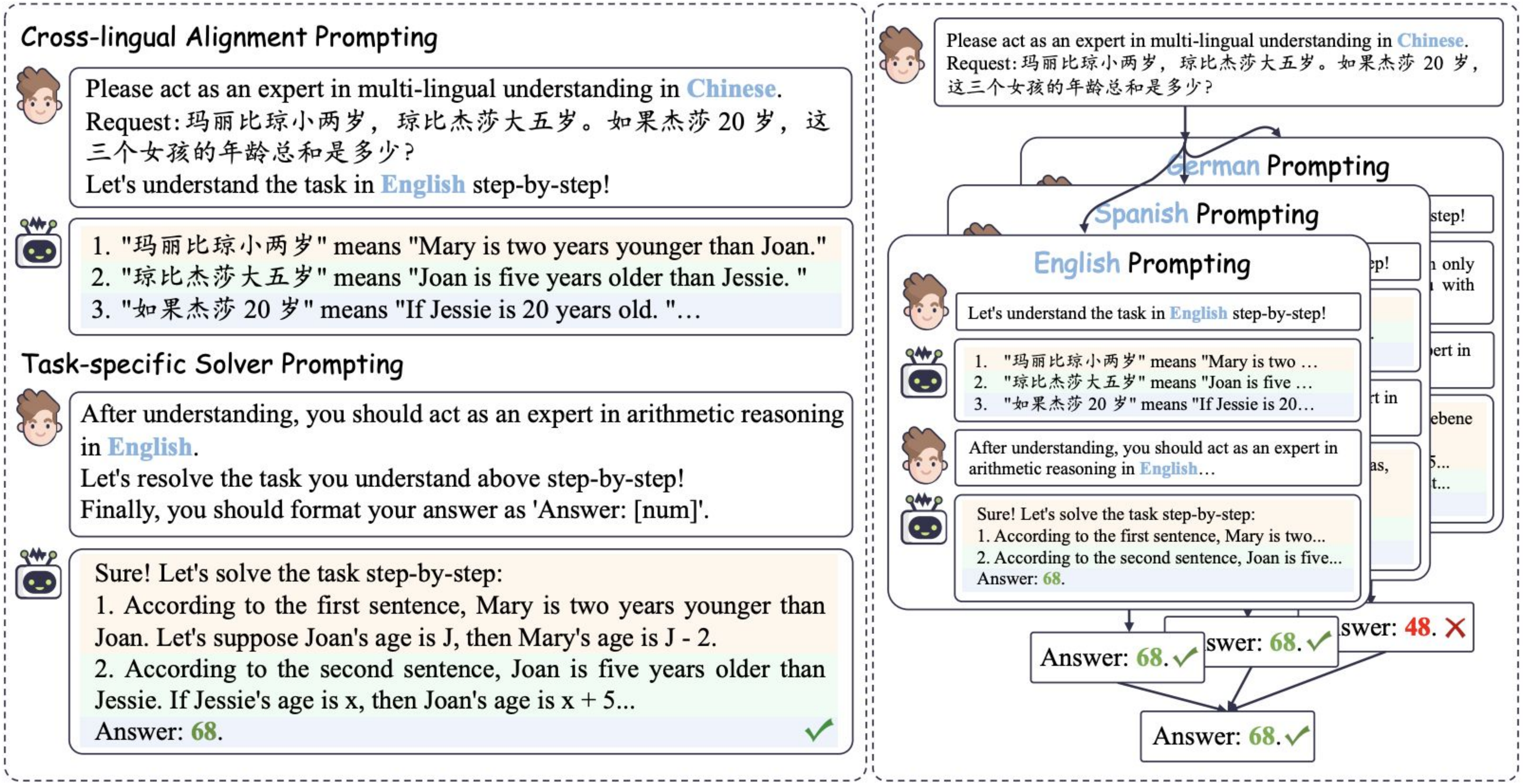
Cross-Lingual-Thought Prompting



Comparing the effectiveness of the Cross-Lingual-Thought prompt versus the baseline basic prompt

Not All Languages Are Created Equal in LLMs: Improving Multilingual Capability by Cross-Lingual-Thought Prompting (Huang et al., EMNLP 2023)

Cross-Lingual CoT Prompting



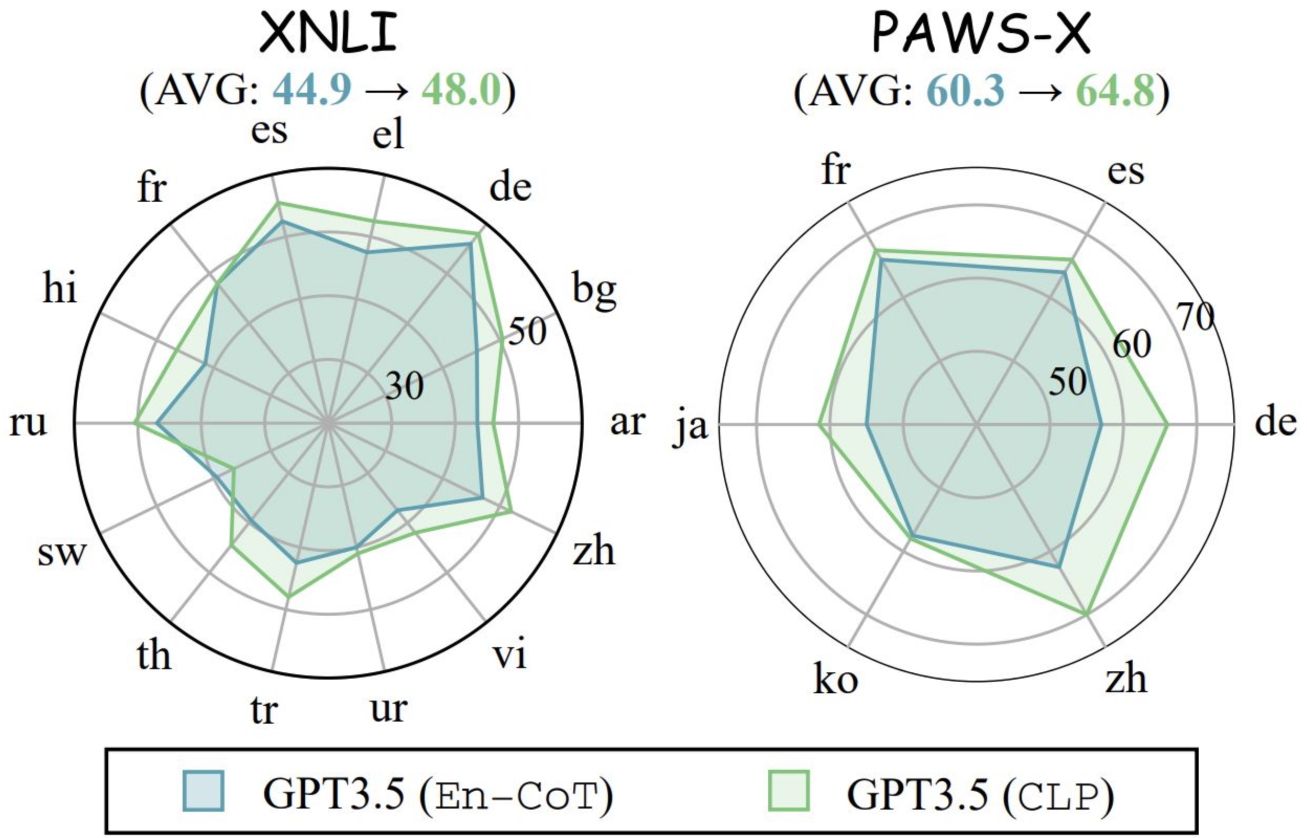
(a) Cross-lingual Prompting (CLP)

(b) Cross-lingual Self-consistent Prompting (CLSP)

Cross-lingual Prompting: Improving Zero-shot Chain-of-Thought Reasoning across Languages (Qin et al., EMNLP 2023)

Cross-Lingual CoT Prompting

Accuracy across languages in two tasks: XNLI and PAWS-X

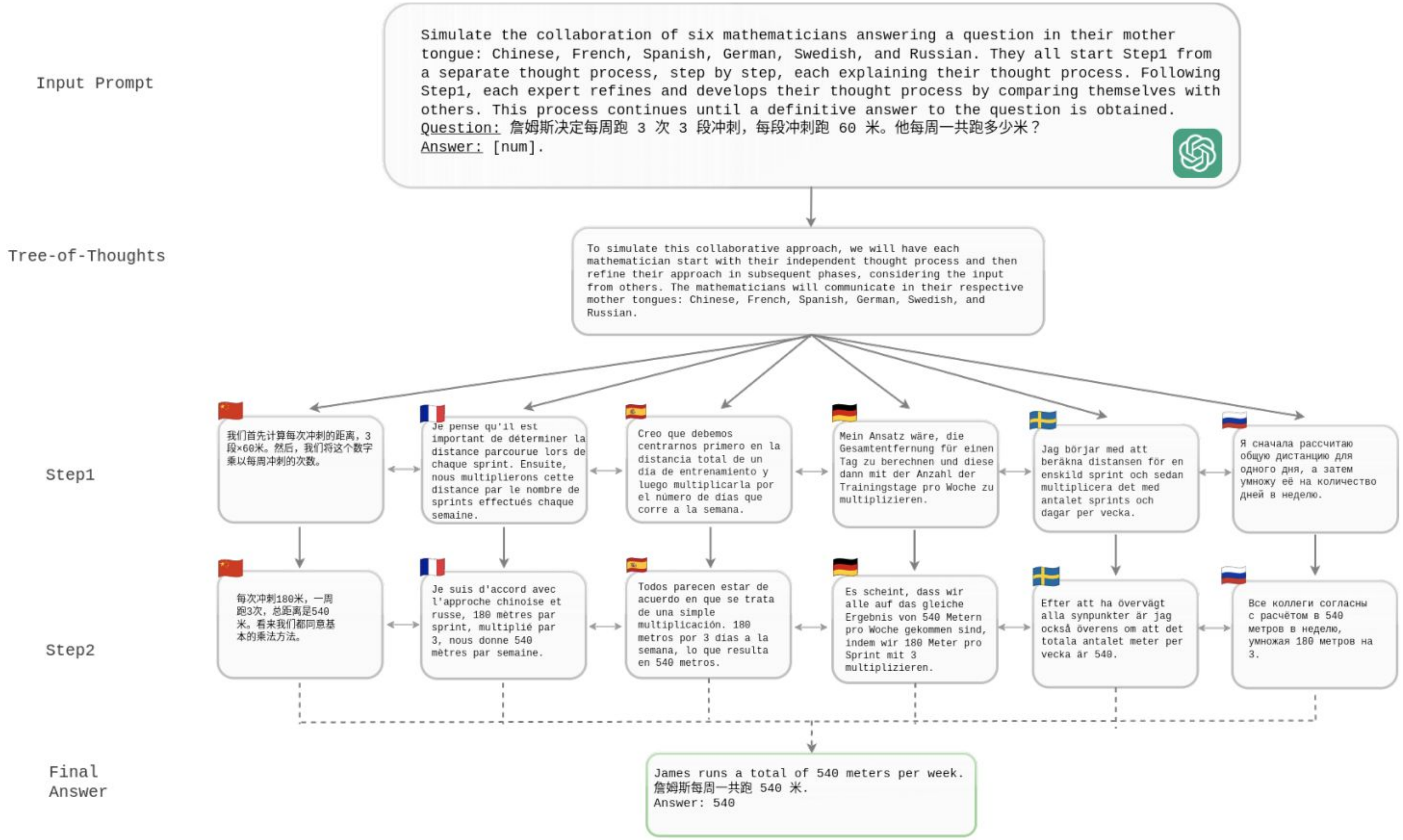


Cross-lingual Prompting: Improving Zero-shot Chain-of-Thought Reasoning across Languages (Qin et al., EMNLP 2023)

Cross-lingual ToT (Cross-ToT) Prompts

Aim: Improve the ability of LLM in deliberate decision making across languages by considering multilingual reasoning paths.

⇒ Use **ToT** style prompting to ask the LLM to deliver the reasoning process in different languages that, step-by-step, converge to a single final solution



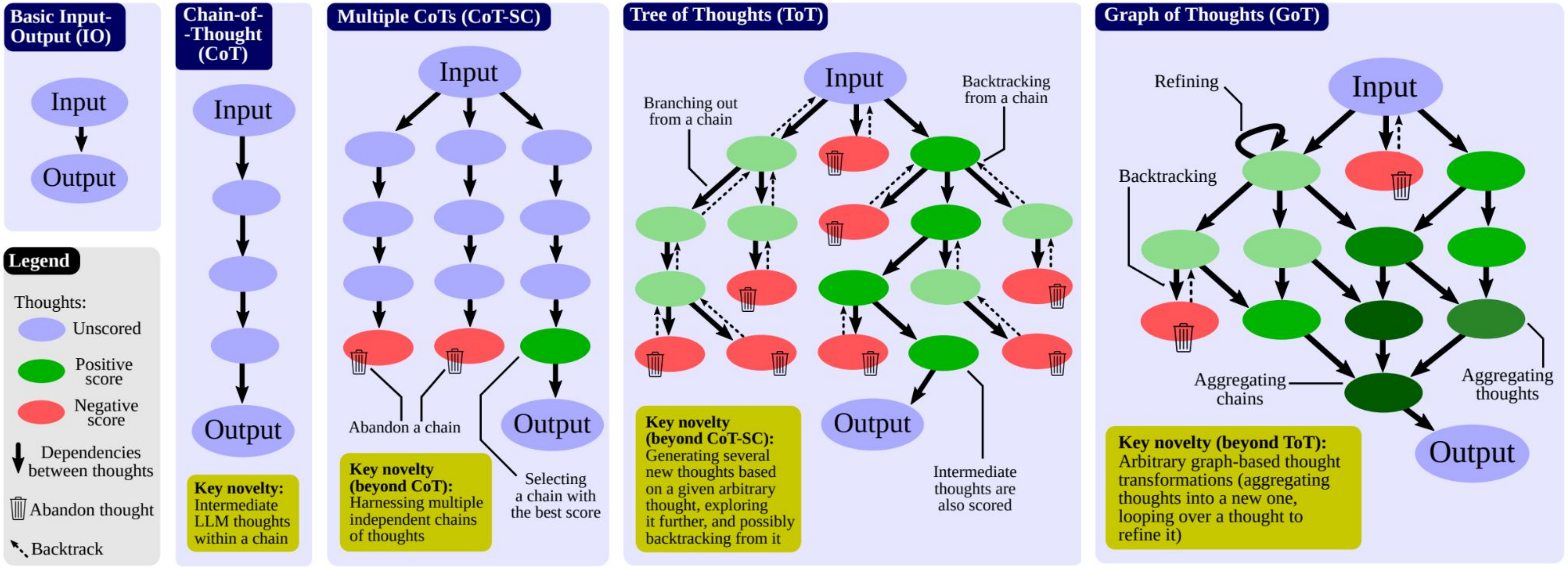
Empowering Multi-step Reasoning across Languages via Tree-of-Thoughts (Ranaldi and Zanzotto, arXiv 2023)

Cross-lingual ToT Prompts

Model	de	zh	fr	ru	sw	es	Average
GPT-3 (text-davinci-002)*							
Direct (Shi et al., 2022)	14.8	18.0	16.8	12.4	8.8	17.2	14.67
Native-CoT (Shi et al., 2022)	36.0	40.0	37.6	28.4	11.2	40.4	32.27
En-CoT (Shi et al., 2022)	44.0	40.8	46.0	28.4	20.8	44.8	37.47
Translate-En (Shi et al., 2022)	46.4	47.2	46.4	48.8	37.6	51.6	46.33
GPT-3.5 (gpt-3.5-turbo)							
Direct (Qin et al., 2023)	56.0	60.0	62.0	62.0	48.0	61.2	58.20
Native-CoT (Qin et al., 2023)	70.0	59.6	64.4	62.4	54.0	70.4	63.47
En-CoT (Qin et al., 2023)	73.6	63.2	70.0	65.6	55.2	69.6	66.20
Translate-En (Qin et al., 2023)	75.6	71.6	72.4	72.8	69.6	74.4	72.73
Cross-CoT (Qin et al., 2023)	86.8	77.2	82.0	**87.6**	**76.0**	84.8	82.40
Cross-ToT	**87.6**	**83.5**	**84.3**	86.5	75.4	**86.2**	**83.91**

Empowering Multi-step Reasoning across Languages via Tree-of-Thoughts (Ranaldi and Zanzotto, arXiv 2023)

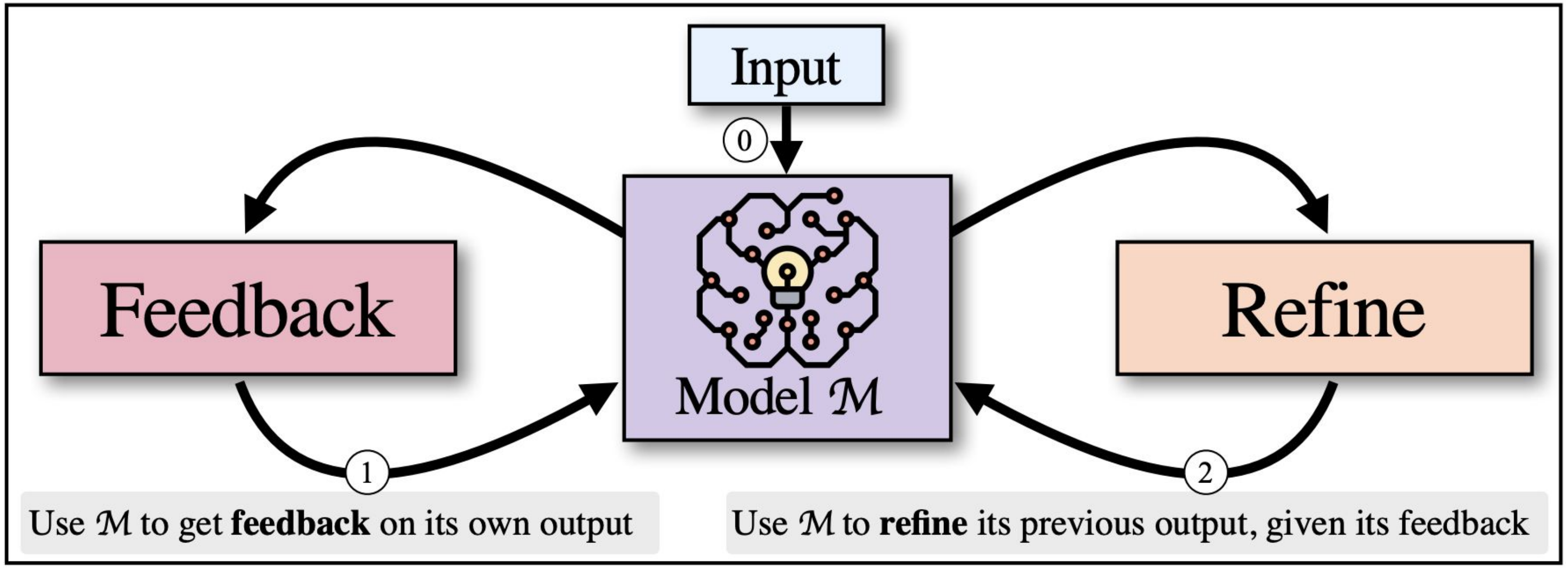
Comparing Prompting Techniques



Graph of Thoughts: Solving Elaborate Problems with Large Language Models (Besta et al., arXiv 2024)

Iterative Prompting

Aim: Improve LLM performance by iteratively prompting it to refine its previous responses.



SELF-REFINE: Iterative Refinement with Self-Feedback (Madaan et al., NeurIPS 2023)

Iterative Prompting

Self-refine technique: Prompt the same LLM iteratively with three prompts (for initial generation, feedback on generation, and refinement)

(a) **Dialogue:** x, $\mathbf{y_t}$

```
User: I am interested
in playing Table
tennis.

Response: I'm sure
it's a great way to
socialize, stay active
```

(b) **FEEDBACK** $\mathbf{fb}$

```
Engaging: Provides no
information about table
tennis or how to play it.

User understanding: Lacks
understanding of user's
needs and state of mind.
```

(c) **REFINE** $\mathbf{y_{t+1}}$

```
Response (refined): That's
great to hear (...) ! It's
a fun sport requiring
quick reflexes and good
hand-eye coordination.
Have you played before, or
are you looking to learn?
```

SELF-REFINE: Iterative Refinement with Self-Feedback (Madaan et al., NeurIPS 2023)

Iterative Prompting

	GPT-3.5		CHATGPT		GPT-4	
Task	Base	+SELF-REFINE	Base	+SELF-REFINE	Base	+SELF-REFINE
Sentiment Reversal	8.8	**30.4** (↑21.6)	11.4	**43.2** (↑31.8)	3.8	**36.2** (↑32.4)
Dialogue Response	36.4	**63.6** (↑27.2)	40.1	**59.9** (↑19.8)	25.4	**74.6** (↑49.2)
Code Optimization	14.8	**23.0** (↑8.2)	23.9	**27.5** (↑3.6)	27.3	**36.0** (↑8.7)
Code Readability	37.4	**51.3** (↑13.9)	27.7	**63.1** (↑35.4)	27.4	**56.2** (↑28.8)
Math Reasoning	**64.1**	**64.1** (0)	74.8	**75.0** (↑0.2)	92.9	**93.1** (↑0.2)
Acronym Generation	41.6	**56.4** (↑14.8)	27.2	**37.2** (↑10.0)	30.4	**56.0** (↑25.6)
Constrained Generation	16.0	**39.7** (↑23.7)	2.75	**33.5** (↑30.7)	4.4	**61.3** (↑56.9)

SELF-REFINE: Iterative Refinement with Self-Feedback (Madaan et al., NeurIPS 2023)

Automated Prompt Engineering

- **Prompt Mining**
  - Scrape a large text corpus (e.g., Wikipedia) for strings containing x and y, and finds either the middle words or dependency paths between the inputs and outputs.
- **Prompt Paraphrasing**
  - Take a seed prompt and paraphrase it into candidate prompts, then select the one that achieves the highest accuracy on the target task.
- **Prompt Generation**
  - Generate instruction candidates through an LLM for a task given output examples and select the most appropriate instruction based on computed evaluation scores.

In-Context/Few-shot Learning

Zero- vs. Few-shot Prompts

Classify the following sentence by the sentiment it expresses given these sentiments: Positive, Negative, Neutral, or Mixed.

Sentence: perfectly executed and wonderfully sympathetic characters
Sentiment:

Classify the following sentence by the sentiment it expresses given these sentiments: Positive, Negative, Neutral, or Mixed. Here are some examples:

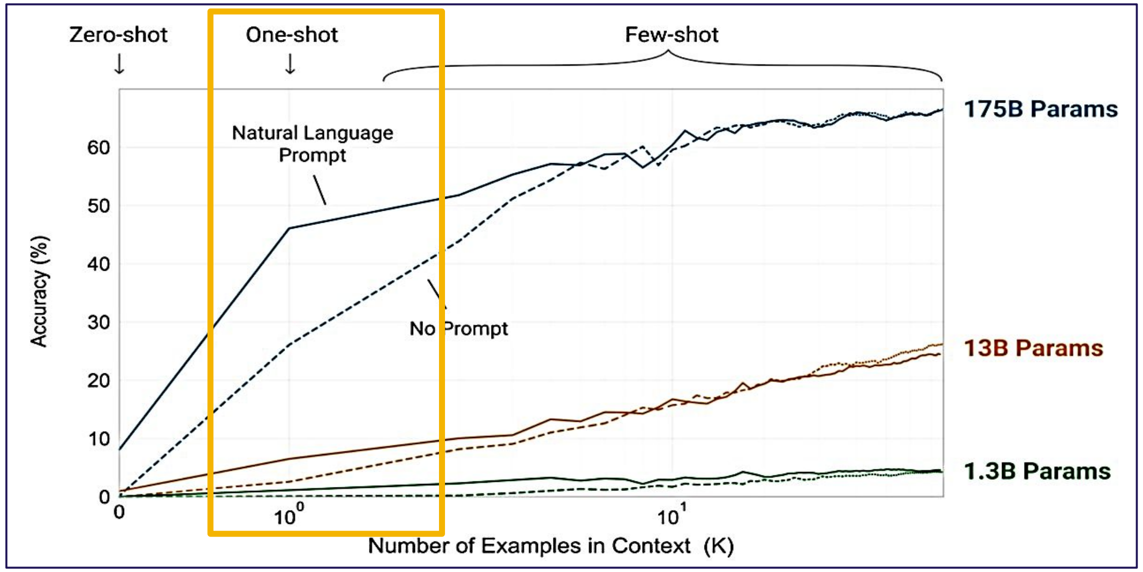
Sentence: a host of splendid performances
Sentiment: Positive

Sentence: felt trapped and with no obvious escape
Sentiment: Negative

Sentence: perfectly executed and wonderfully sympathetic characters
Sentiment:

Why?

- Improved performance over zero-shot
- Smaller task-specific dataset required (vs. fine-tuning)
- Model isn't updated, only pass the examples at inference time

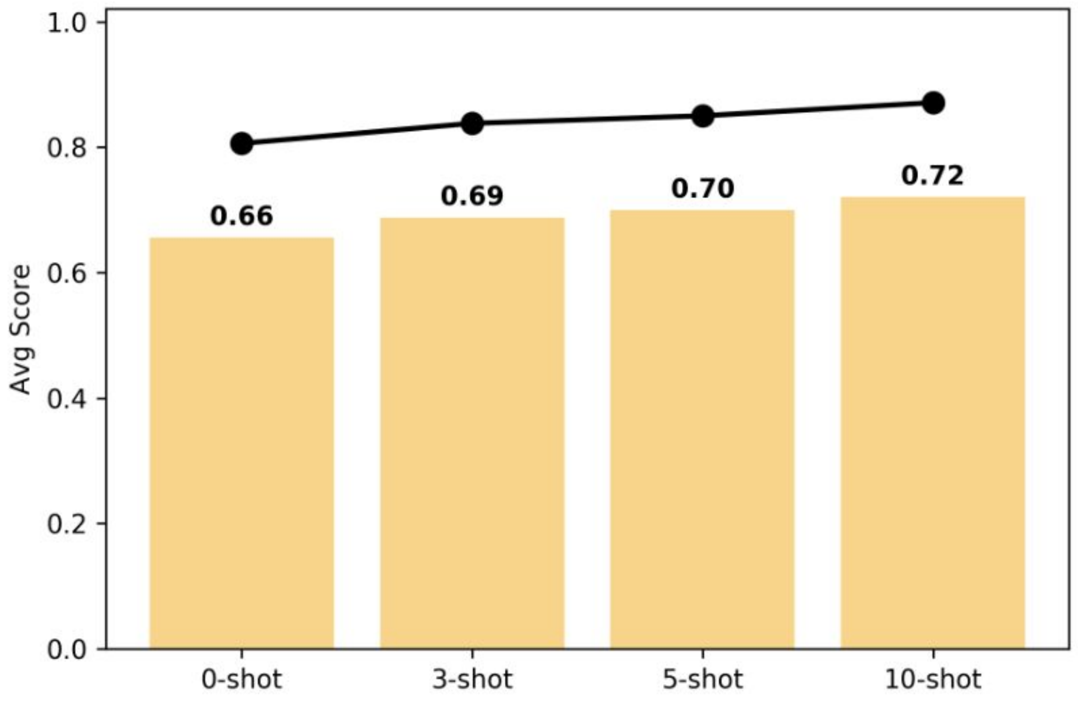


(Brown et al., 2020)

Language models are few-shot learners (Brown et al., arXiv 2020)

How Many Examples?

- Great range of values: [1,2,3,...,48,...]
- Consider document/example length:
  - LLMs have a fixed context window (e.g. GPT-3.5 allows 4,097 tokens as input)
- Tune as hyperparameter on developme set



(Abdelali et al., 2024)

LAraBench: Benchmarking Arabic AI with Large Language Models (Abdelali et al. EACL, 2024)

Which Examples?

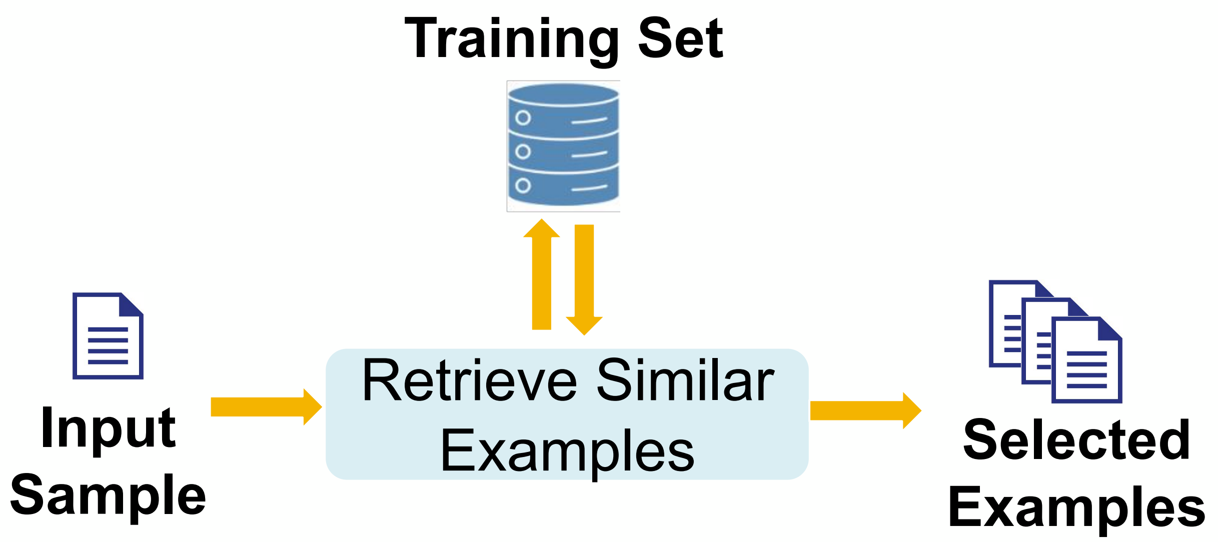
Manual

- Select some examples manually

Sampling

- Uniform class distribution
- Randomly
  ⇒ Might lead to skewed label distribution

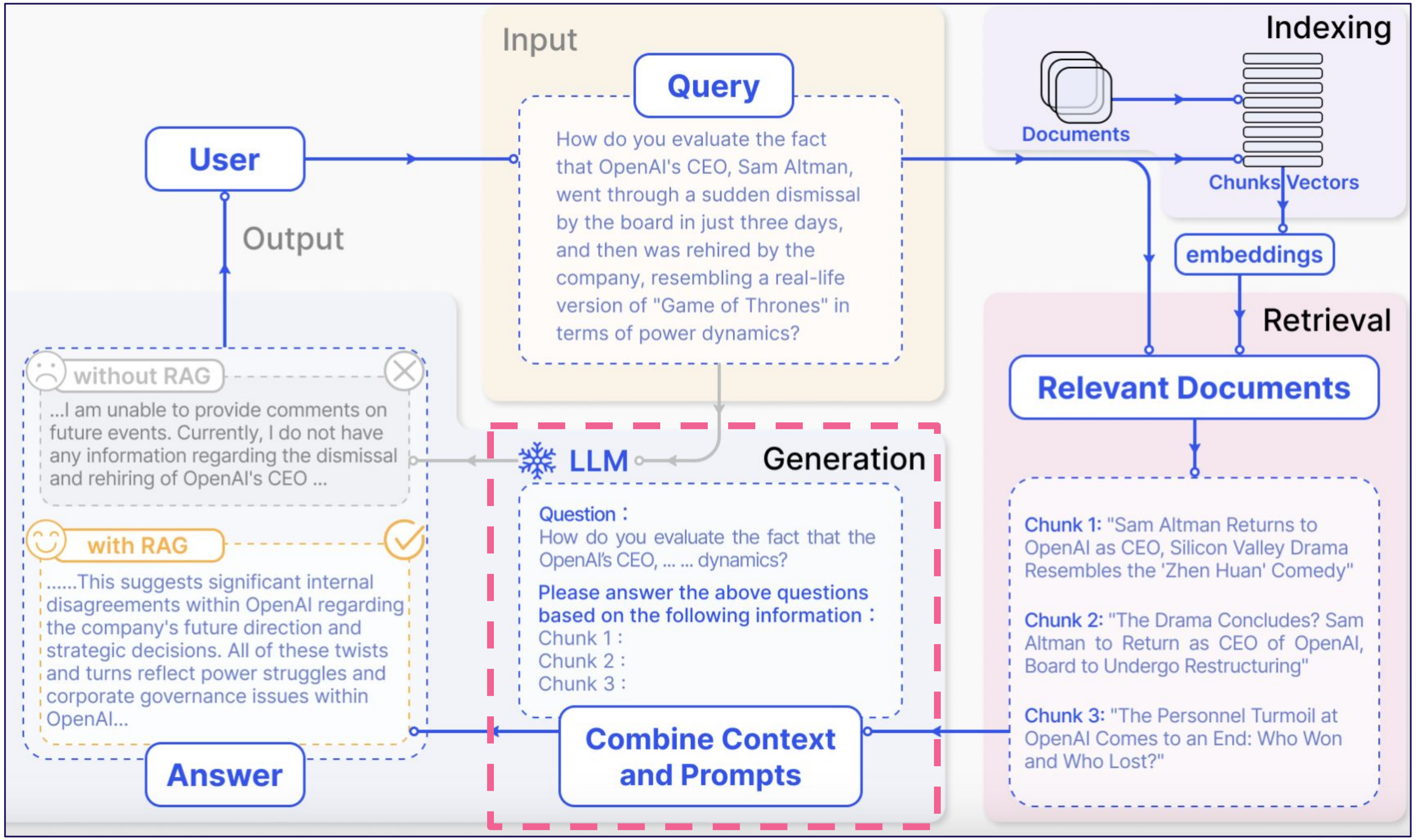
Semantic Similarity



- Cosine similarity
- Word overlap
- Maximal marginal relevance

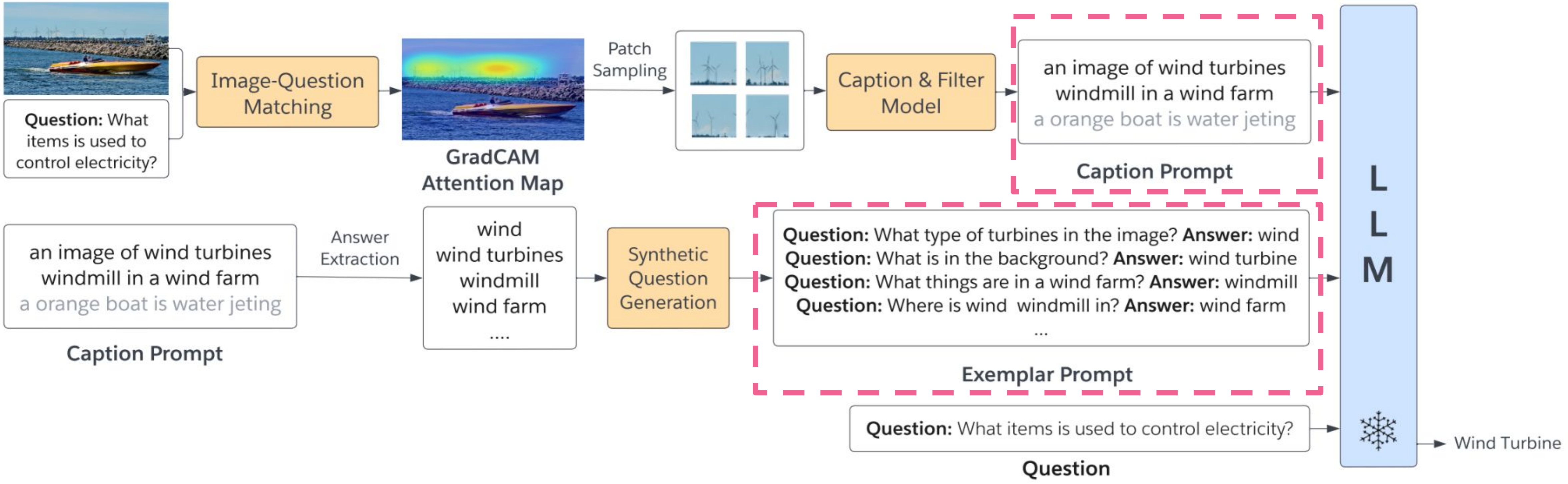
Retrieval-Augmented Generation (RAG)

Aim: Provide additional context for the LLM, leading to improved factual accuracy and coherence in its output.



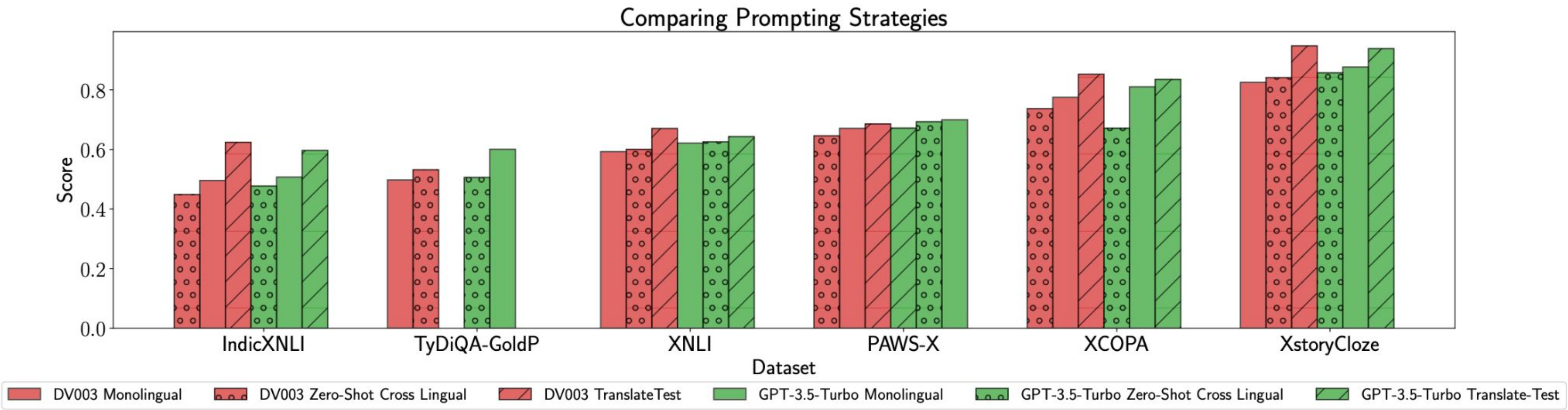
Retrieval-Augmented Generation for Large Language Models: A Survey (Gao et al. arXiv, 2024)

Context for Tasks on Images



From Images to Textual Prompts: Zero-shot Visual Question Answering with Frozen Large Language Models (Guo et al. CVPR, 2023)

Mono-/Cross- Language Prompting



- **Monolingual Prompting:** Few shot examples + test sample in same language.
- **Zero-Shot Cross-Lingual:** Few shot English examples + test sample in different language.
- **Translate-Test:** Few shot English examples + test sample translated to English.

MEGA: Multilingual Evaluation of Generative AI (Ahuja et al. arXiv, 2023)

Mono-/Cross- Language Prompting

Classify the 'sentence' as subjective or objective. Provide only label.
sentence: “والصحيح هو أن السيد أحمد منصور له مواقف ضد الفكر السلفي.”
label:

صنف "الجملة" إلى لاموضوعية أو موضوعية.
الجملة: "والصحيح هو أن السيد أحمد منصور له مواقف ضد الفكر السلفي."
التصنيف:

Task Name	Metric	English	Arabic
NER	Macro-F1	0.355	0.350
Sentiment	Macro-F1	0.569	0.547
News Cat.	Macro-F1	0.667	0.739
Gender	Macro-F1	0.868	0.892
Subjectivity	Macro-F1	0.677	0.725
XNLI (Arabic)	Acc	0.753	0.740
QA	F1 (exact match)	0.705	0.654
Average		**0.656**	**0.664**

LAraBench: Benchmarking Arabic AI with Large Language Models (Abdelali et al. EACL, 2024)

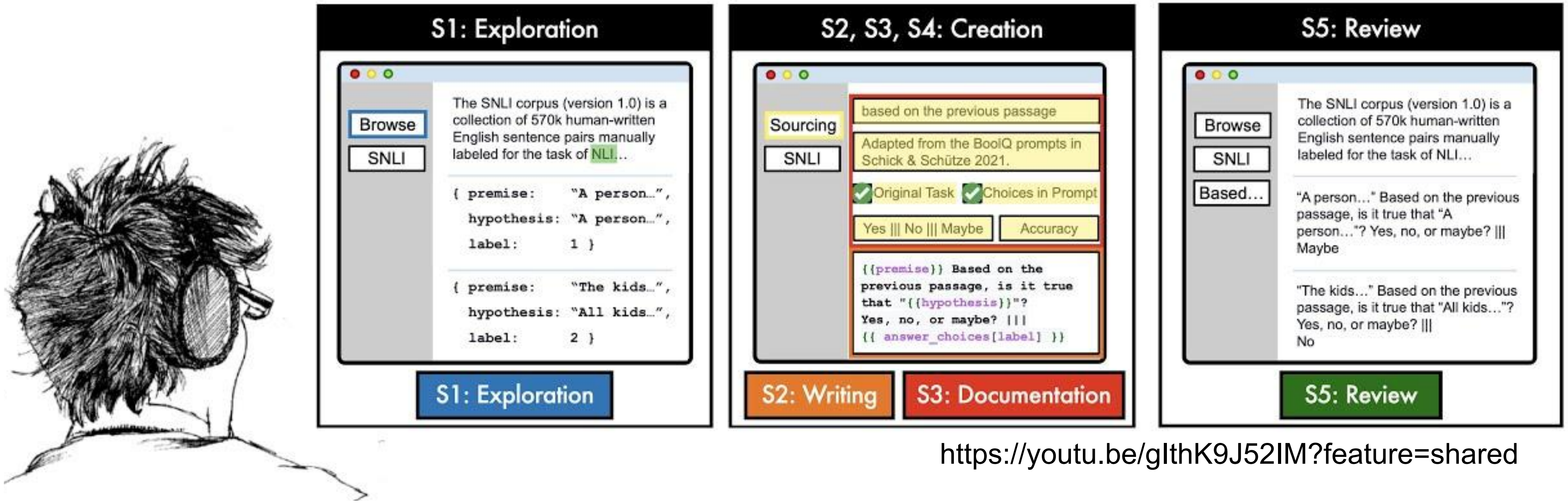
Prompting and Benchmarking Tools

Prompting and Benchmarking Tools

- **Prompt Source** (Bach et al. 2022)
- **LLMeBench** (Dalvi et al., 2023)
- **lm-evaluation-harness** (Gao et al., 2023)
- **Open ICL** (Wu et al., 2023)
- **Prompt Bench** (Zhu et al., 2023)

Prompt Source

"a system for creating, sharing, and using natural language prompts"



https://youtu.be/gIthK9J52IM?feature=shared

https://github.com/bigscience-workshop/promptsource

PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts (Bach et al., ACL 2022)

Prompt Source

Five stages of creating prompts:

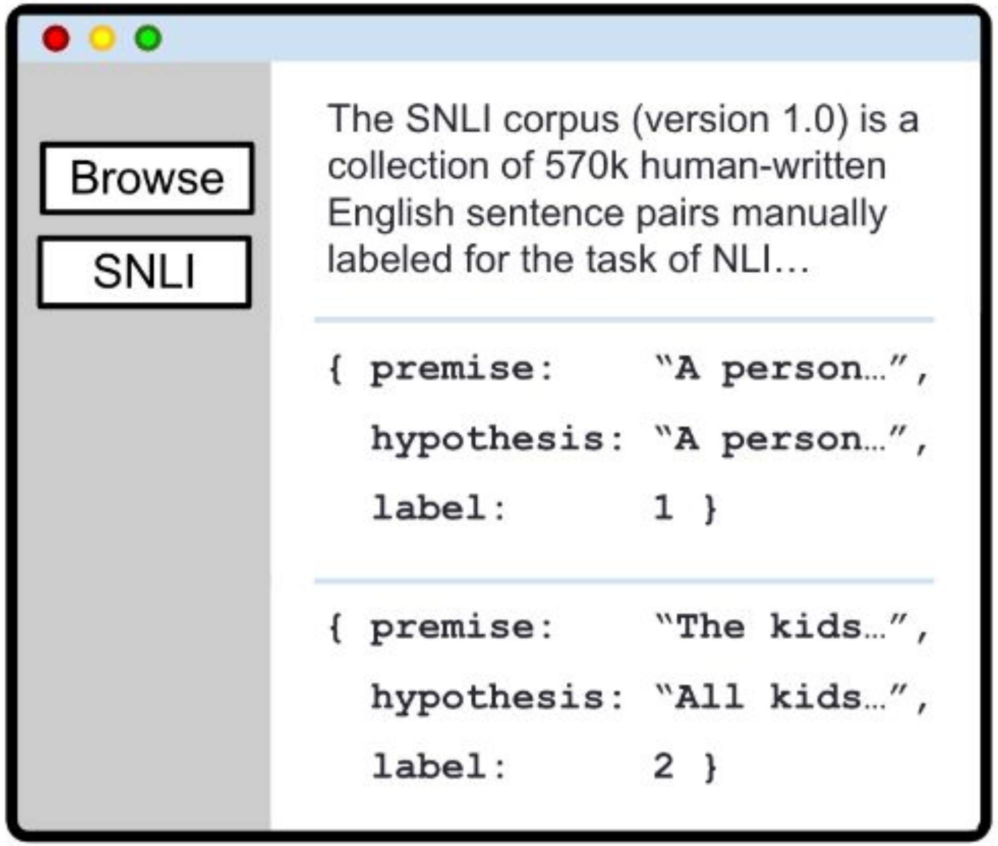
S1: Dataset Exploration

SNLI dataset example:

Assume a given premise sentence is true, the goal is to determine whether a hypothesis sentence is:

- true (entailment),
- false (contradiction),
- or undetermined (neutral)

S1: Exploration



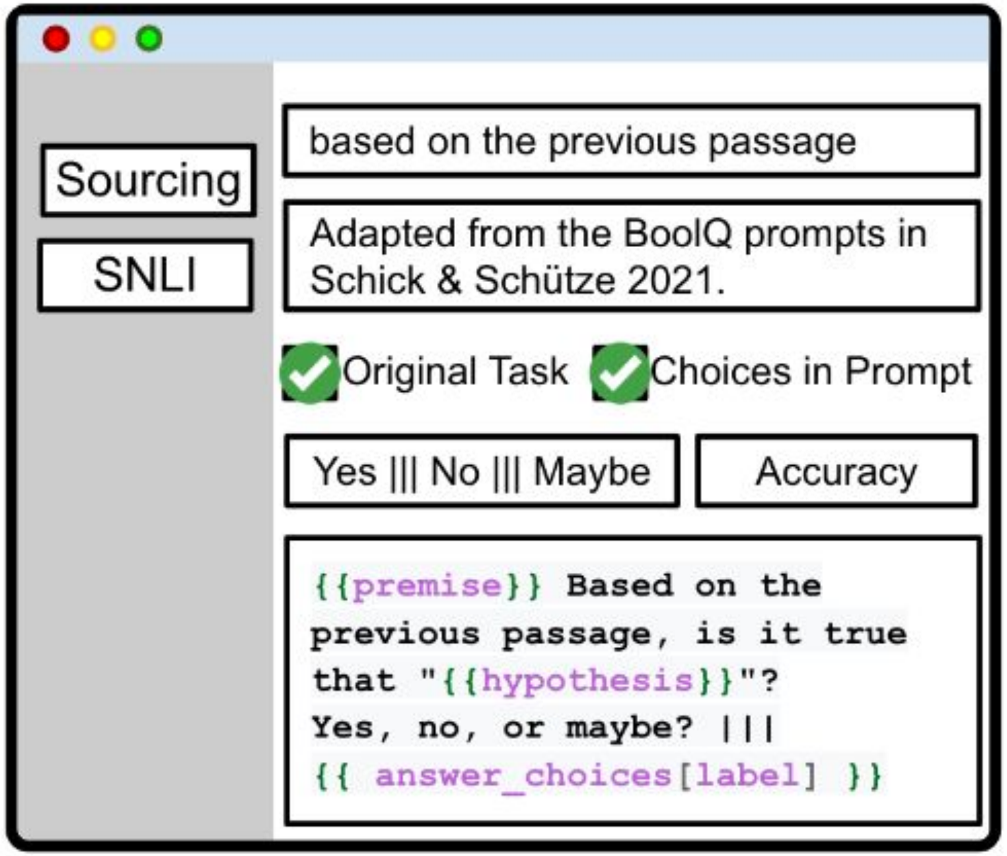
PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts (Bach et al., ACL 2022)

Prompt Source

Five stages of creating prompts:

S2: Prompt Writing
S3: Prompt Documentation
S4: Iteration and Variation

S2 + S3 + S4: Creation

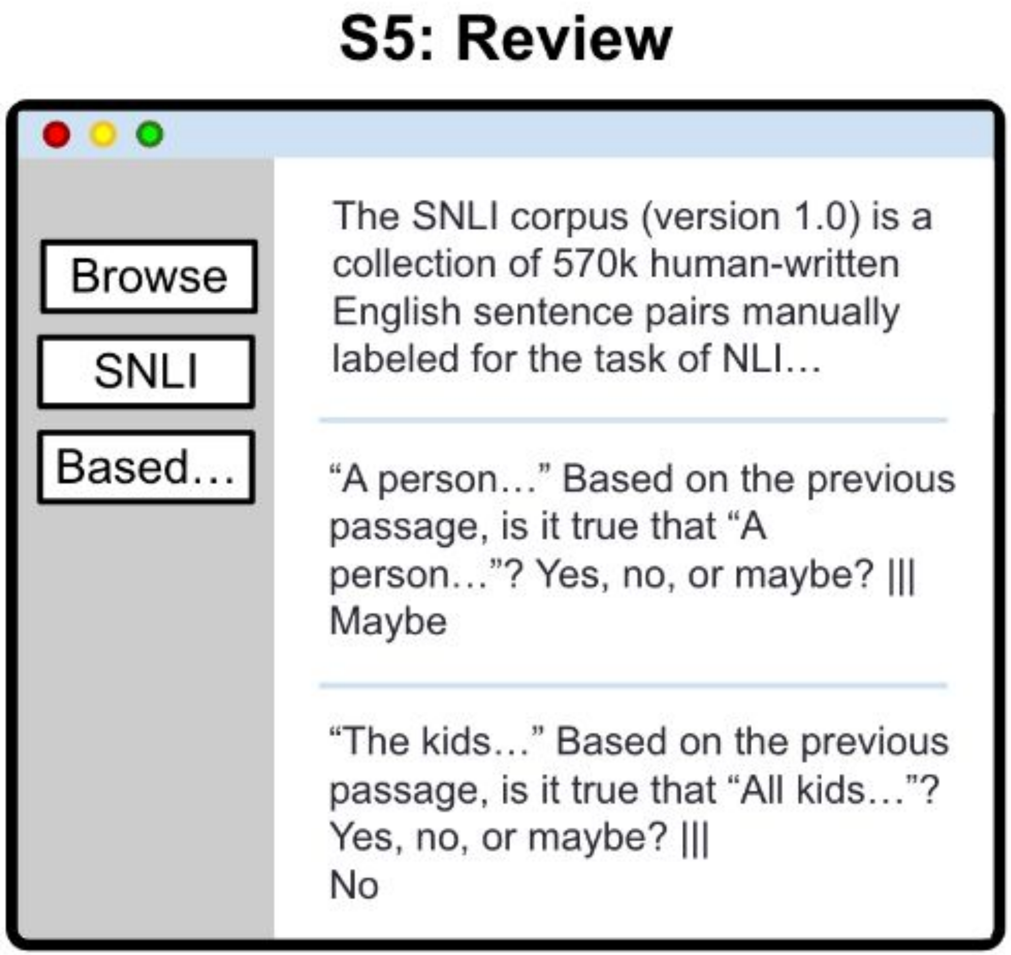


PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts (Bach et al., ACL 2022)

Prompt Source

Five stages of creating prompts:

S5: Global Review



PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts (Bach et al., ACL 2022)

Prompt Source

Prompt Template Creation

```
# Load an example from the datasets ag_news
>>> from datasets import load_dataset
>>> dataset = load_dataset("ag_news", split="train")
>>> example = dataset[1]

# Load prompts for this dataset
>>> from promptsource.templates import DatasetTemplates
>>> ag_news_prompts = DatasetTemplates('ag_news')

# Print all the prompts available for this dataset. The keys of the dict are the UUIDs the u
>>> print(ag_news_prompts.templates)
{'24e44a81-a18a-42dd-a71c-5b31b2d2cb39': <promptsource.templates.Template object at 0x7fa7ae

# Select a prompt by its name
>>> prompt = ag_news_prompts["classify_question_first"]

# Apply the prompt to the example
>>> result = prompt.apply(example)
>>> print("INPUT: ", result[0])
INPUT:  What label best describes this news article?
Carlyle Looks Toward Commercial Aerospace (Reuters) Reuters - Private investment firm Carly]
>>> print("TARGET: ", result[1])
TARGET:  Business
```

PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts (Bach et al., ACL 2022)

LLMeBench

Make it super-simple and quick to **start experimenting** with LLMs,
and **easily transfer that effort** to large scale evaluation

http://llmebench.qcri.org/

LLMeBench: A Flexible Framework for Accelerating LLMs Benchmarking, (Dalvi et al. EACL, 2024)

LLMeBench: Usecases

Exploration	Model comparison	Benchmarking suite	Many more...
Try a model with different prompts over the same dataset	Run the same prompt with multiple models	Create a suite of tasks and datasets and track a model's progress across all	Framework is flexible and extensible for new tasks, datasets, and models

Why LLMeBench?

Current LLM usage and benchmarking process

1. Read the data
2. Figure out how to access an LLM (e.g. GPT4)
3. Understand and write code to read the response
4. Explore with different prompts
5. Write some sort of loop over the data and prompts to see model responses on all samples
    a. Realize the request fails for many reasons ⇒ Write some code to retry failed requests
    b. Realize every time you run your code, you get different results ⇒ Modify code to set all appropriate model parameters for reproducible results
    c. Have an idea for a new prompt, figure out changing existing code to only run for new prompt while keeping results from older prompts
6. Process results
7. Rinse and Repeat for a new problem/dataset/task

Why LLMeBench?

1. Find your task, dataset and model in LLMeBench
   ⇒ Task/Data/Model not found?
   a. Edit existing task/data/model script for your needs
2. Run experiment!

Add a layer of abstraction so that you as a user can focus solely on getting the best performance out of the LLM

LLMeBench

benchmarking asset

```
def config():
    return {
        "dataset": TSVDataset,
        "dataset_args": {
            "column_mapping": {
                "input": "sentence",
                "label": "labels",
            },
        },
        "task": ClassificationTask,
        "model": FastChatModel,
        "general_args": {"custom_test_split": "SST-2/dev.tsv"},
    }

def prompt(input_sample):
    return [
        {"role": "system", "content": "You are an expert in sentiment analysis."},
        {"role": "user", "content": f"Sentence: {input_sample}\nSentiment:"}
    ]

def post_process(response):
    out = response["choices"][0]["message"]["content"].lower()
    return 1 if "positive" in out else 0
```

LLMeBench

Once an *asset* is written, LLMeBench takes care of everything else!

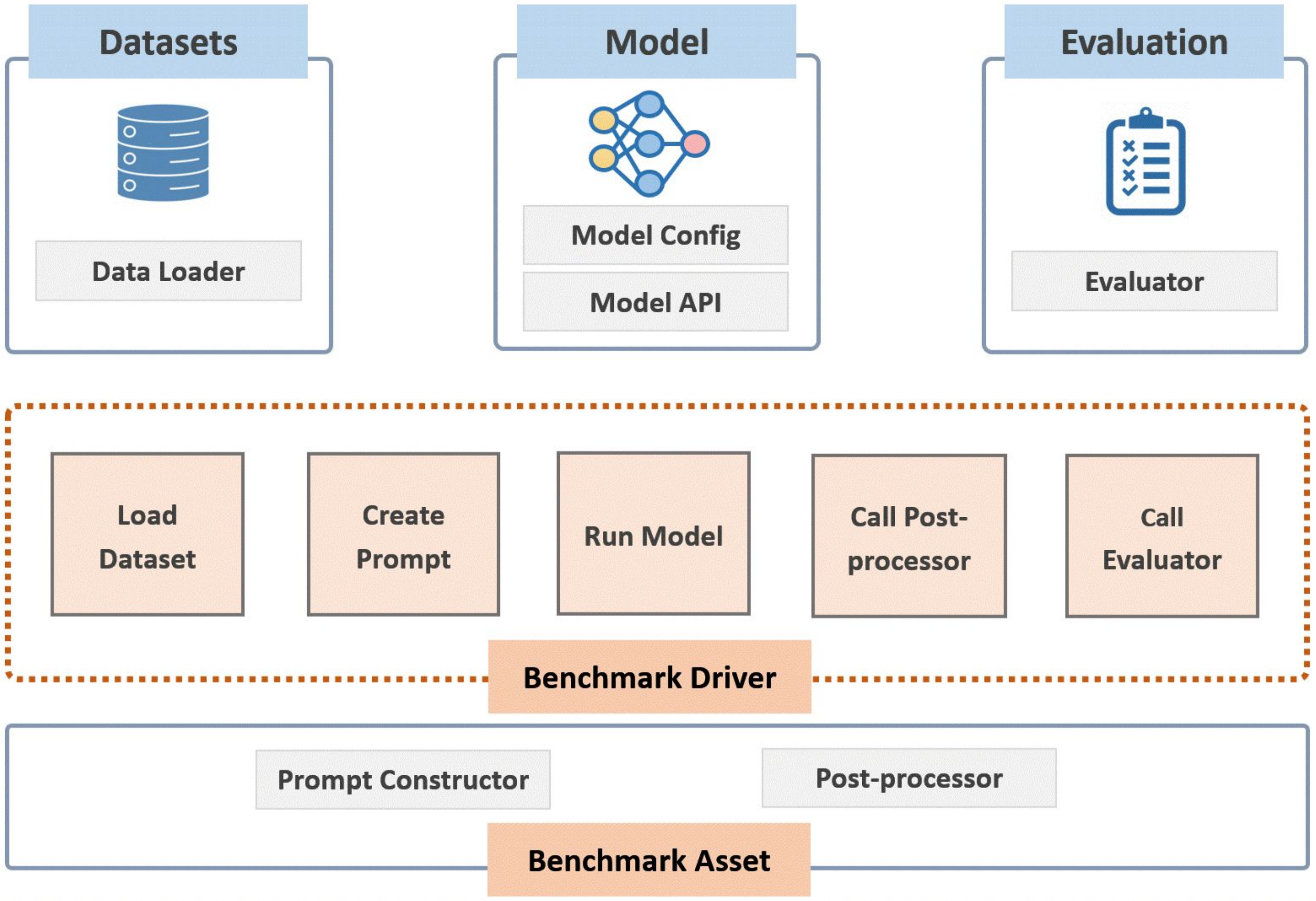
```
python -m llmebench assets/ results/
```


```
{
  "num_processed": 872,
  "num_failed": 0,
  "evaluation_scores": {
    "Macro F1": 0.8586052694703862,
    "Micro F1": 0.8612385321100917,
    "Acc": 0.8612385321100917,
    "Weighted Precision": 0.8821528346701518,
    "Weighted Recall": 0.8612385321100917,
    "Weighted F1": 0.8589593215900104
  }
}
```

LLMeBench Features

- **~300 assets** across **12 languages**
- Extensive support for **reading datasets**
  - HuggingFace datasets + generic data loaders (csv, tsv, json)
  - Over 50 dataset-specific loaders
  - Automatic downloading of data (when allowed)
- Supports popular **task types** (Classification, regression etc.)
- Supports popular **model providers** (OpenAI, FastChat, Petals, HuggingFace Inference API)
- Extensive caching
- **Extensible and Plug-and-play!**
  - Easily add new datasets, tasks, evaluation metrics and model providers

LLMeBench: Technical Overview



Large Scale Experimentation Across:

TASKS	DATASETS	EVALUATION	MODELS
Word Segmentation, Syntax & Information Extraction (e.g., POS tagging)	XNLI	Accuracy	GPT-3.5
Factuality, Disinformation & Harmful Content Detection (e.g., Hate Speech & Propaganda Detection)	XGLUE	F1	GPT-4
Semantics (e.g., Semantic Textual Similarity and Natural Language Inference)	XQuAD	Macro-F1	BLOOMZ
Demographic & Protected Attributes (e.g., Gender and User Country Detection)	ASAD	Micro-F1	
Sentiment, Stylistic & Emotion Analysis (e.g., Stance Detection, Sarcasm Detection)	Aqmar	Weighted-F1	
Machine Translation (e.g., English-Arabic and Arabic dialects)	SANAD	BLEU	
News Categorization	MADAR	WER	
Question Answering	QASR	Pearson Correlation	
	WikiNews	Jaccard Similarity	
	Conll2006		
	ANERcorp		

LEARNING

- Zero-shot
- Few-shot

LLMeBench

A Complete Video Tutorial

https://rb.gy/6m6h2b

Language Model Evaluation Harness

A framework to evaluate LLMs on a large number of tasks and datasets

- Over 60 standard academic benchmarks for LLMs, with hundreds of subtasks and variants implemented.
- Support for models loaded via transformers (including quantization via AutoGPTQ), GPT-NeoX, and Megatron-DeepSpeed, with a flexible tokenization-agnostic interface.
- Support for fast and memory-efficient inference with vLLM.
- Support for commercial APIs including OpenAI, and TextSynth.
- Support for evaluation on adapters (e.g. LoRA) supported in HuggingFace's PEFT library.
- Support for local models and benchmarks.
- Evaluation with publicly available prompts ensures reproducibility and comparability between papers.
- Easy support for custom prompts and evaluation metrics.

https://github.com/EleutherAI/lm-evaluation-harness

A framework for few-shot language model evaluation (Gao et al., 2023)

Language Model Evaluation Harness

Pros

- Does not require explicit prompting
- Evaluation is based on log-likelihood
- Good for fast evaluation of LLMs

Cons

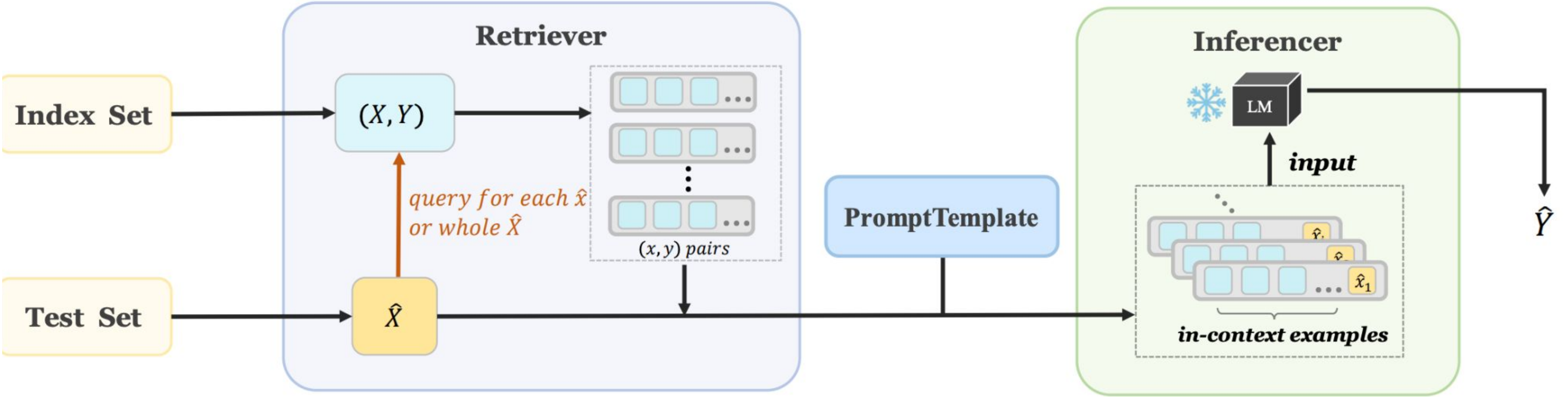
- Evaluation is not based on token(s) to represent candidate answer
- Lack of chat-templates

https://github.com/EleutherAI/lm-evaluation-harness

A framework for few-shot language model evaluation, (Gao et al., 2023)

Open ICL

An easy-to-use and extensible in-context-learning (ICL) framework for zero-/few-shot evaluation of LLMs



- Random
- Heuristic method (BM25, TopK, VoteK)
- Model based approach

- Tokens in candidate answer
- Perplexity

https://github.com/Shark-NLP/OpenICL

OpenICL: An Open-Source Framework for In-context Learning, (Wu et al. ACL, 2023)

Open ICL

Features

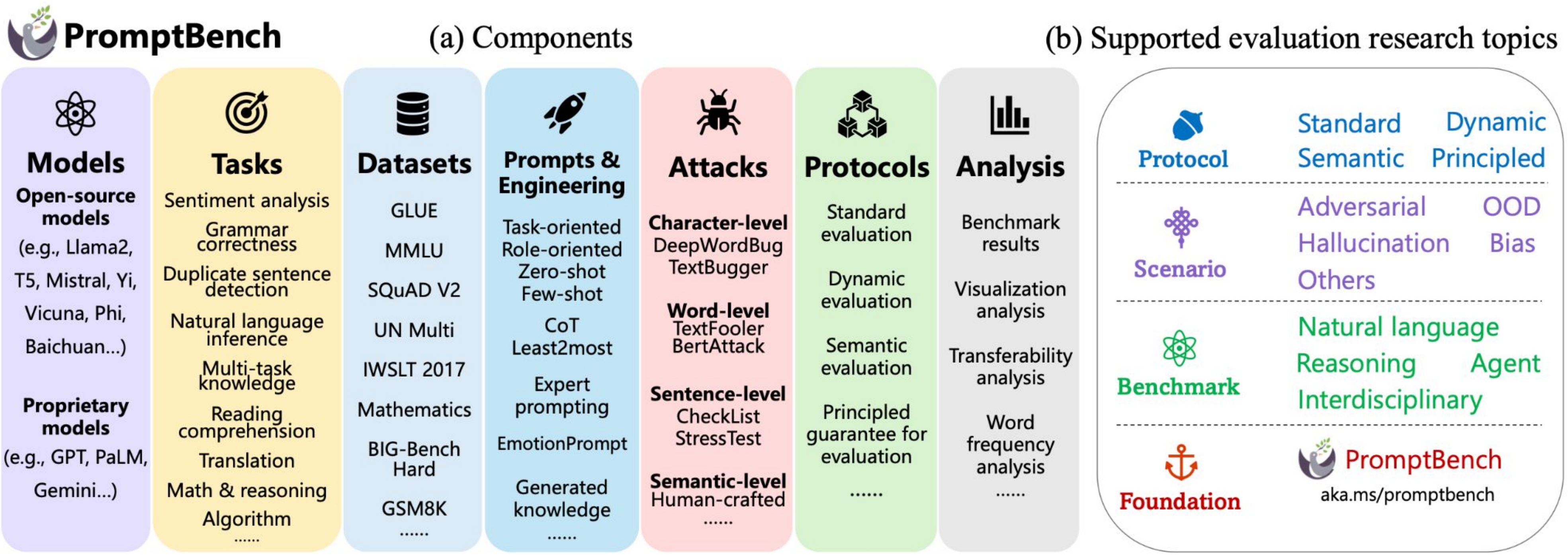
- Supports many state-of-the-art retrieval methods
- A unified and flexible interface for the development and evaluation of new ICL methods
- Implements data parallelism to improve the performance of both the retrieval and inference steps
- Model parallelism that users can easily parallelize their models with minimal modification to the code.

https://github.com/Shark-NLP/OpenICL

OpenICL: An Open-Source Framework for In-context Learning, (Wu et al. ACL, 2023)

Prompt Bench

A Unified Library for Evaluating and Understanding LLMs.
A comprehensive benchmark designed for assessing the robustness of LLMs to adversarial prompts



https://github.com/microsoft/promptbench

PromptBench: A Unified Library for Evaluation of Large Language Models, (Zhu et al, 2023)

Prompt Bench

Features

- Quick model performance assessment
- Prompt Engineering
- Evaluating adversarial prompts
- Dynamic evaluation to mitigate potential test data contamination

https://github.com/microsoft/promptbench

PromptBench: A Unified Library for Evaluation of Large Language Models, (Zhu et al, 2023)

LLM-as-a-Judge

MT-bench is a challenging multi-turn question set designed to evaluate the conversational and instruction-following ability of models

- 80 high-quality, multi-turn questions
- automated evaluation pipeline based on GPT-4

```
[System]
Please act as an impartial judge and evaluate the quality of the response provided by an
AI assistant to the user question displayed below. Your evaluation should consider factors
such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of
the response. Begin your evaluation by providing a short explanation. Be as objective as
possible. After providing your explanation, please rate the response on a scale of 1 to 10
by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]
```

prompt for single answer grading

Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, (Zheng et al, 2023)

QA

Thank You

https://llm-low-resource-lang.github.io/