INF367 25H: Selected Topics in Artificial Intelligence

Diamonds and Rust in the AI Treasure Chest

Dario Garigliotti - UiB - 27.10.2025

Plan for today

- Selected concepts on Large Language Models (LLMs)

- Using this survey article as reference:

  - Matarazzo et al. *A Survey on Large Language Models with some Insights on their Capabilities and Limitations*. 2025. https://arxiv.org/abs/2501.04040

Large Language Models: An intro

(Section 1 of the survey)

LLMs

Intro

- A scientific and industrial revolution
  - approach natural language processing tasks, achieving comprehension, learning, and generation levels that were once considered unattainable
- Application on a variety of **tasks** beyond text generation
  - E.g. question answering, language translation, and summarization
  - In general-purpose and domain-specific (healthcare, finance, law…) applications
- Emergent **abilities** at complex tasks beyond text generation
  - E.g. commonsense reasoning, code generation, arithmetic operations

LLMs

Intro

- Key **factors** driving the evolution of LLMs:
  - Exponential growth in available **data**
  - Exponential growth in **computational resources**
  - Advanced neural network **learning techniques**

- LLMs have introduced **new challenges** and raised **critical questions** about their applicability, limitations, and potential

Large Language Models: Overview

(Sections 2.1, 2.2 of the survey)

Language Models (LMs)

History and families

- **LLMs** are designed to comprehend, learn, and generate coherent and contextually relevant language at large scale.
- Historically, the development of **Language Models (LMs)** seeks to understand and replicate human language
- Four main **stages or families of LMs** can be identified:
  - Statistical Language Models
  - Neural Language Models (NLMs)
  - Pre-trained Language Models (PLMs)
  - Large Language Models (LLMs)
  - Also, Small Language Models (SLMs)

LMs
SLM

- **Statistical Language Models (SLM):**
  - Were developed to capture the statistical properties of language
    - E.g. such as word frequencies and co-occurrences
  - to predict the likelihood of a word sequence based on Markov assumption
  - Are limited by the exponential number of transition probabilities to be estimated and the Markov assumption, which may not always hold true in natural languages

LMs
NLM

- **Neural Language Models (NLM):**
  - Utilised neural architectures to capture language's complex patterns and dependencies
    - E.g. recurrent neural networks (RNNs) and long short-term memory (LSTM) networks
  - Could capture long-range dependencies and contextual information, enabling them to generate coherent and contextually relevant text

LMs
PLM

- **Pre-trained Language Models (PLM):**
  - Trained on large data corpora in an unsupervised or self-supervised way…
  - …before being fine-tuned on specific tasks
  - I.e., ***pre-training and fine-tuning*** paradigm:
    - To pre-train a model on a diverse data set…
    - and then transfer its knowledge to a narrower task by fine-tuning it on a smaller, task-specific dataset

LMs

LLM

- **Large Language Models (LLM):**
  - Characterised by their immense scale and complexity
  - Many LLMs are built on the transformer architecture, designed to capture long-range dependencies and contextual information in language
  - LLMs, especially those based on transformers, are **bidirectional**:
    - they consider the context of preceding and following words, enhancing their language understanding

LLM Application Tasks

- LLMs find applications across various domains or **tasks**, for:
  - **Text Generation:** Producing coherent and contextually relevant text
  - **Question Answering:** Answering questions based on provided context
  - **Language Translation:** Translating text from one language to another
  - **Summarization:** Creating concise summaries of longer texts
  - **Sentiment Analysis:** Determining the sentiment expressed in a text

LMs
SLM

- **Small Language Models (SLM):**
  - Are designed to provide efficient natural language processing (NLP) capabilities
  - Operate with a fraction of the parameters used by LLMs, which allows them to function in resource-constrained environments
  - Are gaining traction due to their adaptability and efficiency
    - E.g. DistilBERT, Gemma, Minstral

LLMs

Emergent abilities

- LLMs “show” (there’s debate about whether they do) **emergent abilities**
  - Their typical case is the ability…
    - to perform tasks for which the LLM was not explicitly trained, such as translation, summarisation, and question answering,
    - and to generalise to new tasks and domains, such as **{zero, one, few}-shot learning** (see also Fig. 34 for depictions of these learning techniques)

LLMs

Emergent abilities

- Three typical examples of emergent abilities are
  - **In-context learning**
  - **Instruction following**
  - **Step-by-step reasoning**
- Again, the LLM has “not” being trained to learn any of these tasks

LLMs

Scaling Law

- The **scaling law** says that as language models increase in size, their capabilities and performance on linguistic tasks also grow
- As LLMs scale up in terms of parameters
  - (tens or hundreds of billions, or even trillions),
- …LLMs demonstrate an unprecedented ability to generalise from diverse datasets and generate contextually coherent text, across a spectrum of language-related tasks
- It requires significant computational resources (processing power and memory)

Multi-task learning in the context of LLMs

- Within **multi-task learning**, all NLP tasks are-framed as a unified text-to-text problem, where every task is cast as generating text from input text
    - This approach simplifies using a single model across diverse tasks, encouraging a more generalised understanding of language

- Google's T5 is a pioneer model in using this key technique

Multi-task learning in the context of LLMs

T5

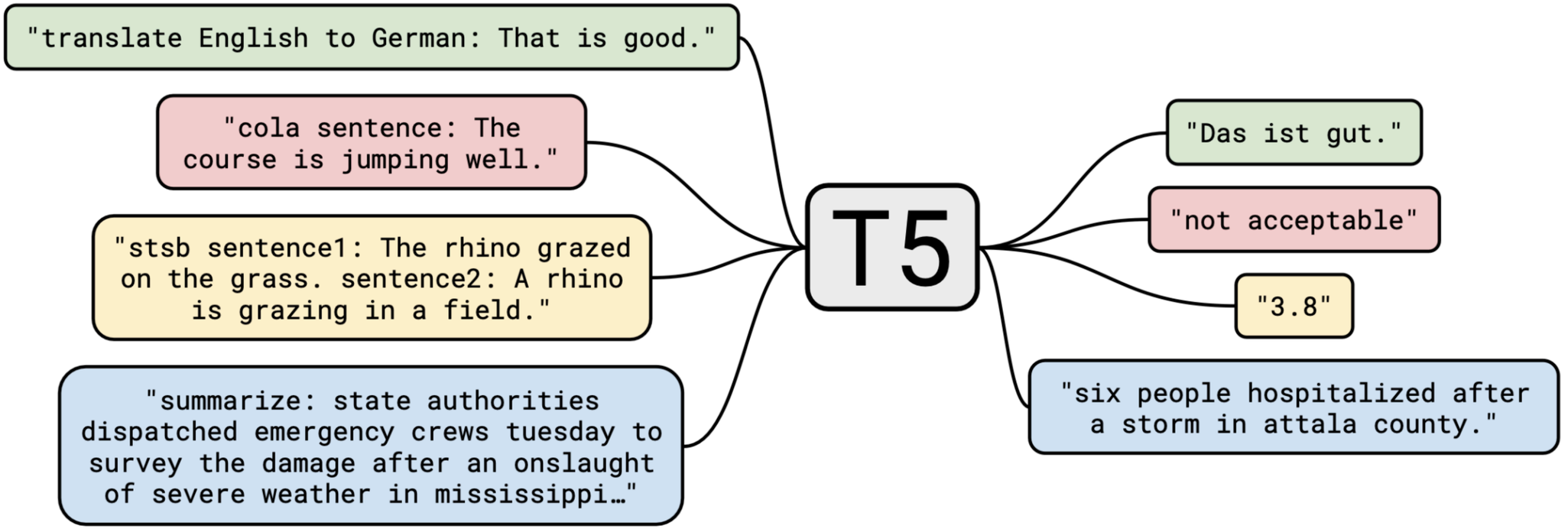


Figure 5: *A diagram of the T5 text-to-text framework. Every task – including translation, question answering, and classification – is cast as feeding the model text as input and training it to generate some target text. This approach allows the same model, loss function, hyperparameters, etc., to be used across diverse tasks. Source: Raffel et al. [99].*

Large Language Models: Foundations

(Sections 3.1-3.4 of the survey)

Pre-training

- **Pre-training** constitutes a foundational phase in developing LLMs
  - It allows the model to capture the relationships between words and generate coherent and contextually relevant text, laying the groundwork for its subsequent performance on specific NLP tasks
- It involves training a language model on a vast corpus of text data…
  - before **fine-tuning** it on a smaller, task-specific dataset, such as text generation or text classification, to improve its performance on that task

Pre-training

Unsupervised pre-training

- **Unsupervised pre-training** trains a model on a large corpus of text data *without* labels or annotations
- The model is trained to predict the next word, given the previous words in the sequence
- By **Autoregressive Language Modeling (ALM)**, the model is trained to predict the probability distribution over the next word in the sequence given the previous words in the sequence in a *unidirectional* manner
- BERT and its variants, instead, employ a **masked language model (MLM)** objective, where random words in a sentence are masked, and the model is trained to predict these masked words based on their context, *bidirectionally*

Pre-training

Supervised pre-training

- **Supervised pre-training** trains a model on a large corpus of text data *with* labels or annotations
- Models learn representations more closely aligned with the end tasks
- LLMs are exposed to a vast array of labelled data across various domains
    - Supervised pre-training involves teaching the model to predict the correct output given an input

Pre-training

Semi-supervised pre-training

- **Semi-supervised pre-training** blends the strengths of supervised and unsupervised learning methodologies

- Various **techniques** underpin semi-supervised pre-training in LLMs;
  - **Self-training**
  - **Consistency regularization**
  - **Transductive learning**
  - **Inductive learning**
  - **Active learning**

- It builds on **certain assumptions** about the underlying structure and distribution of the data, e.g.,
  - **Cluster Assumption**
  - **Continuity Assumption**
  - **Low-Density Separation Assumption**

Data sources

- LLMs strongly depend on extensive, high-calibre **data for pre-training**:

- **General Data**
  - **Webpages**
  - **Conversation text**
  - **Books**
- **Specialized Data**
  - **Multilingual text**
  - **Scientific literature**
  - **Code**

Data preprocessing

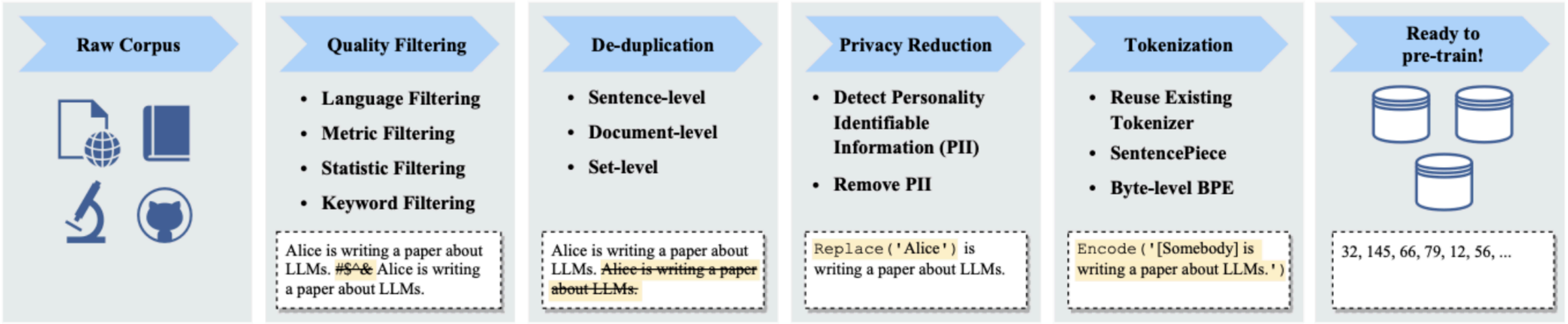


Figure 17: *Common data preprocessing steps for training Large Language Models (LLMs). Source: Zhao et al. [364].*

LLM Adaptation

- The **adaptation of LLMs** is a critical aspect of their deployment in real-world applications
  - It enables the models to be fine-tuned on specific tasks or domains after pre-training,
  - …enhancing their performance by minimizing the loss of generalization capabilities
- Adaptation can be achieved through various techniques, such as
  - **instruction tuning** (to enhance or release LLMs' abilities)
  - and **alignment tuning** (to align LLMs' behaviours with human preferences)

LLM Adaptation

Instruction tuning

- **Instruction tuning** leverages natural language instructions to fine-tune pretrained LLMs
  - It enhances LLMs' ability to follow and comprehend natural language instructions
- *Unlike* traditional fine-tuning, which adapts models to specific tasks, instruction tuning employs a more generalized approach that broadens the model's utility across a variety of tasks through an "instruction-following" paradigm

LLM Adaptation

Instruction tuning

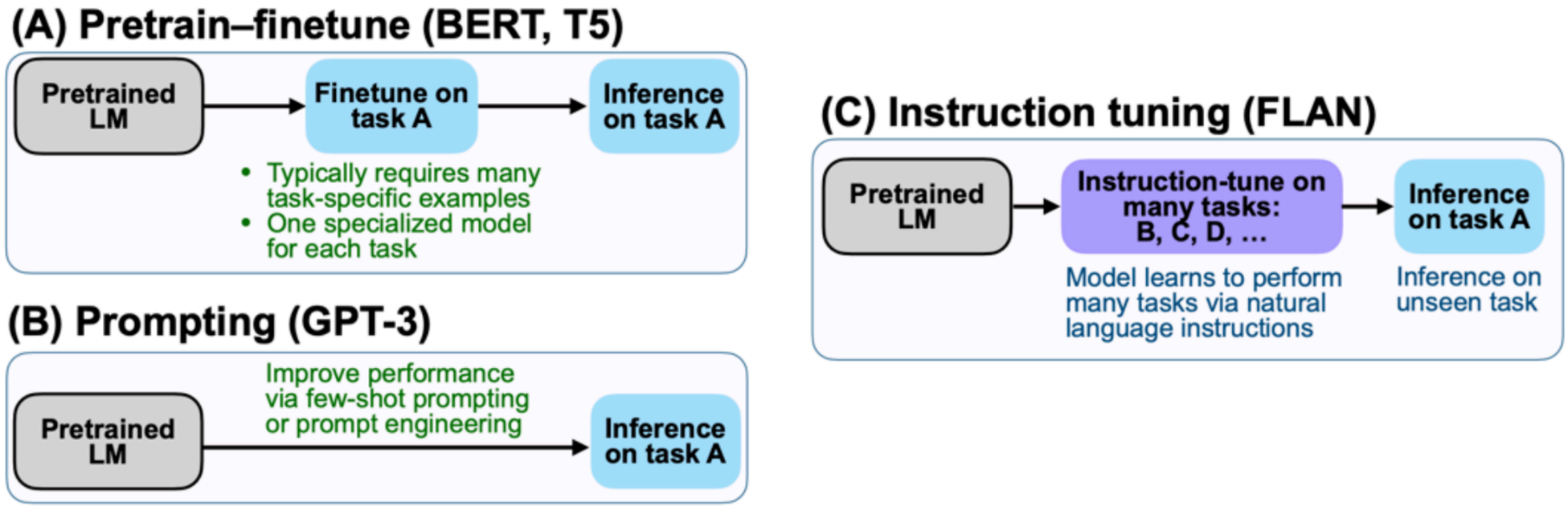
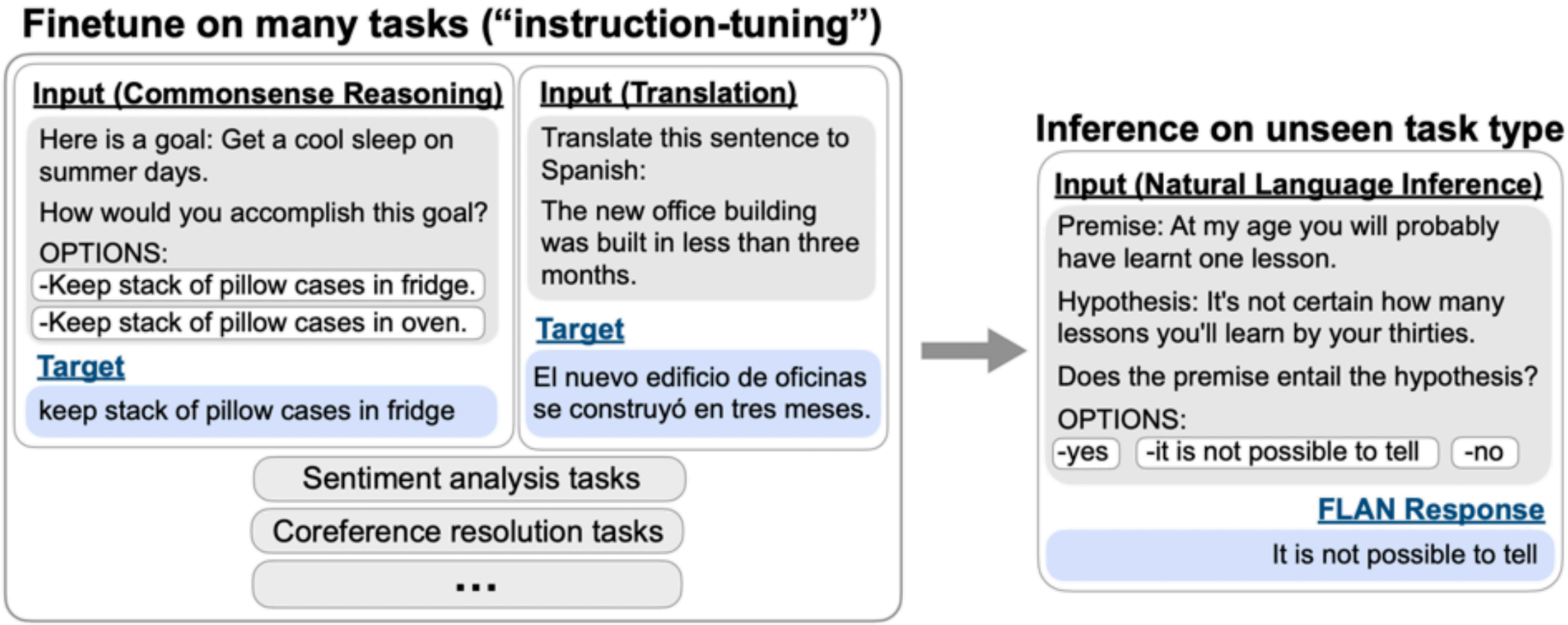


Figure 18: *Overview of instruction tuning. Source: Zhao et al. [364].*

LLM Adaptation

Instruction tuning



LLM Adaptation

Instruction tuning

- Two **essential factors** for the instance construction are:
  - **Scaling the instructions**: Increasing the number of tasks within training data can significantly improve the generalization ability of LLMs
  - **Formatting design**: Related to ICL; related to CoT
- Three **main approaches** to construct instruction-formatted instances
- Several **additional strategies** to improve the instruction tuning process

LLM Adaptation

Instruction tuning

- The **main effects** of instruction tuning are:
  - **Performance Improvement:** Instruction tuning significantly enhances LLMs
  - **Task Generalization:** Instruction tuning endows LLMs with the capability to understand and execute tasks based on natural language instructions
  - **Domain Specialization:** As LLMs often lack the domain-specific knowledge required (for fields like medicine, law, and finance), instruction tuning facilitates the transformation of general-purpose LLMs into domain-specific experts

LLM Adaptation

Instruction tuning

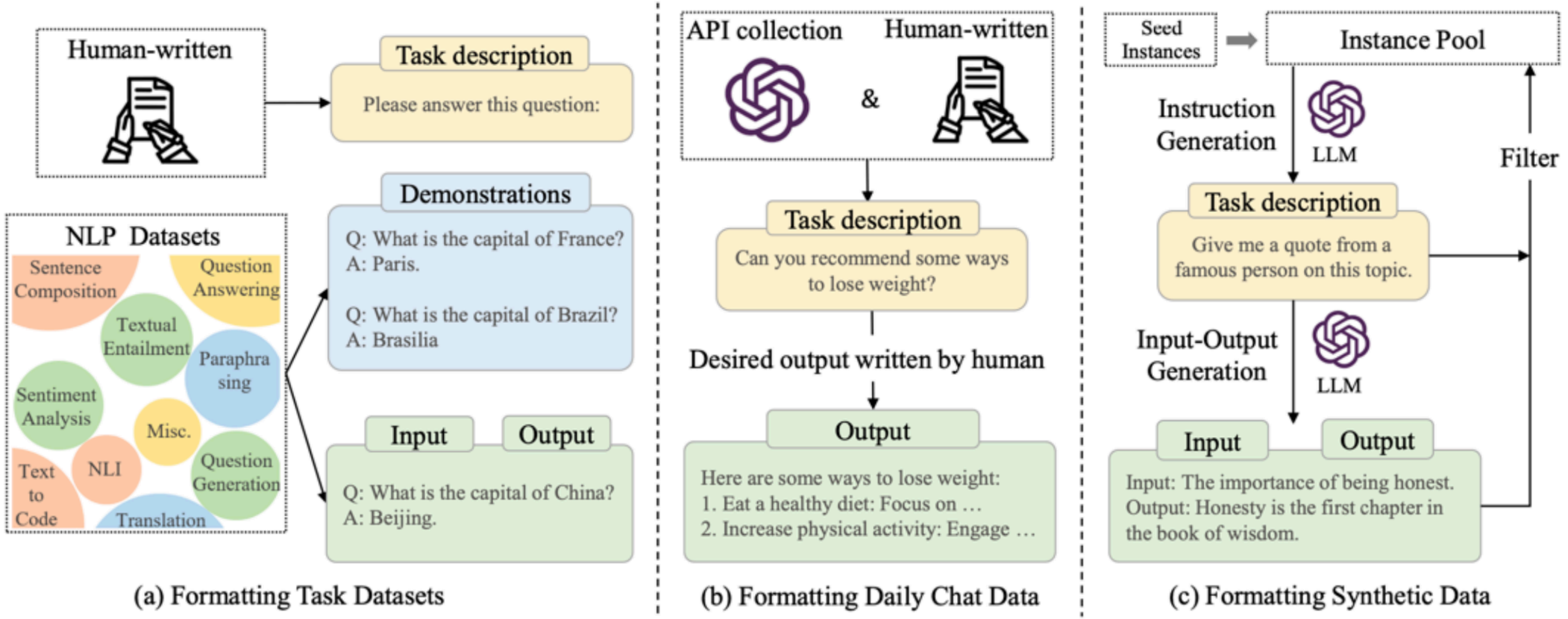


Figure 21: *Three main approaches to construct instruction-formatted instances. Source: Zhao et al. [364].*

LLM Adaptation

Alignment tuning

- LLMs may sometimes generate outputs inconsistent with human values or preferences (e.g., fabricating false information, pursuing inaccurate objectives, and producing harmful, misleading, or biased content)
- To avoid such undesirable outcomes, **alignment tuning** ensures that LLMs' outputs align with specified ethical guidelines or desired behaviours
- Unlike pre-training and fine-tuning (which focus on optimizing model performance), alignment tuning aims to optimize the model's behaviour to conform to human values and norms
- Three **primary criteria for regulating** the behaviour of LLMs are helpfulness, honesty, and harmlessness

Large Language Models: In-Context Learning

(Section 4.1 of the survey)

In-Context Learning (ICL)

- **ICL** is a prompting technique to allow the model learn from the context of the prompt
- ICL consists of:
  - **the task description and/or a few examples** of the task as demonstrations
    - combined in a specific order to form natural language prompts with designed templates
  - and, finally, **the test instance** appended to the prompt, to form the input for LLMs to generate the output
- ICL's performance heavily relies on demonstrations, to be designed properly in prompts

In-Context Learning (ICL)

In-Context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

Traditional fine-tuning

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

```
1  sea otter => loutre de mer          ← example #1
```

↓

gradient update

↓

```
1  peppermint => menthe poivrée        ← example #2
```

↓

gradient update

↓

• • •

↓

```
1  plush giraffe => girafe peluche     ← example #N
```

gradient update

```
1  cheese =>                           ← prompt
```

In-Context Learning (ICL)

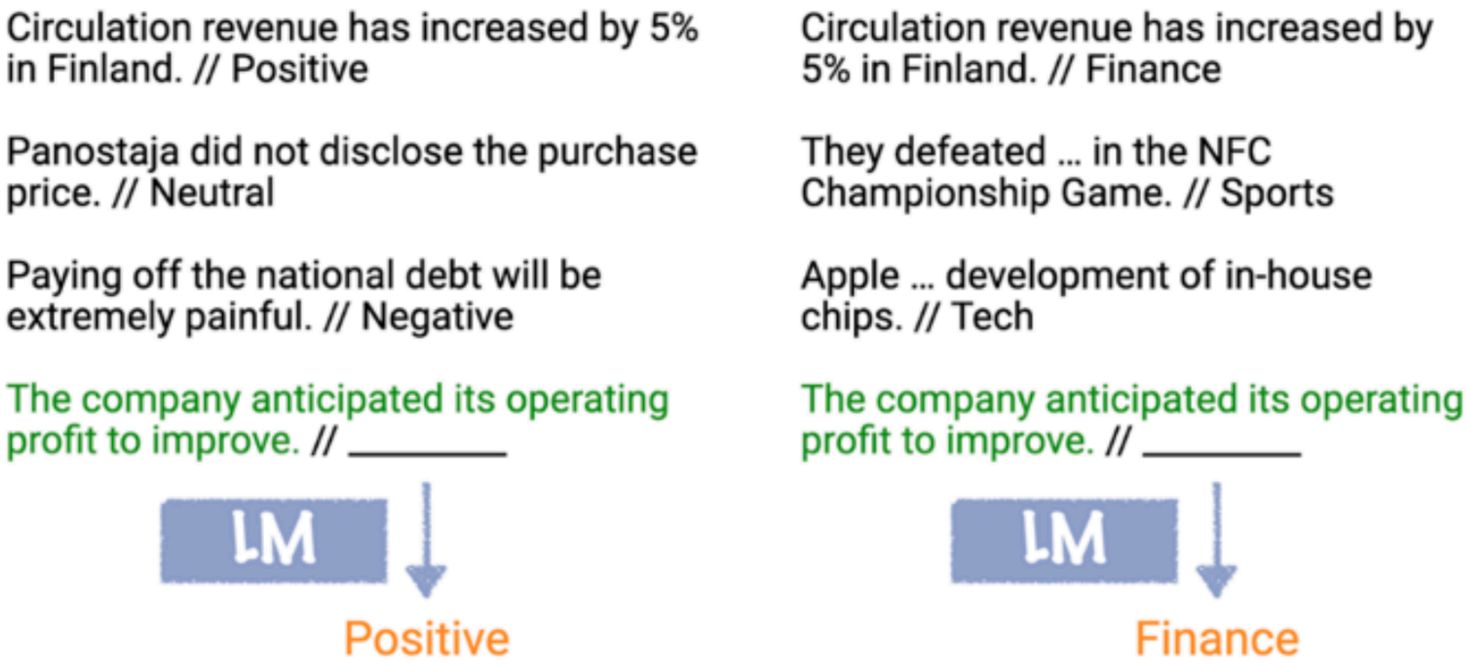


Figure 1: *Two examples of in-context learning, where a language model (LM) is given a list of training examples (black) and a test input (green) and asked to make a prediction (orange) by predicting the next tokens/words to fill in the blank. Source: Lab [288].*

Large Language Models: Chain-of-Thought

(Section 4.2 of the survey)

Chain-of-Thought (CoT)

- **Chain-of-Thought (CoT)** prompting is an enhanced strategy developed to augment the performance of LLMs on complex reasoning tasks such as arithmetic, commonsense, and symbolic reasoning
  - CoT integrates intermediate reasoning steps within the prompts, providing a more structured path towards the solution
  - CoT can be considered a special case of ICL
- CoT is considered by many as an emergent ability, that suddenly appears and greatly enhances the performance of LLMs when they reach a certain scale

Chain-of-Thought (CoT)

CoT vs. ICL

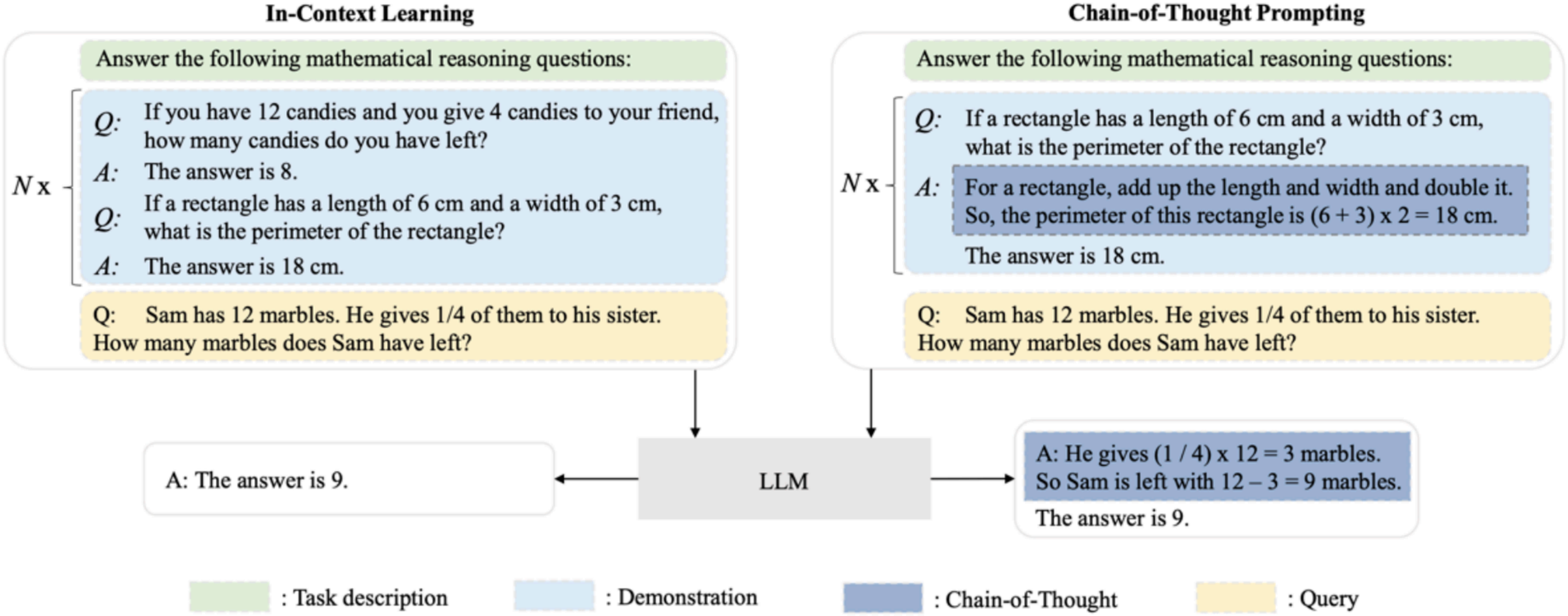


Figure 38: *A comparative illustration of in-context learning (ICL) and chain-of-thought (CoT) prompting. ICL prompts LLMs with a natural language description, several demonstrations, and a test query, while CoT prompting involves a series of intermediate reasoning steps in prompts. Source: Zhao et al. [364]*

Chain-of-Thought (CoT)

CoT + ICL

- CoT can be effectively combined with In-context Learning in these settings:
  - **Few-shot CoT**: CoT augments standard input-output pairs with intermediate reasoning steps. The design of CoT prompts is crucial; incorporating diverse and complex reasoning paths has shown to boost LLM performance
  - **Zero-shot CoT**: *Unlike* its few-shot counterpart, **zero-shot CoT** does not rely on annotated demonstrations. Instead, it generates reasoning steps directly from a prompt, significantly improving performance when scaled to larger models

Chain-of-Thought (CoT)

CoT Paradigms

- Two major **CoT paradigms**:
  - **Zero-Shot-CoT** is a task-agnostic paradigm that generates reasoning steps directly from the prompt, eliminating the need for annotated CoT datasets, adding a single prompt like *“Let’s think step by step”* after the test question to facilitate the reasoning chains in LLMs
  - **Manual-CoT** uses manually designed demonstrations one by one, which can be expensive and time-consuming to create
- The other paradigm is **few-shot prompting with manual reasoning** demonstrations one by one. Each demonstration has a question and a reasoning chain that comprises a rationale (a series of intermediate reasoning steps) and an expected answer

Chain-of-Thought (CoT)

CoT Paradigms

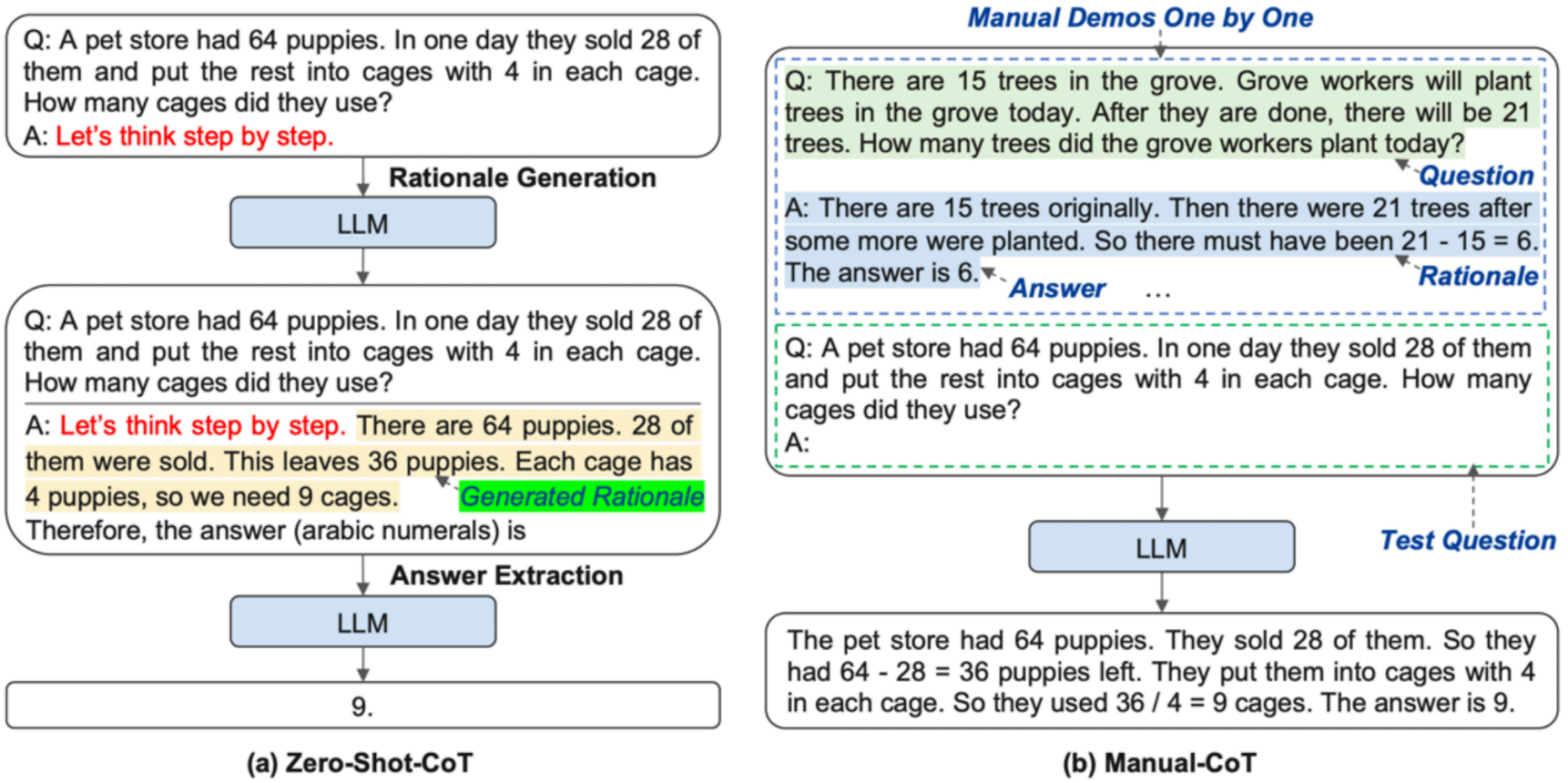


Figure 42: *Zero-Shot-CoT [285] (using the "Let's think step by step" prompt) and Manual-CoT[230] (using manually designed demonstrations one by one) with example inputs and outputs of an LLM. Source: Zhang et al. [243]*