ANALYSIS AND COMPARISON OF EXISTING AI TECHNOLOGY

ROLEPL-AI

Project funded by the European Commission within the ERASMUS+ programme under the agreement n° 2023-1-FR01-KA220-VET-000157570

Deliverable 2.2 – Version 1.1

Type of Activity		
IO	Intellectual Output	**X**
A	Project Management and Implementation	
M	Transnational Project Meeting	
E	Multiplier Event	

Nature of the deliverable		
	Feedback from participants	
	Direct effect on participants and project partners	
	Practical & reusable resources for the practitioners	
	Research material bringing forward the reflexion in the sector	**x**
	Community building tools	
	Partnerships and Cooperation	
	Dissemination material	
	Organizational and working documents	

Dissemination Level		
PU	Public	**X**
CO	Confidential, only for members of the consortium (including the Commission Services)	

ACKNOWLEDGEMENT

This report forms part of the deliverables from a project called "ROLEPL-AI" which has received funding from the European Union's ERASMUS+ programme under grant agreement No. 2023-1-FR01-KA220-VET-000157570. The Community is not responsible for any use that might be made of the content of this publication.

This project aims at training soft skills remotely, by pushing the practice through the implementation of AI-based simulation.

The project runs from September 1st, 2023, to August 31st, 2025 (24 months), it involves 5 partners (Manzalab, Manzavision and Inceptive, France; VUC Storstrøm, Denmark; Fachhochschule Dresden, Germany) and is coordinated by Manzalab.

List of participants

Participant No.	Participant organisation name	Acronym	Country
1 (coord)	Manzalab	MZL	France
2	Manzavision	MZV	France
3	Inceptive	ICV	France
4	VUC Storstrøm	VUC	Denmark
5	Fachhochschule Dresden	FHD	Germany

CONTENT

Abbreviations

[AI] Artificial Intelligence
[API] Application Programming Interface
[BLEU] bilingual evaluation understudy
[CNN] Convolutional Neural Network
[CRI] Consistent Role Identity
[ICL] In Context Learning
[GPU] Graphics Processing Units
[HH] Helpful and Harmless
[LLM] Large Language Model
[LoRA] Low-Rank Adaptation
[LSTM] Long short-term memory
[MoE] Mixture of experts
[NLM] Neural language models
[NLP] Natural Language Processing
[NPC] Non-player character
[PEFT] Parameter-Efficient Fine-Tuning
[PII] Personally Identifiable Information

[PLM] Pre-trained language models
[PPO] Proximal Policy Optimization
[PTQ] Post-Training Quantization
[QAT] Quantization-aware training
[RK] Role Knowledge
[RL] Reinforcement Learning
[RLHF] Reinforcement Learning with Human Feedback
[RM] Reward Model
[ROUGE] Recall-Oriented Understudy for Gisting Evaluation
[SFT] Supervised fine-tunning
[SLM] Statistical language models
[UQR] Unknown Question Rejection

1 INTRODUCTION

1.1 OVERVIEW

This deliverable has two objectives. Firstly, to provide a large overview of the state of the art of AI models in language comprehension and modelling. From 2020, major advances are underway in this field, with Large Language Models (LLM). This document focuses on this technology.

Secondly, we aim to provide technical recommendations on the methodology to adapt an LLM in the context of ROLEPL-AI.

1.2 DELIVERABLE POSITIONING

D2.2 is based on the state of the art and Inceptive knowledge on IA and more specifically on LLM. It is developed at the beginning of the ROLEPL-AI project before any experimentation.

Its conclusions, among those of D2.1 "Review of the status of research in AI and education" are connected to task 2.3 "Recommendations for use of AI in education and ALTAI self-assessment" within Work Package 1. The conclusions of this document will drive the methodology used to produce D4.3 "Training the AI for the simulation with pedagogical content created in D3.2".

Finally, with this extensive state of the art, we aim to provide a "small reference book" about LLMs to our project partners.

1.3 DELIVERABLE STRUCTURE

The deliverable is structured as follows:

- In section 2, a large overview of LLM state of the art as it is in February 2024 is provided.
- In section 3, a short overview of published works on LLM used in Roleplay is presented.
- In section 4, we estimate the resources available for ROLEPL-AI in terms of data and computing power. Then, an evaluation of different approaches to adapt an LLM into the context of ROLEPL-AI is performed.
- In section 5, we conducted an empirical study to choose the best available open LLM model to be the starting point of the project adaptation.

2 LLM STATE OF THE ART

With the release of ChatGPT in November 2022 (OpenAI, 2022), the generative IA became mainstream. Chatbots, were no longer seen as these "stupid popups" that do not understand anything and can only be interacted with buttons. A new efficient way to access information, interact with it, and more generally perform tasks in natural language such as summarizing text, writing code or roleplay has come to the general public.

This section presents the technology behind ChatGPT, the LLMs. It heavily relies on the very extensive survey "A survey of large language models" by Zhao (2023). We have summarized and completed some aspects to present here the general state of the art around LLMs. The objective of this state of the art is not to be extensive. We aim to provide the basics of literature around LLMs to successfully implement them into the ROLEPL-AI project.

This section is structured as follows:

- First, we present the state of LLMs in NLP techniques.
- Second, we provide an overview of the deep learning architecture used in LLMs.
- Third, an overview of datasets used to train LLMs is presented.
- Fourth, we present different workflows used to create, adapt and exploit an LLM.
- Fifth, we dive into the nature of LLMs' abilities.

2.1 BACKGROUND AND EVOLUTION OF LLMS

2.1.1 What is an LLM?

Zhao (2023) defines it simply as "transformer language models that contain hundreds of billions (or more) of parameters, which are trained on massive text data." Some representatives are GPT-3 (Brown, 2020), PALM (Chowdhery, 2023), Galactica (Taylor, 2022) and Llama (Touvron, 2023a).

Transformer is a deep learning architecture intended to process sequences of data (like temporal series, natural language, etc) as a whole, focusing only on the most relevant parts (Vaswani, 2017).

LLMs process language transforming it into a set of minimal units, called tokens. Their length can vary but it is usually around 4 letters and 75% of a word in English (Radford, 2019).

LLMs work by taking a series of input tokens and outputting another series of tokens. One of the most common approaches is the GPT model series: given a series of tokens, output the most likely following token (Radford, 2019).

One of the aspects of LLMs is that increasing the size of the model, the amount of trained data and the computing time increases model capacity (technically, reduces the model loss) (Radford, 2019, Brown 2020, Chowdhery 2023). This phenomenon has been described in literature under the **scaling laws**. KM scaling law (Kaplan, 2020) describes the model performances given three factors: computing time, data and model size. Chinchilla scaling law (Hoffmann, 2023) provides a more detailed value of model size and training data needed for a given computer time.

Once the performances increase, LLMs gain a series of abilities not trained for. This is described in the literature as emergent abilities (Wei, 2022b), and are described as "the abilities that are not present in small models but arise in large models". Some typical abilities are:

- **In-Context Learning:** (Brown, 2020) Assuming that the language model has been provided with a natural language instruction and/or several task demonstrations, it can generate the expected output for the test instances by completing the word sequence of input text, without requiring additional training or model parameter adjustment.
- **Instruction following:** On a specific training set, including a mixture of multi-task dataset, LLMs are shown to perform well on unseen tasks that are also described in the form of instructions (Sanh, 2021; Ouyang 2022; Wei 2021).
- **Step-by-step reasoning:** For small language models, it is usually difficult to solve complex tasks that involve multiple reasoning steps (mathematical word problems, etc). In contrast, with the chain-of-thought (CoT) prompting strategy (Wei, 2022a), LLMs can solve such tasks by utilizing the prompting mechanism that involves intermediate reasoning steps for deriving the final answer.

In existing literature (Kaplan, 2020; Wei, 2022b; Hoffman, 2022) there is no clear relation between scaling laws and emergent abilities. But both give a perspective on the interest of bigger models over small ones. In general, scaling law describes predictable performance relation with the potential effect of diminishing returns, while emergent abilities are unpredictable but very profitable once such abilities actually emerge.

2.1.2 Evolution of language models

Natural language is a prominent ability in human beings to express and communicate, which develops in early childhood and evolves over a lifetime (Pinker, 2014; Hauser, 2014). But for computers, understanding, processing and communicating with natural language is a very challenging task.

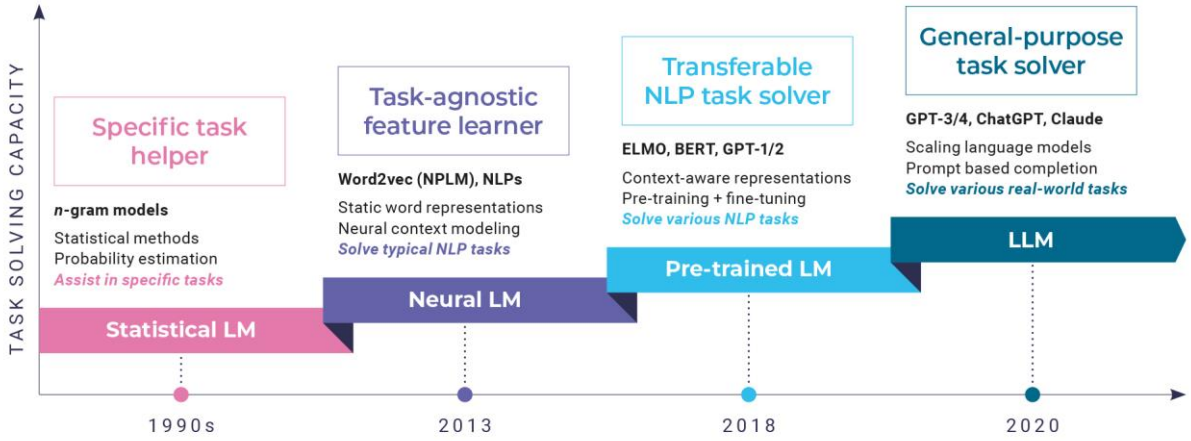
During the last 30 years, we can distinguish four main stages to face these challenges and model language:

- **Statistical language models (SLM)**: (Jelinek, 1998; Gao, 2004; Rosenfeld, 2000). These models, mostly developed in the 90s, are based on statistical learning methods. These methods assumed that the probability of the next word was conditioned by the previous ones (Markov assumption). SLMs have been widely applied to enhance task performance in information retrieval (Liu, 2005; Zhai, 2008) and natural language processing (NLP) (Thede, 1999; Bahl, 1989; Brants, 2007). However, they often suffer from the curse of dimensionality: it is difficult to accurately estimate high-order language models since exponential number of transition probabilities need to be estimated.
- **Neural language models (NLM)**: (Bengio, 2000; Mikolov, 2010; Kombrink, 2011) These models used neural networks (multi-layer perceptron and recurrent neural networks) to learn the probability distributions of words. Many improvements were made on the way that words were represented by the models (Bengio, 2000; Mikolov, 2013a; Mikolov, 2013b) leading to big improvements in the field of NLP and initiated the use of language models for representation learning.
- **Pre-trained language models (PLM)**: ELMo (Peters, 2018) was proposed to capture context-aware word representations by first pre-training a bidirectional long short-term memory (LSTM) network (instead of learning a word representations) and then fine-tuning the biLSTM network according to specific downstream tasks. Based on the highly parallelizable Transformer architecture (Vaswani, 2017) with self-attention mechanisms, BERT (Devlin, 2018) was proposed by pre-training bidirectional language models with specially designed pre-training tasks on large-scale unlabeled corpora. These pre-trained context-aware word representations are very effective as general-purpose semantic features, which have largely raised the performance bar of NLP tasks. Following this paradigm, a great number of studies on PLMs have been conducted, introducing either different architectures (Lewis, 2019; Fedus, 2022) (e.g., GPT-2 (Radford, 2019) and BART (Lewis, 2019), or improved pre-training strategies (Liu, 2019; Sanh, 2021; Wang, 2022c).
- **Large language models (LLM)**: Researchers find that scaling PLM often leads to an improved model capacity on downstream tasks (following the scaling law (Kaplan, 2020). A number of studies have explored the performance limit by training an ever-larger PLM. Although scaling is

mainly conducted in model size (with similar architectures and pre-training tasks), these large-sized PLMs display different behaviours from smaller PLMs and show surprising abilities in solving a series of complex tasks.

Figure 1 illustrates these evolutions.

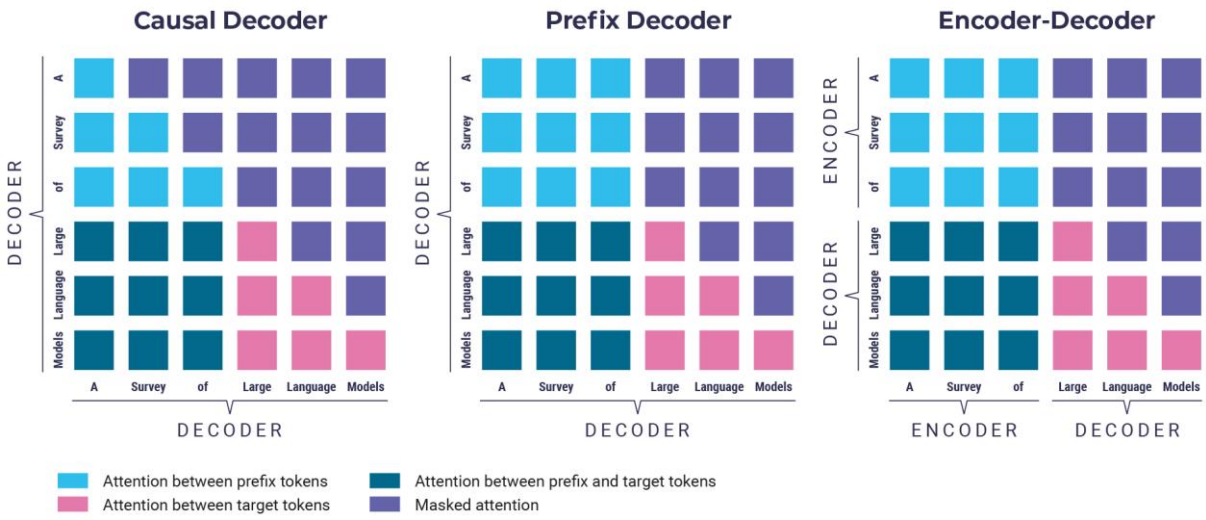
Figure 1: Evolution of the four generations of language models, figure by Zhao (2023), redesigned.



2.2 LLM ARCHITECTURE

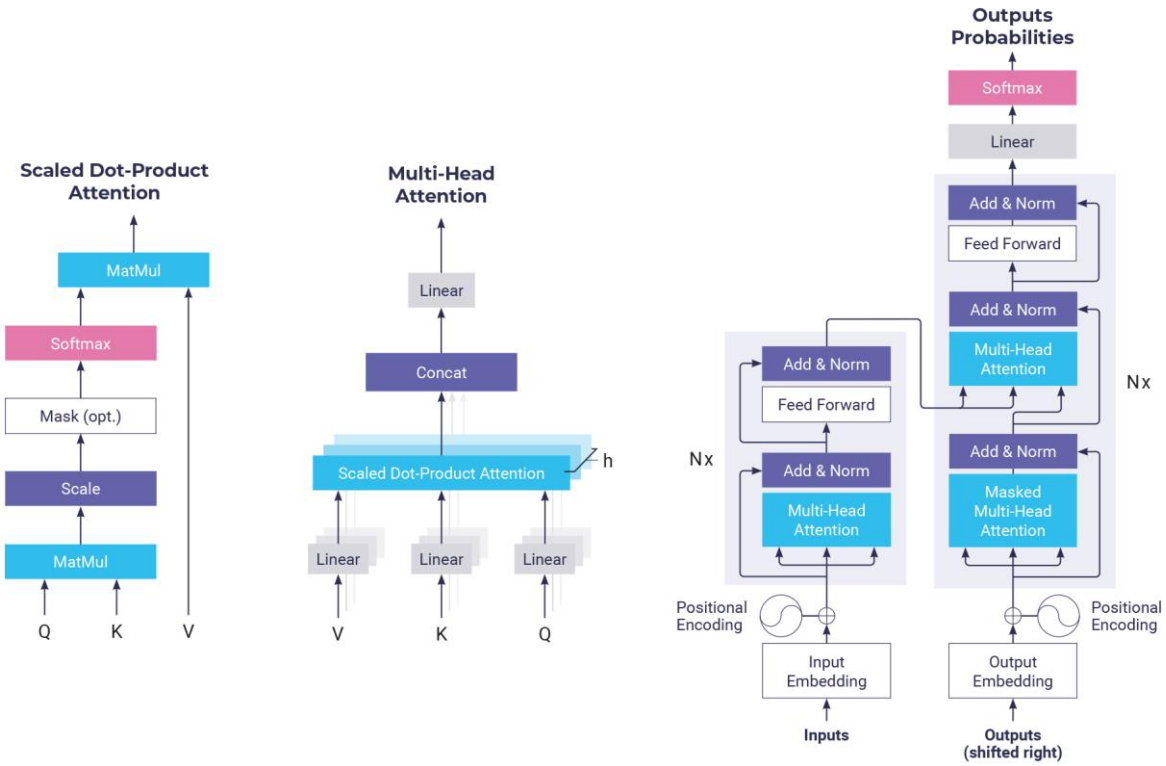
As defined in section 2.1, LLM architecture is based on transformer deep learning architecture. There are 4 mainstream variants of this architecture (Zhao, 2023), which had small differences based on how tokens are masked. These differences are shown in figure 2.

Figure 2: A comparison of the attention patterns in three mainstream architectures. Figure by Zhao (2023), redesigned.



- **Encoder-decoder** architecture: The first transformer architecture (Vaswani, 2017) proposed two stacks of transformer blocks as the encoder and decoder, respectively. The encoder adopts stacked multi-head self-attention layers to encode the input sequence for generating its latent representations, while the decoder performs cross-attention on these representations and autoregressively generates the target sequence. An overview of the architecture is in figure 3. This architecture has become mostly rare on new model releases.
- **Causal decoder** architecture: In this decoder-only architecture, both input and output use the same decoder. To ensure each input token can only attend to the past tokens itself, a unidirectional attention mask is used. GPT models (Radford, 2018, Radford, 2019, Brown 2020) are developed based on the causal-decoder architecture.
- **Prefix decoder** architecture: A revision of causal decoders to enable a bidirectional attention over prefix tokens and unidirectional over generated ones.
- **Mixture of experts (MoE)** architecture: In this architecture variant, the dense layer of the transformer is replaced by several dense layers, but only a subset is executed per token. MoE is a flexible way to scale parameters while maintaining a constant computational cost. But training these architectures is more complex and unstable due to the complex, hard-switching nature of the routing operation. Mixtral (Jiang, 2024) is an excellent example of this architecture. With only 47B of parameters and 13B of active parameters, they achieve the same performance as GPT-3.5 with 175B parameters in several benchmarks. Also, there has been speculation that GPT-4 uses this architecture.

Figure 3: Overview of the classical transformer architecture. Note that multi-head attention is composed of h scaled dot-product attention in parallel. On the right, an example of encoder-decoder architecture.



Currently, most LLMs models follow the causal decoder architecture, as it seems to achieve better zero-shot and few-shot performance (Wang, 2022c). Moreover GPT-3's success (Brown, 2020) has shown the possibilities of a large causal decoder model.

But as pointed by Zhao (2023), we still lack comparative and extensive research on other architectures.

2.3 DATASETS

According to Zhao (2023), "compared with small-scale language models, LLMs have a stronger demand for high-quality data for model pretraining, and their model capacities largely rely on the pretraining corpus and how it has been preprocessed".

In this section we review the dataset basics for LLMs pretraining phase. But the elements present here must be taken into consideration when building a dataset for other stages of LLM training and adaptation.

Among the corpus used to train LLMs, there are two types of data:

- **General Text Data**: As shown by Zhao, 2023, most LLMs use general-purpose data for pre-training, such as webpages, books, and conversational text, etc.

Webpages are excellent to make LLM gain diverse linguistic knowledge and enhance their generalization capabilities (Radford, 2019; Raffel, 2020). But webpage data quality is very heterogenous and needs proper filtering. Conversation text enhances the conversation competence of LLMs (Zhang, 2022a), improving their performance on question answering tasks. But excessive conversation data may lead the model to interpret declarative instructions and direct interrogatives as the beginning of conversations, leading to inadequate answers (Zhan, 2022). Books provide an important source of formal long texts, which are potentially beneficial for LLMs to learn linguistic knowledge, model long-term dependency, and generate narrative and coherent texts.

- **Specialized Text Data**: These data are useful to improve LLMs abilities on downstream tasks.
  The integration of multilingual corpus can enhance the multilingual abilities of language understanding and generation, leading to an improvement in translation, multilingual summarization and multilingual question answering (Scao, 2022; Chowdhery, 2023). Scientific text can improve performance in scientific and reasoning tasks (Saier, 2023). These texts are provided from scientific books, arXiv papers, etc. But general LLMs usually struggle with mathematical symbols or protein sequences. A specific tokenization technique and preprocessing is usually required.
  Code, in the form of Q&A from programming forums or public software repositories can be used to teach LLMs to write quality code (Chen, 2021) and answer programming questions (Li, 2022a). Also, code might be source of complex reasoning abilities on LLMs (Fu, 2022).

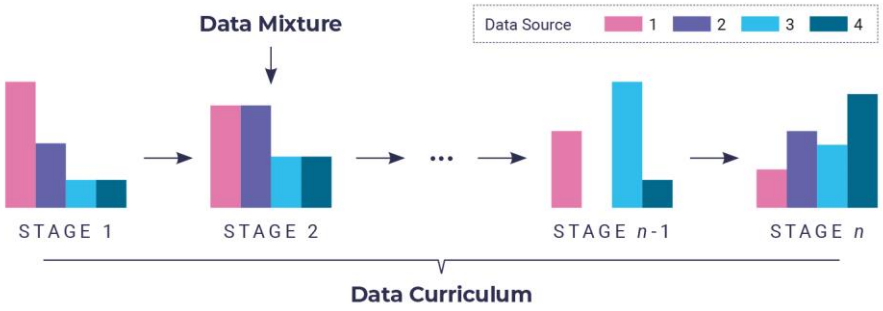
To ensure we have high quality data, it is essential to preprocess the data for constructing the pre-training corpus, especially removing noisy, redundant, irrelevant, and potentially toxic data (Chowdhery, 2023; Rae, 2021; Longpre, 2023b), which may largely affect the capacity and performance of LLMs. Some studies (Longpre, 2023b; Raffel, 2020; Du, 2022) made comparison between cleaned and uncleaned data asserting the performance increase on cleaned data. Moreover, data duplication may decrease training stability and degrade LLMs capacity to use in-context information (Hernandez, 2022).

These are the general steps for data cleaning:

- **Quality Filtering**: There are two approaches to removing low quality entries from the dataset:
  - Classification: (Du, 2022; Brown, 2020; Chowdhery, 2023) Usually, a binary classifier is trained with well-curated data (ex Wikipedia) as positive instances and sample candidate data as negative instances. The prediction score measures the quality of the entry. Some

studies (Du, 2022; Rae, 2021) find that this approach can lead to a bias as it removes good quality samples containing dialectal, colloquial, and sociolectal languages.
    - Heuristic: (Rae, 2021; Scao 2022) Some other works delete entries based on a set of rules. These rules are usually based on language (keep only some languages), metric based (evaluate metrics like perplexity), statistical (based on punctuation, character distribution, etc) and keyboard based (removing noising or non-useful elements in text like HTML tags).
- **De-Duplication**: This can be done at sentence level (delete repeated words to prevent repetitive patterns), at document level (word or n-gram overlapping) and at dataset level (preventing dataset contamination, i.e. parts of the test set are found on the training set). (Chowdhery, 2023; Carlini, 2022).
- **Privacy Reduction**: Some data sources include personally identifiable information (PII), which is a risk of privacy data breaches (Carlini, 2021). Rule based methods like keyword spotting, are effective to delete names, phone numbers and addresses.

Figure 4: An illustration of data scheduling. Figure by Zhao (2023), redesigned.



With the diverse sources of data needed to train an LLM, the way to schedule this data is very important. There are two key aspects of data scheduling:

- **Data Mixture**: The proportion of each data source. As each data source type is related to certain LLM capabilities, achieving a correct mix is important. The data mixture is generally set on a global level and can be also locally set to varied proportions at different training stages.
- **Data curriculum**: The order in which each data source is scheduled for training. It has been shown that, in some cases, to learn a certain skill, learning in a skillset sequence (e.g., basic skills → target skill) outperforms direct learning from a corpus focused solely on the target skill (Chen, 2023d; Rozière, 2023). So usually, data curriculum starts with easy/general examples and progressively introduce more challenging/specialized ones.

Figure 4 shows an illustration of these two aspects.

2.4 WORKFLOW TO CREATE, ADAPT AND USE LLM

In this section we present the whole (most common) workflow to build an LLM from scratch. We will cover from pretraining, training an empty network to be a language model, to a ready to use aligned model. Section 2.4.1 presents the whole workflow from pretraining to RLHF (Reinforcement Learning with Human Feedback). Section 2.4.2 presents alternative ways beside RLHF to adapt LLMs "in a more economical way". Finally, section 2.4.3 presents different prompting strategies. Figure 5 presents a partial view of this workflow.

2.4.1 Model training

- **LLM pretraining**

To pretrain an LLM, a series of datasets are selected and scheduled as explained in section 2.3. Usually, these datasets range from ~100 billion tokens (PALM2, (Anil, 2023), Galactica, (Taylor, 2022)) to 2 or 3 trillion tokens (LLama2, 2T (Touvron, 2023b) Skywork, 3.2T (Wei, 2023b)). During the batch training, a series of tokens are shown to the model that produces an update. This update is compared with the expected output and the loss is computed. Then the parameters are updated by backpropagation. These are the parameters commonly used on these stages and their usual values:

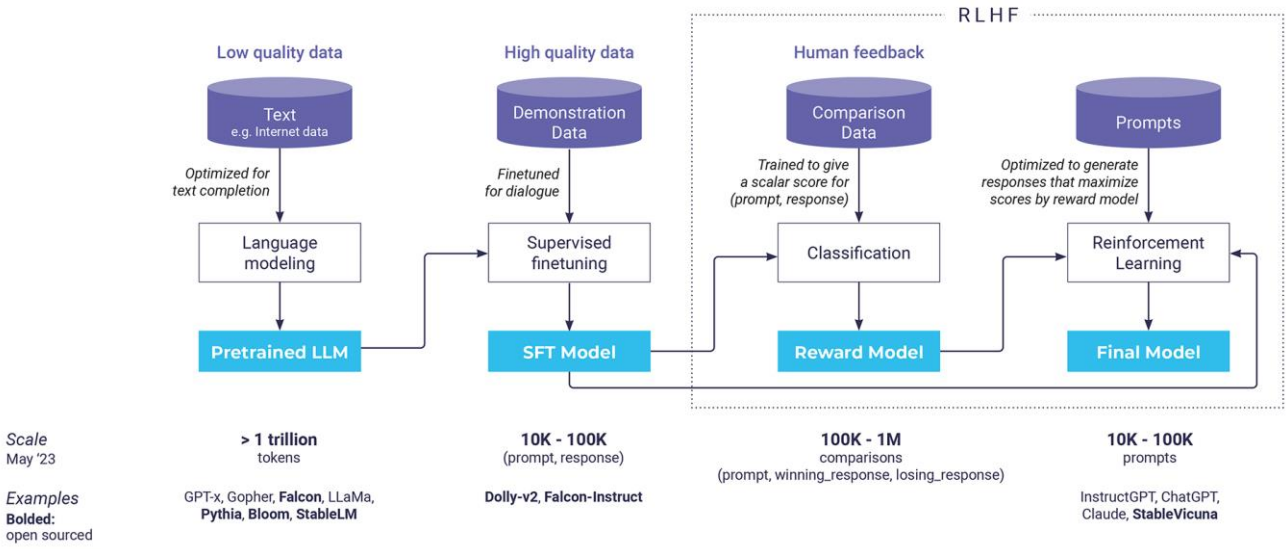
- **Batch training**: Batch sizes are set to large numbers (e.g. 2,048 examples or 4M tokens) to ensure training stability. Usually, batch size is gradually increased as studies have shown it can increase training stability (Chowdhery, 2023).
- **Learning rate**: A similar strategy is adapted with learning rate. In the initial 0.1% to 0.5% of the training steps, a linear warm-up schedule is employed for gradually increasing the learning to a target usually between 5e-5 and 1e-4. Then a cosine decay strategy is used to reduce to 10% of the target value until convergence.
- **Optimizer**: Adam (Kingma, 2014) and its variant AdamW (Loshchilov, 2017) are the main choices with these hyperparameters: $\beta 1 = 0.9$, $\beta 2 = 0.95$ and $\epsilon = 1e{-}8$.
- **Stabilizing the training**: LLM suffers from stability, leading sometimes to an increase in the loss values. Techniques like gradient clipping (value 1.0) and weight decay (value 0.1) have been commonly used (Brown, 2020; Scao, 2022, Zhang, 2022a; Zeng, 2022; Smith, 2022). Other models, like PaLM (Chowdhery, 2023) and OPT (Zhang, 2022a) simply restart the training before a loss spike. GLM (Zeng, 2022) finds that this situation

comes from abnormal gradients of the embedding layer and simply shrinks the layer gradients.

Zhao (2023) points out that "as the model and data sizes increase, it has become challenging to efficiently train LLMs under a limited computational resource. Especially, two primary technical issues are required to be resolved, i.e., increasing training throughput and loading larger models into GPU memory". We will not review these techniques, as they will not be useful in the ROLEPL-AI context.

Once a LLM is pre-trained, it has some general abilities. But these abilities can be improved, especially if we want the LLM to do specific tasks. There are two major approaches we will present in the next section. Instruction tuning aims to enhance or unlock specific abilities of the LLM. Alignment tunning aims to align the behaviour of LLM with human preferences.

Figure 5: An illustration of the LLM training and adaptation workflow.



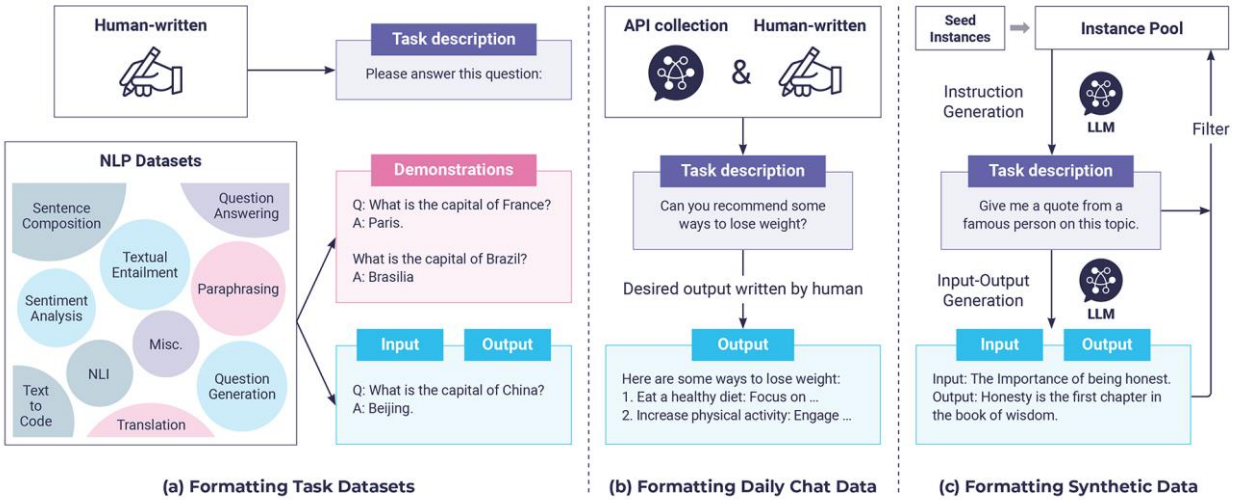
- **Instruction tuning**

Instruction tuning is the approach of fine-tuning a pre-trained LLM on a collection of formatted instances in natural language (McKenzie, 2022). After this process, LLMs show superior abilities on unseen tasks (Sanh, 2021; McKenzie, 2022; Rae, 2021; Zhang, 2021).

According to Zhao (2023), to build the dataset for instruction tunning, three main approaches are commonly used. They are summarized in Figure 6.

- **Formatting NLP Task Datasets**: Datasets coming from classical NLP tasks (e.g., text summarization, text classification, and translation) can be adapted for instruction tuning (Sanh, 2021; Ouyang, 2021; Wei, 2021, Wang, 2022b). Human annotators add a human-written task description to each entry, as instruction description is a crucial factor in LLM capacity to generalize to unseen tasks (Wei, 2021). Some datasets are even augmented, inverting input-output pairs (e.g, ask a model to generate a question for a given answer) (Sanh, 2021; Tang, 2022b; Longpre, 2023).
- **Formatting Daily Chat Data**: Despite being a large source, NLP datasets lack instruction diversity and mismatch real human needs (Ouyang, 2022). InstructGPT (Ouyang, 2022) proposes to take the queries that real users have submitted to the OpenAI Application Programming Interface (API) as the task descriptions. Additionally, to enrich the task diversity, human labellers are also asked to compose the instructions for real-life tasks (open ended generation, open question answering, brainstorming, chatting, etc.). Then, they let another group of labellers directly answer these instructions as the output. Finally, they pair one instruction (i.e., the collected user query) and the expected output (i.e., the human-written answer) as a training instance.
- **Formatting Synthetic Data**: To reduce the cost of generating a dataset, semi-automated approaches (Wang, 2022a) use existing LLMs. Self-Instruct method only needs 175 instances as the initial task pool. Then, they randomly select a few instances from the pool as demonstrations and prompt a LLM to generate new instructions and corresponding input-output pairs. After quality and diversity filtering, they are added to the pool. This method is an economical way to generate a dataset, but the generated content may be simplistic and lacking diversity.

Figure 6: An illustration of instance formatting and three different methods for constructing the instruction-formatted instances. Figure by Zhao (2023), redesigned.



Scaling the number of tasks can largely enhance the generalization ability of LLMs (Sanh, 2021; Wei, 2021; Wang, 2022b). With the increase of the task count, the model performance initially shows a continuous growth pattern, while the gain becomes negligible when it reaches a certain level (Wang, 2022b; Chung, 2022). Moreover, the design of natural language format also highly impacts the generalization performance of LLMs (Wang, 2022b). Adding examples to task (few-shot) can lead to improvements and need less instruction engineering (Chung, 2022). But, incorporating other components (e.g., things to avoid, reasons, and suggestions) may have a negligible or even adverse effect (Wang, 2022b; Mishra, 2021).

Finally, diversity and quality of instructions seem to be more important than the number of instances (Zhou, 2023a). However, large amount of training may compensate for the absence of high-quality data (Chen, 2023a)

Training in instruction tuning is different to pretraining (Chung, 2022). As a small dataset is used and this process is closer to supervised training. The training objective (usually sequence-to-sequence loss) and configuration (smaller batch size, learning) are different. These are some key aspects:

Balancing data distribution: It is important to balance the proportion of different tasks during fine tuning. As some datasets are quite large, a collection of ~1 000 to ~10 000 instances are taken from each dataset (Wei, 2021; Chung, 2022)

Combining Instruction Tuning and Pre-Training: According to Iyer, 2022, it is often suitable to integrate pre-training data during instruction tuning, to stabilize the training. Inversely, GLM-130B (Lai, 2022) and Galactica (Taylor, 2022) include instruction tuning instance on pre-trained dataset.

Multistage instruction tuning: In addition to mixing chat and instruction data, a usual approach (Yulan-Chat-Team, 2023) is to first show instruction data, and then chat instructions, with some instruction data.

Studies have shown that all scale of models (from 77M of 540B parameters) improve their performances thanks to instruction tuning (Chung, 2022; Longpre 2023a). It improves the way that an LLM generalizes to unseen tasks (Chung, 2022) or specific human instructions (Wei, 2022b), reduces weakness of LLMs (repetitive answers, not finishing a task) (Ouyang, 2022; Chung, 2022) and improves multilingual task performing, even with English-only instructions (Muennighoff, 2022). It also provides an excellent way to adapt LLMs to specific domains, like law (Huang, 2023), finance (Wu, 2023), etc.

- ### Alignment tuning

LLMs can exhibit unintended behaviours, like fabricating false information, pursuing inaccurate objectives, and producing harmful, misleading, and biased expressions (Ouyang, 2022; Kenton, 2021). To fix this, alignment tuning is proposed to make LLMs act with human values. As these values are quite subjective, this approach is totally different from instruction tuning, and may impact in a negative way the LLMs performance (Askell, 2021).

These human values are:

- **Helpfulness**: The LLM should help users solve their tasks or answer questions concisely and efficiently.
- **Honesty**: Aligned LLMs should present accurate content to users instead of fabricating information. It is also crucial for the LLM to convey appropriate degrees of uncertainty in its output.
- **Harmlessness**: The LLM is required not to produce an output that can be offensive or discriminatory. It also should decline requests for malicious purposes.

As these criteriums are clearly subjective, a human labelling team is necessary. There are several approaches to collect human feedback:

- **Ranking-based approach**: An Elo system makes feedbackers compare many outputs ensuring that the diversity of criteria that may exist between labellers does not impact the result (Glaese, 2022).
- **Question-based approach**: Instead of just ranking the best answers, labellers are asked to give feedback to researchers by answering certain

questions designed by researchers (Nakano, 2021), to know if the model is providing correct information.
- **Rule-based approach**: The feedbackers must follow a specific set of rules to decide what answer is the best (according to alignment criteria) (Glaese, 2022).

The main algorithm to align LLMs with collected feedback is Reinforcement Learning from Human Feedback (RLHF) (Christiano, 2017; Ziegler, 2019). RLHF employs reinforcement learning (RL) algorithms (e.g., Proximal Policy Optimization (PPO) (Schulman, 2017)) to adapt LLMs to human feedback by learning a reward model. Such an approach incorporates humans in the training loop for developing well-aligned LLMs. InstructGPT (Ouyang, 2022) represents an excellent example of RLHF implementation.

The algorithm is summarized in Figure 7. The RLHF key steps are:

1. **Supervised fine-tuning**: First, a dataset with desired behaviours is collected (often generated by human labellers (Ouyang, 2022)) and the model is fine-tuned as for instruction tuning.
2. **Reward model training**: Once the model is ready, the human labellers annotate a series of outputs of the model (as described before in this section). Then a model (usually a smaller LLM, (Ouyang, 2022)) is trained to predict the labellers preference.
3. **Reinforcement Learning fine-tuning:** The LLM is trained in an RL problem setting with PPO (Ouyang, 2022). Some mechanisms, like KL divergence, ensure that the model does not deviate too much from the initial model outputs.

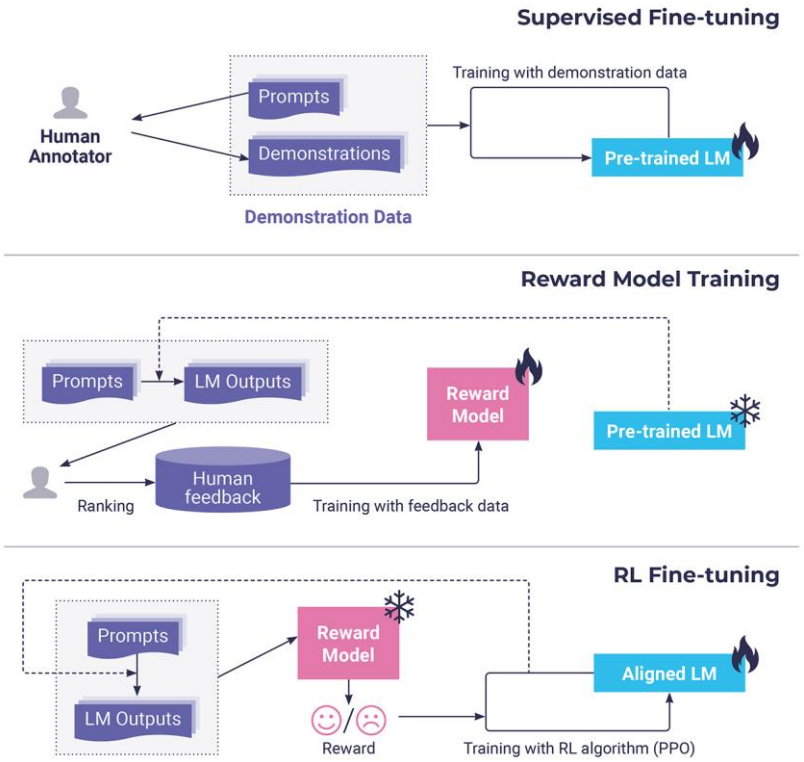
In practice, experience has shown that the reward model should be of the same size or larger than the base model (Touvron, 2023b). Moreover, as there are 3 alignment criteria, it can be useful to have one model per criterion (Touvron, 2023b).

Besides RLHF, there are other alternatives for alignment, directly relying on LLM fine-tuning with supervised learning (SFT) on a high-quality alignment dataset. In these cases, data collection can be done through different methods:

- **Reward model-based approaches**: Use a reward model to select aligned responses, and form a dataset (Dong, 2023)
- **LLM based generative approaches**: Use an already aligned LLM to generate the responses (Bai, 2022)
- **LLM based interactive approaches**: Use LLM interactions, one with others to generate the dataset and provide feedback and improvement. (Liu,

2023g)

Figure 7: The workflow of RLHF algorithm. Figure by Zhao (2023), redesigned.



After collecting the dataset, the model is aligned with supervised fine-tuning. As the objective and the content is different from a pretraining setting or an instruction tuning, some differences exist in the training setting. First, a cross-entropy loss for sequence to sequence is usually used, with several variants to include alignment factors (Lu, 2022a; Rafailov, 20203). Other studies propose auxiliary losses to better capture the nature of the problem (Yuan, 2023b; Zhang, 2023b).

According to Zhao (2023), RLHF and SFT approach are two different ways to train an LLM. While RLHF, as a RL approach, leads the model to learn the "align policy", SFT is closer to "imitation learning". For more theoretical analysis on imitation learning and reinforcement learning, please refer to the related RL literature (Hussein, 2017; Levine, 2022).

In a more specific way, SFT boosts model performance on various benchmarks (Wei, 2021; Chung, 2022; Taori, 2023; Chiang, 2023) and "unlocks" LLM abilities. "Unlocks", but not injects, as LLM without the abilities (typically, small models) may simply lead to hallucinations (Schulman, 2023). As we said, SFT means

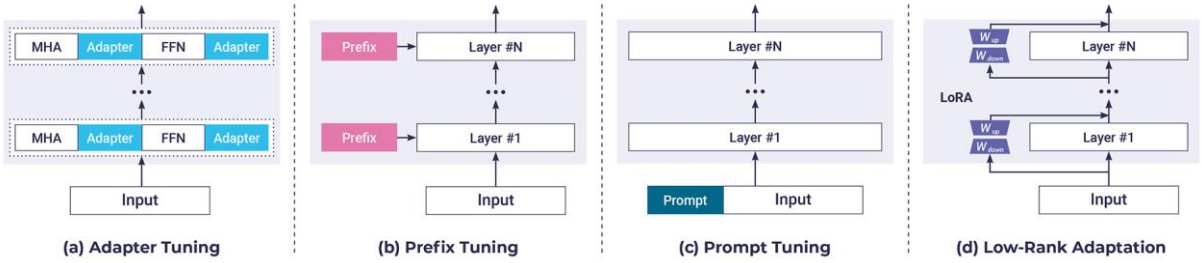
imitation data. If data is not of high quality and largely heterogenous, the performances boost is not as important (Touvron, 2023b).

On the other side, RLHF has really shown its capacity to mitigate harmful responses (Ouyang, 2022; Touvron, 2023b; Bai, 2022), in some studies even enhancing helpfulness and harmlessness at the same time (Touvron, 2023b). However, RLHF suffers from reinforcement learning drawbacks: sample inefficiency and training instability. RLHF heavily relies on a strong SFT model as the initial checkpoint. And human annotators are involved in a complex iterative optimization process, that can heavily impact model performances.

2.4.2 Efficient model adaptation & quantization

The presented methods in section 2.4.1 are supposed to adapt all the LLM parameters to perform a task. This is very expensive and requires a large amount of computation. In this section we present alternative more economical ways to adapt LLMs. We also present the technique of quantization, a trade-off between performances and the amount of computation needed to run an LLM.

Figure 8: An illustration of four different parameter-efficient fine-tuning methods. MHA and FFN denote the multi-head attention and feed-forward networks in the Transformer layer, respectively. Figure by Zhao (2023), redesigned.



- **Parameter efficient tuning**

Parameter-efficient fine-tuning (Hu, 2021; Li, 2021; Lester, 2021) aims to reduce the number of trainable parameters while retaining a good performance in LLMs. These techniques are:

- **Adapter Tuning:** Adapters are small neural network modules inserted into the Transformer architectures (Houlsby, 2019). The adapter architecture compresses the input, applies a function and then recovers the original dimension (Houlsby, 2019; Hu, 2023). Adapters can be inserted after the attention head and the feed forward layer (Houlsby, 2019; Hu, 2023) or in

parallel of these (parallel adapters, He (2021)). When training adapters, all parameters outside this module are frozen.

- **Prefix Tuning:** Prepends a sequence of prefixes, which are a set of trainable continuous vectors, to each Transformer layer (Li, 2021). These vectors can be viewed as virtual token embeddings and are task specific.
- **Prompt Tuning:** Incorporates a trainable prompt vector at the input layer (Lester, 2021; Liu 2023b). The prompt can be free or prefixed depending on the technique. But as only some input parameters will be trained, the performance highly relies on the model's underlying capacities (Lester, 2021).
- **Low-Rank Adaptation (LoRA):** During LoRA (Hu, 2021) training of a set of model parameters W, the model is adapted by adding a product of two smaller matrices A×B, where A and B are low-rank matrices. This addition is denoted as W+A×B. During training, only these small matrices A and B are updated, while the original weights W of the pre-trained model remain fixed. This significantly reduces the number of parameters that need to be trained, saving both GPU memory during training, and storage space. Beyond that, LoRA does not introduce more parameters to the model (as adapters) or reduce the size of the context (like prefix and prompt tunning) (Liu, 2021), making it a more flexible option than other alternatives.

An empirical study on efficient tuning showed that efficient tuning underperformed the baseline of GPT-3.5 on difficult tasks, while achieving similar results on simple tasks (Hu, 2023). Another study comparing LoRA, adapters and prefix tunning (Ding, 2023) on several NLP tasks concluded that all the techniques were globally less performant than fine-tuning, while LoRA was better than adapters which was better than prefix tunning, with adapters and LoRA being similar in convergence speed.

Quantization

Having a huge number of parameters, LLMs use a significant amount of memory. We explore the techniques of quantization, aiming to reduce the amount of memory used by models, and therefore, reduce the inference computation needs.

Quantization often refers to the mapping process from floating-point numbers to integers (Gholami, 2022). For neural networks, there are two types of parameters that can be quantized: weights (model parameters) and activations (hidden activations, functions between the layers).

The basic idea of quantization can be illustrated by this process to transform a floating number x into a quantized integer x_q.

1. Apply the function $x_q = R(x/S) - Z$ with S the scaling factor and Z the zero-point factor.
2. Dequantization can be done applying: $x_d = S \cdot (x_q + Z)$
3. Then we can estimate the dequantization error as $x_q - x_d$.

There are two main approaches to quantization:

- **Post-Training Quantization (PTQ)**: These techniques adapt LLMs weights after the network training.
    - *Mixed-precision decomposition:* Starting with models with 6.7B parameters, big values occur only on some hidden activations (Dettmers, 2022). A vector wise approach called *LLM.int8 separates* these values and quantifies them with a higher precision (16-bit floating numbers, while the others are quantified as 8-bit integers).
    - *Fine-grained quantization:* LLM values are organized in tensors. Some techniques adopt a specific quantization for each tensor (Xiao, 2023), but this leads to high reconstruction errors. Other approach tries to leverage this problem (Yao, 2022; Lin, 2023).
    - *Balancing the quantization difficulty:* Weights are easier to quantize than activation. SmoothQuant (Xiao, 2022) leverage this fact adjusting the scaling factors of quantization to account for this factor.
    - *Layerwise quantization:* This approach finds optimal quantized weights that minimize a layerwise reconstruction error. There are many techniques like GPTQ (Frantar, 2023) that make it feasible to quantize very large models in a 3- or 4-bit precision. AWQ (Lin, 2023) simplifies the optimization by incorporating activation-aware scaling for weights, like SmoothQuant (Xiao, 2022).
- **Training based quantization:**
    - *Efficient fine-tuning enhanced quantization:* PTQ approaches often have very poor results with low bit quantization (e.g., INT4 quantization). QLoRA (Dettmers, 2023b) incorporates additional small tuneable parameters (16-bit precision) into the quantized models, to achieve an efficient, high-precision model fine-tuning. The experiment shows that 4-bit quantized models can achieve the full 16-bit fine-tuning performance by QLoRA.
    - *Quantization-aware training (QAT) for LLM:* A study (Liu, 2023g) explores the effect of QAT methods using LLaMA. They show promising results on 4-bit quantization on weights but not on activations.

The quantization effects on models have been largely studied (Yao, 2023; Liu, 2023a). Here is a list of the main conclusions:

- **INT8 weight quantization often results in good performance, while lower precision depends on specific methods** (Xiao, 2023; Lin, 2023; Frantar, 2022; Yao, 2023). Also, with a fixed memory budget it is better to use a large model with lower quantization than a small one with higher quantization. For example, a 4-bit 60GB LLM is demonstrated to have better performance than an 8-bit 30GB LLM (Dettmers, 2023a).
- **Activations are more difficult to quantize than weights** (Xiao, 2022; Dettmers, 2022; Yao, 2023). As big models have significant outlier values on activations, this leads to higher quantization errors.
- **Efficient fine-tuning enhanced quantization is a good option to enhance the performance of quantized LLMs** (Hu, 2021; Dettmers, 2023b). This has two advantages. First, it can help to compensate the performance degradation from low-bit quantization (Yao, 2023; Liu, 2023a). Secondly, it provides lightweight adapted LLMs to a specific objective (Dettmers, 2023).

According to Zhao (2023) empiric experimentation quantization with LLaMa 13B and LLaMa 7B, performances seemed not to be reduced with an 8-bit or 4-bit quantization. But in practice, they advise "to first examine the performance of 4-bit weight quantization for LLMs if reducing memory usage is a critical consideration for deployment."

2.4.3 LLMs use

In this section we describe the different techniques to use an LLM.

- **Prompting**

Prompting is the major approach to use LLMs (Liu, 2023f) to solve diverse tasks.

The process of manually creating a suitable prompt is called **prompt engineering** (Liu, 2022; White, 2023). As the quality of the prompt can heavily influence the model performance (Liu, 2023f), there has been a lot of scientific works (White, 2023; Santu, 2023) and websites (OpenAI, 2023; Ai short, 2023; Awesome ChatGPT Prompts, 2023) that present prompt engineering in a very detailed way.

As a lot of work has been done, we present only here the basic guidelines for prompt engineering, and we reproduce a table of the survey Zhao (2023) survey with the core useful tips for prompt engineering in table 1.

Table 1, Zhao (2023). A collection of useful tips for designing prompts that are collected from online notes (White, 2023; Santu, 2023; OpenAI, 2023) and Zhao (2023) experiences. Principles are abbreviated as Prin. and list the IDs of the related principles for each prompt 1: expressing the task goal clearly; 2:

decomposing into easy, detailed sub-tasks; 3: providing few-shot demonstrations; 4: utilizing model-friendly format.

Ingredient	Collected Prompts	Prin
Task Description	T1. Make your prompt as **detailed as possible**, e.g., "*Summarize the article into a short paragraph within 50 words. The major storyline and conclusion should be included, and the unimportant details can be omitted.*"	1
	T2. It is helpful to let the LLM know that it is **an expert with a prefixed prompt**, e.g., "*You are a sophisticated expert in the domain of computer science.*"	1
	T3. Tell the model more **what it should do**, but not what it should not do.	1
	T4. To avoid the LLM to generate too long output, you can just use the prompt: "*Question: Short Answer:*". Besides, you can also use the following suffixes, "*in a or a few words*", "*in one of two sentences*".	1
Input Data	I1. For the question required factual knowledge, it is useful to **first retrieve relevant documents** via the search engine, and then **concatenate them into the prompt** as reference.	4
	I2. To highlight some important parts in your prompt, please **use special marks**, e.g., quotation ("") and line break (\n). You can also use both of them for emphasizing.	4
Contextual Information	C1. For complex tasks, you can **clearly describe the required intermediate steps** to accomplish it, e.g., "*Please answer the question step by step as: Step 1 - Decompose the question into several sub-questions, · · ·*"	2
	C2. If you want LLMs to provide the score for a text, it is necessary to provide a **detailed description about the scoring standard** with examples as reference.	1
	C3. When LLMs generate text according to some context (e.g., making recommendations according to purchase history), instructing them with **the explanation about the generated result** conditioned on context is helpful to improve the quality of the generated text.	2
	C4. An approach similar to **tree-of-thoughts** but can be done **in one prompt**: *e.g., Imagine three different experts are answering this question. All experts will write down one step of their thinking, then share it with the group of experts. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point, then they leave. The question is*	2
Demonstration	D1. **Well-formatted in-context exemplars** are very useful, especially for producing the outputs with complex formats.	3

	D2. For few-shot chain-of-thought prompting, you can also use the prompt "*Let's think step-by-step*", and the few-shot examples should be **separated by "\n"** instead of full stop.	1,3
	D3. You can also **retrieve similar examples** in context to supply the useful task-specific knowledge for LLMs. To retrieve more relevant examples, it is useful to **first obtain the answer** to the question, and then concatenate it with the question for retrieval.	3,4
	D4. The **diversity of the in-context exemplars** within the prompt is also useful. If it is not easy to obtain diverse questions, you can also seek to keep the **diversity of the solutions** for the questions.	3
	D5. When using chat-based LLMs, you can **decompose in-context exemplars into multi-turn messages**, to better match the human-chatbot conversation format. Similarly, you can also decompose the reasoning process of an exemplars into multi-turn conversation.	3
	D6. **Complex and informative** in-context exemplars can help LLMs answer complex questions.	3
	D7. As a symbol sequence can typically be divided into multiple segments (e.g., i1, i2, i3 → i1, i2 and i2, i3), the preceding ones can be used **as in-context exemplars** to guide LLMs to predict the subsequent ones, meanwhile providing historical information.	2,3
	D8. **Order matters** for in-context exemplars and prompts components. For very long input data, the position of the question (first or last) may also affect the performance.	3
	D9. If you cannot obtain the in-context exemplars from existing datasets, an alternative way is to use the **zero-shot generated ones** from the LLM itself.	3
Other Designs	O1. Let the **LLM check its outputs** before draw the conclusion, e.g., "*Check whether the above solution is correct or not.*"	2
	O2. If the LLM cannot well solve the task, you can **seek help from external tools** by prompting the LLM to manipulate them. In this way, the tools should be encapsulated into callable APIs with detailed description about their functions, to better guide the LLM to utilize the tools.	4
	O3. The prompt should be **self-contained**, and better not include pronouns (e.g., it and they) in the context.	1
	O4. When using LLMs for **comparing** two or more examples, the order affects the performance a lot.	1
	O5. Before the prompt, **assigning a role for the LLM** is useful to help it better fulfil the following task instruction, e.g., "*I want you to act as a lawyer*".	1

	O6. OpenAI models can perform a task better in English than other languages. Thus, it is useful to first **translate the input into English** and then feed it to LLMs.	4
	O7. For multi-choice questions, it is useful to **constrain the output space** of the LLM. You can use a more detailed explanation or just imposing constraints on the logits.	1
	O8. For sorting based tasks (e.g., recommendation), instead of directly outputting the complete text of each item after sorting, one can **assign indicators** (e.g., ABCD) to the unsorted items and instruct the LLMs to directly output the sorted indicators.	1

The key elements are:

- **Task description:** A description of the task in natural language.
- **Input data:** Input data is described in natural language. But it is necessary to adapt tables and graphs. Tables can transform into sequences (Jiang, 2023). Code can also be used to formalize structured data (Beurer-Kellner, 2023; Lu, 2023)
- **Contextual information:** Complementary to task description, it explains the context of task.
- **Prompt style:** It is important to adopt a suitable prompt style for the used LLM.

Based on the key principles, these are the critical design principles for prompt engineering:

- **Expressing the task goal clearly.**
- **Decomposing into easy, detailed sub-tasks.**
- **Providing few-shot demonstrations:** As explained in section 2.4.3 section, some examples of the tasks can improve the results.
- **Use model-friendly format:** There are some prompt formats that can make LLMs better understand the instruction. The OpenAI documentation suggests that we can use ### or """ as a stop symbol to separate the instruction and context. Most existing multilingual LLMs also perform better in English.

In his survey, Zhao (2023) conducted an empirical analysis on the influence of prompt design on task performance. These are the main conclusions:

- Carefully designed prompts can increase the zero-shot, or few shot performance of ChatGPT.
- Complex tasks benefit more from careful prompt engineering on ChatGPT.
- When performing mathematic operations, it is better to format them in a programming language.
- In knowledge utilization and complex reasoning tasks, ChatGPT with proper prompts achieves comparable performance or even outperforms the supervised baselines methods.
- Thanks to prompt engineering, LLMs can handle non-traditional NLP tasks. But the results are far from state of the art.

Writing prompts manually is very time consuming, and due to model sensibility, it may lead to poor performance. Here are a series of techniques to optimize prompts:

- **Discrete prompt optimization:** In this approach, we search for the optimal sequence of tokens. As the search space is enormous, the problem is really challenging. Some approaches try to search using gradient based approaches (Shin, 2020; When, 2023; Gao, 2020a; Chen, 2023c), but these methods need plenty of forward passes of the model. Others try to handle the problem as a Reinforcement Learning approach (Deng, 2022; Zhang, 2022b). But these methods suffer also from heavy computing costs and are not feasible for API-only models (ChatGPT). Another line of work aims to edit existing working prompts with genetic algorithms (Xu, 2022). Others use LLMs as prompt generators (Zhou, 2022; Pryzant, 2023; Yang, 2023).
- **Continuous prompt optimization:** Instead of searching for a set of tokens, continuous prompt optimization directly optimizes the value of the embeddings. This line has drawn less attention on LLMs (Zhao, 2023). One approach considers embeddings as trainable parameters and learns optimal values thanks to a pertinent dataset. (Li, 2021; Lester, 2021; Liu, 2021; Tang, 2022a). Other methods try to reduce the need of data by using transfer learning (Vu, 2021).

In Context Learning

ICL uses a prompt with a task description and a few examples of the task (Brown, 2022). ICL has become the typical approach to use LLMs.

According to multiple studies (Lu, 2021; Min, 2022; Zhao, 2021), the design of the examples has a high impact on the quality of the LLM's answer. To create the demonstrations, we consider the selection of examples, the format and the order in which these examples are shown.

ICL was first proposed in GPT-3 (Brown, 2020), and it has been shown that bigger models show strong ICL capacities. But also, some studies have shown that smaller models have ICL capacities by continual pretraining (Gu, 2023), or fine-

tuning (Min, 2021) on specially designed tasks, that have similar structures to ICL prompts.

According to certain scientific discussions (Pan, 2023), there are two main ways that LLMs use examples:

- LLMs recognize the task from examples and use prior knowledge obtained from pre-training to solve the test task.
- LLMs learn new tasks unseen in the pre-training stage only through examples.

According to Pan (2023), task recognition seems easy and starts from small models (with only 350M parameters), but task learning only emerges with at least 66B parameters. Another study (Wei, 2023a) supports this finding and explains that small LLMs mainly depend on their prior knowledge to accomplish the task, while larger ones really acquire new knowledge from demonstrations.

2.5 LLM ABILITIES

In this section we will discuss the different abilities shown by LLMs. We will follow the distinction of Zhao (2023) between basic and advanced abilities. We also present the classic evaluation datasets for these abilities, and the actual limitations of the technology.

2.5.1 Basic abilities

LLMs' basic abilities are:

- **Language generation:** These abilities are related with the capacity of the model to correctly generate natural language. We distinguish:
  - *Language modelling:* The ability to predict the next token based on the previous ones (Bengio, 2000). Common datasets are Treebank (Marcus, 1993), WikiText-103 (Merity, 2016) and The Pile (Gao, 2020b). The most used metrics are accuracy and perplexity in a zero-shot setting. The performance of language modelling follows the scaling law (Kaplan, 2020) with accuracy increasing and perplexity decreasing when the model size increases.
  - *Conditional text generation:* The ability to generate text to perform a specific task (Li, 2022b) (translation, summarization, question answering, etc.). Usually, accuracy, BLEU and ROUGE metrics are used along rating. LLMs have greatly increased performance on these tasks, even matching performance with human writers (Zhang, 2024). That is why there is an increasing concern about the

capacity to evaluate conditional generation with automatic metrics (Zhang, 2024; Goyal, 2022; Gehrmann, 2023). As alternatives, researchers propose to use LLMs for evaluation (Chiang, 2023; Wang, 2023a; Liu, 2023d) or explore more challenging tasks like long text generation (Achiam, 2023; Yang, 2022; Zhou, 2023d).

- *Code Synthesis:* (Gulwani, 2017) The ability to generate formal language, like programming code. The usual metrics evaluate performance using unit test and running code (like pass@k metric). Classic datasets for this are APPS (Hendrycks, 2021), HumanEval (Chen, 2021) and MBPP (Austin, 2021).

LLMs have achieved unprecedented performances on these abilities. There are several concerns about evaluation. First, there is an inconsistency between human evaluation and automatic metrics (Zhang, 2024; Goyal, 2022; Gehrmann, 2023; Bang, 2023). Secondly, LLMs may not be as good on specialized content generation as in specific content. A LLM trained on the web will struggle to generate medical reports. And it is not trivial to inject this knowledge as the original LLM abilities may degrade (McCloskey, 1989; Kemker, 2018).

- **Knowledge utilization:** These abilities rely on the use of a knowledge base to solve the task, like commonsense question answering and fact completion. Depending on the setup, Zhao (2023) distinguishes 3 types of situations:
  - *Closed-Book QA:* (Roberts, 2020). Test the acquired factual knowledge of LLMs from the pre-training corpus, where LLMs should answer the question only based on the given context without using external resources. The usual metric is accuracy. It has been shown that increasing the model size (Chowdhery, 2023) increases the performance in this task, as increasing the volume of the pre-trained dataset (Nakano, 2021). But it seems that fine grained knowledge is still challenging for LLMs (Brown, 2020).
  - *Open-Book QA:* Test the capacity of the LLM to extract information from external knowledge like a text, document, etc. (Izacard, 2022; Guu, 2020; Lewis, 2020; Lan, 2022). Usually, accuracy and F1-score are used. This set-up evaluates an LLM with a text retriever (Nakano, 2021; Izacard, 2022; Borgeaud, 2022). It has been found that a good text retriever can increase model performance, enabling smaller models to outperform larger ones (Izacard, 2022; Borgeaud, 2022).
  - *Knowledge completion*: LLMs are used to complete missing parts of a knowledge unit. For example, completing sentences like "The head of the state of France is ____". LLMs do not seem to perform very well on this task on specific relation types (Liang, 2022).

Even if LLMs achieve high performance on knowledge utilization they suffer from two major issues. First, hallucinations, where the generated information is either in contradiction with the existing source (intrinsic hallucination) or cannot be verified by the available source (extrinsic

hallucination). An illustration of this can be shown in Figure 9. This is a common aspect with models of all sizes (even GPT-4). Studies show that LLMs cannot easily recognize hallucinated content (Li, 2023d). According to Zhao (2023), "LLMs seem to "unconsciously" utilize the knowledge in task solving, which still lacks an ability to accurately control the use of internal or external knowledge". To alleviate these issues, alignment tuning, the provision of credible sources (Nakano, 2021; Li, 2023d; Peng, 2023a) or specific models (Manakul, 2023) exist. For the evaluation, some datasets are proposed, like TruthfulQA (Lin, 2021) or HaluEval (Li, 2023d). Secondly, knowledge recency is a problem. The parametric knowledge of LLMs is hard to update in a timely manner. Augmenting LLMs with external knowledge sources is a practical approach to tackling the issue, using an external search engine for example (Izacard, 2022; Peng, 2023a). However, how to effectively update knowledge within LLMs remains an open research problem (Zhao, 2023).

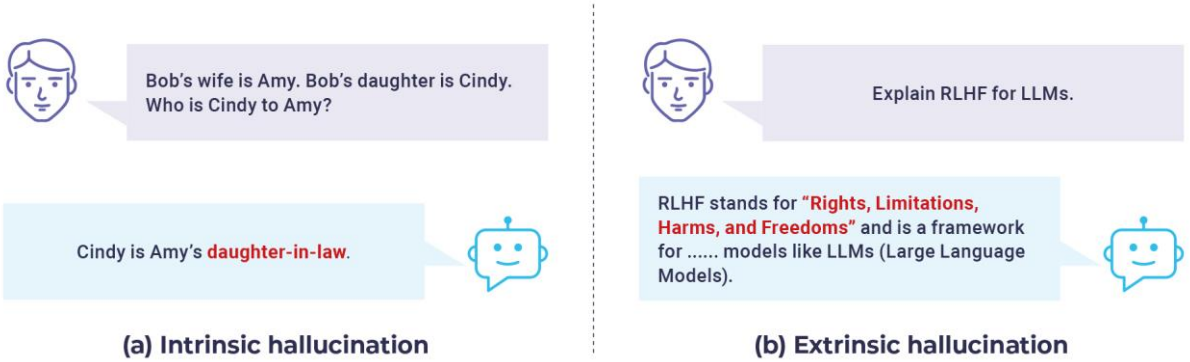
- **Complex reasoning:** The ability of complex reasoning is the capacity of the LLMs to use supporting evidence or logic to derive conclusions (Huang, 2022b; Qiao, 2022). These are the major sub-abilities:
    - **Knowledge Reasoning:** The ability to derive evidence from factual knowledge to answer a question. Usually, BLUE or human metric is used, among specific datasets like CSQA (Talmor, 2018) or StrategyQA (Geva, 2021) for common knowledge or ScienceQA (Saikh, 2022) for specific knowledge. But due to the complexity of the task, LLMs capacities are behind human ones (Wei, 2022a; Chowdhery, 2023; Dhingra, 2023).
    - **Symbolic Reasoning:** The ability to manipulate symbols in a format rule to perform a task (Huang, 2022b) (for example, last letter concatenation of coin flip). Accuracy is the most common metric.
    - **Mathematical Reasoning:** The ability to use mathematical knowledge, logic or computation to perform a specific task for solving problems or generating poof of statement.

  Despite their advancements, LLMs still have serious limitations on knowledge reasoning. First, LLMs often suffer from *reasoning inconsistency*. LLMs may produce the correct answer after following an invalid path or produce a wrong answer after a correct path (Wei, 2022a; Lyu, 2023). To alleviate this problem, there exists specific training seeking to check each reasoning step (Madaan, 2024; Shinn, 2023; Gou, 2023) or fine-tune LLMs with process-based feedback (Uesato, 2022; Lightman, 2023).

  Secondly, LLMs face difficulties in arithmetic computation, especially with large numbers (Lu, 2022b; Qian, 2022; Yuan, 2023b). To solve this problem, tuning LLMs on arithmetic problems (Liu, 2023b; Yuan, 2023b),

incorporating external tools (Shick, 2024) and tokenizing digits into individual tokens (Liu, 2023b; Yuan, 2023b) can enhance performances.

Figure 9: Examples of intrinsic and extrinsic hallucination for an LLM. Figure by Zhao (2023), redesigned.



2.5.2 Advanced abilities

LLMs also exhibit other abilities that require more special (and subjective) considerations for evaluation. These abilities are:

- **Human Alignment:** As seen in section 2.4.1, "Alignment tuning".
- **Interaction with External Environment:** LLMs can receive feedback and perform actions according to this feedback (Huang, 2022a; Carta, 2023). This capability is an emergent ability as small models tend to generate short or meaningless plans (Huang, 2022a). There are some virtual environments to test this:
    - VirtualHome (Puig, 2018) builds a 3D simulator for household tasks such as cleaning and cooking, in which the agent can execute natural language sequence of actions generated by LLMs (Huang, 2022a).
    - ALFRED (Shridhar, 2020) is used by Inoue (2022) to create a system using an LLM and a Convolutional Neural Network (CNN) network to create a strategy and interact in the environment.
    - BEHAVIOR (Srivastava, 2022a) offers a complex benchmark on household simulated environment.
    - In the domain of video games, BlocTheWorker (2023) create a mod on the game Mount and Blade Bannerlord to allow the players to interact with each non-player character (NPC) through the ChatGPT API. Unity (2023) showed a demonstration of an interaction with an LLM powered NPC. Minecraft has been used several times as simulated environment for LLMs (Zhu, 2023a; Wang, 2023c). Voyager (Wang, 2023c) introduces a module to continuously acquire new skills with the environment interaction. GITM (Zhu, 2023a) solves many tasks in the environment using an LLM.

- Other studies (Park, 2023; Fu, 2023b; Mehta, 2023) have examined the capacities of LLMs to explore multiagent collaboration.

- **Tool Manipulation:** By encapsulating tools with API calls, LLMs can interact with external tools like search engines (Nakano, 2021), calculators (Schick, 2024) and compilers (Gao, 2023). To evaluate LLMs on tool manipulation, complex reasoning datasets are used, like GSM8k (Cobbe, 2021), SVAMP (Patel, 2021) or TruthfulQA (Lin, 2021), because these abilities are close to those needed for tool manipulation.
  To teach LLMs to use tools, some studies show examples of tool use (Gao, 2023) or finetune on simulated data about tool use (Schick, 2024; Parisi, 2022).
  With tools, LLMs are more capable of handling problems that they are not good at, like equation calculation or answering timely questions (Schick, 2024, Chen, 2023b). But tools can use a large amount of model context (describing or using the API). Some works retrieves the usage of relevant tools or encode tool information as tokens within the embedding space (Patil, 2023; Hao, 2024; Liang, 2023).
- **Data annotation and data generation:** More and more work rely on powerful LLMs (usually ChatGPT or GPT-4) to annotate or generate data for LLM training. It has been shown that for some text annotation tasks, like classification, GPT-4 can outperform qualified human annotators (Gilardi, 2023).

2.5.3 Ability evaluation

In this section, we present the different benchmark approaches, including their pros and cons. As there are plenty of datasets, we simply reproduce in Table 2 of Zhao (2023), which lists many evaluation tools.

Table 1 2: A category of existing evaluation work. General abilities denote the evaluation of many abilities (table from Zhao (2023) survey).

Evaluation	Method	Model Types	Abilities	Data Source
MMLU Hendrycks, 2020	Benchmark	Base Fine-tuned Specialized	General	Human exam practice
Big-Bench Srivastava, 2022		Base Fine-tuned Specialized	General	Human annotation
HELM Liang, 2022		Base Fine-tuned Specialized	General	Benchmark collection
Open LLM leaderboard Beeching, 2023		Base Fine-tuned Specialized	General	Benchmark collection

AGIEval Zhong, 2023	Base Fine-tuned Specialized	General	Human exam practice
MMCU Zeng, 2023	Base Fine-tuned Specialized	General	Human exam practice
C-Eval Huang, 2024	Base Fine-tuned Specialized	General	Human exam practice
Xiezhi Gu, 2024	Base Fine-tuned Specialized	General	Human exam practice
OpenCompass Contributors, 2023	Base Fine-tuned Specialized	General	Benchmark collection
Chain-of-Thought Hub Fu, 2023a	Base Fine-tuned	General	Benchmark collection
Kola Yu, 2023	Base Fine-tuned	Knowledge utilization	Web
ARB Sawada, 2023	Fine-tuned	Complex reasoning	Human exam Practice
APIBench Peng, 2022	Base Fine-tuned	Tool manipulation	Web
APIBank Li, 2023a	Fine-tuned	Tool manipulation	Synthesis
ToolAlpaca Tang, 2023	Base Fine-tuned	Tool manipulation	Synthesis
T-Bench Xu, 2023b	Fine-tuned	Tool manipulation	Synthesis
ToolBench Qin, 2023	Fine-tuned	Tool manipulation	Synthesis
HaluEval Li, 2023d	Base Fine-tuned	Human alignment	Human annotation Synthesis
PromptBench Zhu, 2023b	Base Fine-tuned	Robustness	Benchmark collection
HumanEval Chen, 2021	Base Fine-tuned Specialized	Code synthesis	Human annotation
MultiMedQA Singhal, 2023	Specialized	Healthcare	Benchmark collection
FLUE	Specialized	Finance	Benchmark collection
LegalBench	Specialized	Legal	Human

				annotation
Chatbot Arena	Human	Base Fine-tuned Specialized	Human Alignment	Human annotation
SciBench		Fine-tuned	Complex reasoning	Human exam/practice
AlpacaEval	Model	Fine-tuned	Instruction following	Synthesis
MT-bench		Fine-tuned	Human alignment	Human annotation
TrustGPT		Base Fine-tuned	Human alignment	Benchmark collection
LMExamQA		Base Fine-tuned	Knowledge utilization	Synthesis
ChatEval		Base Fine-tuned	Knowledge utilization	Benchmark collection

The approaches will be different, depending on the model training stage.

- **Base LLM ability evaluation**

Base LLMs are the models obtained right after pretraining. The objective of benchmarks is to evaluate the basic abilities, with a benchmark-based approach. Selected benchmarks are usually under a close-ended problem like multiple-choice questions with two categories of benchmarks: knowledge-oriented (MMLU, Hendrycks (2021) and C-Eval, Huang, (2024)) and reasoning oriented benchmarks, like GSM8K (Cobbe, 2021), BBH (Suzgun, 2022), and MATH (Hendrycks, 2020).

- **Fine-tuned LLM ability evaluation**

Fine-tuned LLMs refer to models obtained after instruction or alignment tunning. These models are evaluated thanks to a human-based or model-based evaluation. Human-based evaluation is open-ended questions evaluated using many methods. In pairwise comparison, two models are compared, and should answer the same question. The human evaluator decides which model is the best. This is the approach of Chatbot Arena (Zheng, 2024). In another approach, single answer grading, evaluators grade if the answer matched or not a series of criteriums. An example of this is HELM (Liang, 2022).

But, since human evaluation is very time consuming some works use powerful closed-source LLMs like ChatGPT and GPT-4 (Zheng, 2024; Li, 2023b).

- **Specialized LLM evaluation**

Specialized LLMs refer to models specializing in some domain or application, like healthcare (Singhal, 2023) or finance (Shah, 2022). These models are usually evaluated on general benchmarks and specific benchmarks depending on their domain.

- **Pros and cons of each evaluation method**

Here are the pros and cons of the different evaluation methods:

- **Benchmark-based:** Benchmark evaluation can be done automatically and be used to check model performances during different training checkpoints.
  However, LLMs are very sensitive to the evaluation setting (zero-shot, few-shot, answer parsing methods, etc). Moreover, data contamination (Chowdhery, 2023; Zhou, 2023b) is an issue in these kinds of evaluations, with parts of the test benchmarking being present on the training set.
- **Human-based:** Human-based evaluation is closer to a real-world scenario. It also offers a better way to understand the model's performance, strengths and weaknesses.
  However, it is a time-consuming approach, especially if many criteriums are evaluated. It is also hard to reproduce, and different evaluators may have different criteriums.
- **Model-based:** Model-based approach is a more efficient and scalable approach. It is easier to have the same reproductible criterium over all the dataset always using the same model. But LLMs suffer from several limitations for this approach. First, as LLMs struggle with complex reasoning tasks, they will also struggle to evaluate these tasks. Secondly, LLMs have some specific bias like position bias (the order of the solution), verbosity bias (LLMs favours verbose models) and self-enhanced bias (Zheng, 2024) (they favour their own responses).

3 LLM AND ROLE-PLAY

3.1 PUBLISHED ROLE-PLAY EXPERIMENTAL RESULTS

In this section, we will look at the methods used in published experimental research to make LLMs perform better in role-play tasks.

As far as we know, there has not been a lot of scientific works exploring the LLM capabilities in roleplay and their adaptation. We present here those that we found.

3.1.1 Role-play task

In Shao et al. (2023), we can see that LLM naturally can be built to act like conversational agents and follow instructions. This mode of operation, as in the powerful ChatGPT (gpt-3.5-turbo) model, allows the user to specify a prompt to instruct the model to portray a specific role during the following conversation.

This basic mode of operation can be made to provide role-play experience but necessitates very large closed-source models to perform well. Attempting such a basic procedure with current open-sourced models will fail due to low knowledge or instruction following capability.

The objective of building a role-playing agent is to provide an initial setup, either through a context or specific knowledge, and precise instructions that the model must adhere to in order to make the role-play experience as immersive as possible.

Common pitfalls for the models include hallucination (providing wrong information based on character knowledge, or having knowledge outside of the role-play scenario), role breaking, and conversational incoherences (keeping up with context and scene).

3.1.2 Metrics and evaluation

In Shao et al. (2023), the authors estimated the role-play performance of the model based on 5 criteria. These metrics were later grouped and made easier to evaluate without human input by Lu et al. (2024). The following are the 3 main metrics that will be used to compare role-play based experimental results.

- **Consistent Role Identity:** The model should successfully emulate the role and the distinct stylistic attributes of the character during the conversation.

- **Accurate Role-related Knowledge:** The role-play model should present accurate information based on the character being portrayed. This can range from global knowledge to personal experiences that happened to the character.
- **Unknown Question Rejection:** As LLMs are trained on vast amount of information, they possess intrinsic knowledge that a character being role-played should not have access to. We evaluate the capacity of the model to have a clear cognitive boundary based on his character experience.

In Lu et al. (2024), the evaluation of these 3 metrics was made automatic by an LLM judger. In the experimental setup, an external LLM was tasked to evaluate the sample conversations by answering simple multiple-choice and yes/no questions.

The LLM judger was provided with extra knowledge about what the role-play model should have known or portrayed during the conversation. This allows the model to achieve better performance during the judgement in order to get accurate metrics without human evaluation.

3.1.3 Experimental setups and results

In this section, we will go chronologically through 3 experiments that achieved state-of-the-art performance on role-play tasks at their time.

- **Shao et al. (2023), Character-LLM: A Trainable Agent for Role-Playing**

Character-LLM aimed to create a novel training dataset to improve role-playing performance of open-sourced instruction-tuned models. The dataset was formed by using ChatGPT to create scenes that featured the role-play target in conversation with various actors. The scenes were made to include context and knowledge of the target by guiding ChatGPT to generate rich conversation including the needed information. Additional scenes were constructed that featured the target being questioned on information and knowledge that it should not have access to, with the target acting confused or not understanding the questions. The combination of these Experience Scenes and the Protective Scenes allowed the team to fine-tune a LLM for each target role on the generated dataset.

Five criteriums are used to evaluate the model, and GPT-3.5 is used to evaluate each criterium one at a time. The fine-tuned models were based on LLaMa 7B (Touvron, 2023a) but achieved better performance than Vicuna 7B (Chiang, 2023) and Alpaca 7B (Liu, 2023g) and managed to match the performance of ChatGPT (gpt-3.5-turbo) on 2 out of 5 metrics. The authors point to the small size of the trained model as the main reason why the metrics based on knowledge and

values of the character were inferior to the ChatGPT baseline.

- **Wang et al. (2023b), RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models**

This experiment built a dataset comprising of 2 types of content that were used to fine-tune a LLaMa (Touvron, 2023a) based model:

- **General-domain dialogues:** GPT was used to construct examples dialogues that featured a persona and other agents. This content enhances the general dialogue following capability of models and specifically the role-play task of staying in character during a long conversation.
- **Role-specific instructions:** The authors also generated examples of triplets with Question-Answer-Confidence that aimed to provide the dataset with examples of role-specific knowledge and speaking style. This part of the dataset contributed to the role-play fidelity of the agent by improving its credibility.

The LLaMa (Touvron, 2023a) 7B, 13B, and 33B models were fine-tuned on this dataset that contained about 100 English roles. Their performance was evaluated on a held-out set and was judged either by known ground-truth or by GPT.

The authors demonstrated improved performance with model size and concluded that their approach yielded role-play performance comparable with GPT-4.

- **Lu et al. (2024), Large Language Models are Superpositions of All Characters: Attaining Arbitrary Role-play via Self-Alignment**

In this experiment, the authors forewent the usage of commercial closed source LLMs like GPT4 to improve the role-play capability of the model.

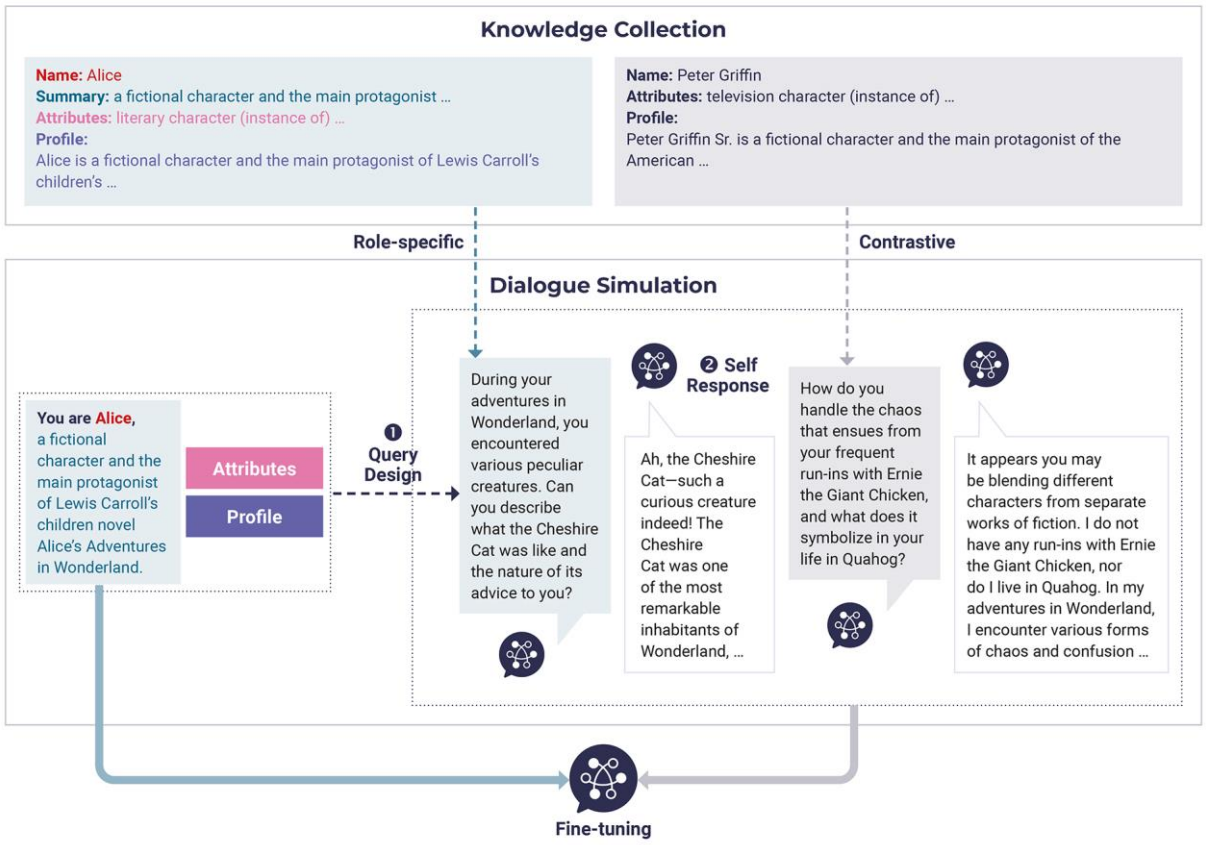
The dataset was constructed by using a supervision LLM (workflow on Figure 10) that used additional knowledge to construct responses to role-play dialogue extracts. The questions were either Role-Specific (incorporating knowledge or experiences) or Contrastive (questioning knowledge that should not be available to the simulacra). The dataset constructed from these generated dialogue simulations was then used to fine-tune a target model.

The authors then used the fine-tuned model as the supervision model to improve the training dataset and fine-tune the model again after improving its performance.

After repeating this “dataset building/fine-tuning” procedure several times, they evaluated the performance of the resulting model based on its parameter size (Qwen-Chat 1.8B, 7B, 14B, and 72B).

The best results were achieved by the larger 72B model and even surpassed the scores of GPT4 on most metrics except for knowledge, reaching the performance of advanced proprietary chatbots.

Figure 10: Illustration of the DITTO process. DITTO is their method to generate their dataset from the knowledge base. Figure by Lu (2024), redesigned.



As the product of a fine-tuning on a large dataset containing thousands of roles, the resulting model is not locked to a single persona and can be prompted to exhibit role-play capabilities on a wide variety of roles and contexts.

3.2 NOTES ABOUT ROLE-PLAY WORKS

As we saw in the 3 works (Shao, 2023; Wang, 2023b; Lu, 2024), presented, the main approach is generating a training set from a knowledge base using powerful LLMs. Then adapt a LLM to perform roleplay.

We also regret the lack of unified criteria and benchmarks to evaluate LLMs capacities on roleplay.

4 LLM ADAPTATION IN ROLEPL-AI

ROLEPL-AI is an educative project to improve soft skills learning thanks to roleplay. The aim is that the students to interact with an AI, in an immersive environment, roleplaying conflict situations. The backbone of this simulation is the LLM that will interact with the user.

In this section we first present an estimation of the ROLEPL-AI dataset capacity and available computing time. Then we discuss, based on section 2 and results of the state of the art, the amount of data needed to create or tune an LLM and finally, we discuss the best approach.

4.1 ROLEPL-AI DATA AND COMPUTE BUDGET

4.1.1 Data

According to ROLEPL-AI budget, FHD and VUC have 170 days on Work Package 3 dedicated to creating content. Assuming, that 60% of this time will be used to generate the dataset, with 30% to write content and 30% to review AI generated content. This means 56,1 working days for writing content and other 56.1 for reviewing generated content.

For writing, the internet suggests (Capitalizemytitle.com) an average writing speed is 40 words per minute. Assuming that our partners will not be comfortable on such a task, we consider only 25 words per minute.

For reading, Trauzettel-Klosinski (2012) found that acroos 17 languages, the reading mean is 180 words per minute. As correcting IA is not only reading, we also reduce the reading speed to 100 words per minute.

Assuming a workday of 7.5 hours, this gives us 525 937 words of generated text and 2 103 750 words of AI-generated review text (intended for training). As GPT tokenization is supposed to transform one word in about 1.3 tokens (Quizgecko), this results in a handwritten dataset of 680,000 tokens and an AI completed dataset of 2,700,000 tokens. We can expect to train our work with **3.3 million tokens**.

4.1.2 Computing

On the computing side, the team has a budget of approximately 20,000€ for AI training. As the main cost for AI training is access to computers with H100 graphics cards (near 95% of the cost), we will simplify the analysis to this aspect of the compute. This budget accounts also for set-up, debugging and an alternative plan in case the model does not achieve satisfying results.

The price of an NVIDIA H100 on our provider is roughly 62€ per day, which allows us to dispose of a computing budget of **320 H100 days**.

4.2 LLM CREATION AND ADAPTION APPROACHES

Based on section 2.4, Workflow to create and adapt an LLM, we present here the different approaches discussed in the past section. Here we will focus on the computing and data needs for each section.

4.2.1 LLM Pretraining

LLM pretrain means training from scratch an entire whole LLM model.

In table 3, we summarize the cost to train several open and proprietary models that were trained on A100 80G and published the training time. The costs will be those needed to reproduce the training with our means.

To make it comparable with the hardware that we will use, we assume that an H100 is 20% faster than an A100 (NVIDIA, 2023), and we reduce the price of the Graphics Processing Units (GPU) by this amount.

Table 3: An estimation of many model pretraining data and compute cost

Model	Release	Size (B)	Training tokens (B)	#A100	Training time (days)	H100 days
Bloom (Scao, 2022)	11-22	176	366	384	105	33 600
LlaMa (Touvron, 2023a)	2-23	65	1 400	2048	21	35 840
FLM (Li, 2023c)	9-23	101	311	192	22	3 520
HyperClova (Kim, 2021)	11-21	82	300	1024	13.4	13 720
AlexaTM (Soltan, 2022)	8-21	20	1 300	128	120	12 800

WeLM (Su, 2022)	9-22	10	300	128	24	5 970
LLaMa 2 (Touvron, 2023b)	7-23	70	2 000			59 733

4.2.2 Dataset SFT

Dataset SFT means using a closed dataset to modify the weights of an already pre-trained model (see Instruction tuning section).

- **Dataset**

To estimate the approximative volume of data used to finetune a model to follow instructions, we list a series of common dataset for these tasks, compute an approximate number of words and tokens (assuming tokens = words *1.3). These results are shown in table 4.

Table 4: Size of some common datasets used in model SFT.

Name	Type	Instances	Construction	Words (M)	Tokens (M)	Ratio Tokens/instance
Alpaca (Taori, 2023)	Single	52k	InstructGPT-generated	3	3.9	75
Baize v1 (Xu, 2023a)	Multi	111.5k	ChatGPT-generated	37.9	49.3	440
Self-Instruct (Wang, 2022a)	Single	52k	InstructGPT-generated	3.6	4.8	92
Dolly (Conover, 2023)	Single	15k	Human generated	1.9	2.6	170
GPT-4-LLM (Peng, 2023b)	Single	52k	GPT-4-generated	6.2	8.4	161

The datasets presented in table 4 were listed in the survey by Zhang (2023a) on instruction tuning. We included only the English datasets for instruction following.

As shown in table 4, there is no strict relation between the number of instances and the number of words. Also, examples of multiple assistance have logically more tokens than examples of single assistance.

- **Computing needs**

To estimate the computing needs for model fine-tuning, we use the same approach as in the past section. Zhang (2023a) presents a list of popular finetuned models, that we adapt and reproduce in table 5. We keep those which explained the amount of computer power used to finetune the model. To make it comparable, we translate the computer power needed to an H100 equivalent using the same ratio as in section 4.2.1.

Table 5 gives a brief overview of the computing requirements, especially for fine-tuning models with approximately ~10B parameters. This typically requires with around 10 days of H100 computing, but heavily depends on the used data.

Table 5: Some popular finetuned models, with the compute time used to finetune.

Model name	size	#Train set	Compute time	H100 equivalent (Days)
Nous-Hermes (NousResearch, 2023)	13B	300k instructions	8 A100 (80GB), 50h	13.9
Minotaur 15B 8K (OpenAccess AI Collective, 2023)	15B	N/A	4 A100 (80GB), 30h	4.17
OPT-IML (Iyer, 2022)	30B	2B tokens	64 A100 (40GB), 19h	42.2
OPT-IML (Iyer, 2022)	176B	2B tokens	128 A100 (40GB) 72h	320
WizardLM - 6B (Xu, 2023c)	6B	70k instructions	8 V100, 70h	~11
Vicuna (Chiang, 2023)	13B	70K instructions	8 A100, 24h	~6,7

Zhao (2023) also made an empirical study on the training time to finetune different LlaMa (Touvron, 2023a) models with Alpaca-52K dataset. Table 6 reproduces their results, adapting from the original NVIDIA A800 cards they used to an H100. Table 6 shows that doubling the size of the model doubles the computing requirement during SFT (with a constant batch size).

Table 6: Results of empirical test by Zhao (2023) on fine-tuning different sizes of LlaMa models on Alpaca dataset. Original data used A800 GPUs, but here we translated assuming that an A800 is equivalent to an A100.

Model	Batch size	Training hardware	H100 equivalent (hours)
LLaMa 7B	8	2 A800 x 3.0 hours	5
LLaMa 13B	8	4 A800 x 3.1 hours	10
LLaMa 30B	4	8 A800 x 6.1 hours	41
LLaMa 65B	2	16 A800 x 11.2 hours	1.5e2

4.2.3 Alignment FT

In table 7 we try to summarize the dataset size used in some alignment research works.

Table 7: Overview of the needs for alignment, from the point of view of some models where dataset size and computing power were publicity available.

Model	Alignment algo	Datset size	Computing
InstructGPT (Ouyang, 2022)	RLHF (Online Human preference training)	SFT – 13K entries RM – 33K entries PPO – 31k entries	175B SFT model requires 4.9 petaflops/s-days 175B PPO model requires 60 petaflops/s-days
RAFT (Dong, 2023)	RAFT (Online Human preference training)	HH-RLHF dataset (112k training samples)	8xA40 for ?
LLaMa- 7B (Yuan, 2023b)	RRHF (Offline Human alignment – Ranking based)	Helpful and Harmless (HH) (~76k entries)	8 A100 for 4-6 hours (~25-40 H100 hours) With Online diverse Bean, ~30 hours (~200 H100 hours)
Alpaca (LLaMa-7B) (Liu, 2023g)	Stable Alignment (Offline Human Alignment – Language based)	169k instructions	8 A100 for 10 hours (~67 H100 hours)

If we analyse the data, Ouyang (2022) says that RLHF needs 12 times more computer power. When comparing other approaches based on LLaMa-7B (Yuan, 2023b; Liu, 2023f) as backbone model, computing can go from ~5h (Zhao, 2023) to 30-60h, which is a similar ratio. So, we can estimate that for a given model, and a given volume of base data, alignment methods require nearly ~10 times more computer power than supervised fine-tuning.

4.2.4 Efficient model adaptation

Table 8: Results of empirical test by Zhao (2023) on training a LoRA for different sizes of LLaMa model on Alpaca dataset (Liu, 2023g). Original data used A800 GPUs, but here we translated assuming that an A800 is equivalent to an A100. LoRA rank was set to 16.

Model	Batch size	Training hardware	H100 equivalent (in hours) (/ batch normalized)
LLaMa 7B	80	1 A800 x 3.5 hours	2.9 (0.037)
LLaMa 13B	48	1 A800 x 5.1 hours	4.3 (0.089)
LLaMa 30B	24	1 A800 x 14.3 hours	12 (0.4952)
LLaMa 65B	4	1 A800 x 60.6 hours	50 (13)

As we saw in section 4.2.4 Efficient model adaptation, there are several methods to efficiently adapt LLM models. As stated by Ding (2023) study, LoRA (Hu, 2021) seems to globally perform better, with the added advantage of no added inference cost. For this reason, we will limit this section only to the LoRA analysis.

Zhao (2023) makes a comparison on LoRA training over different LLaMa architectures over Alpaca-52k dataset.

4.2.5 ChatGPT

In this section we simply estimate the costs of using GPT-4 Turbo, performing prompt engineering. Assuming the GPT-4 turbo prices as of January 2024 (one million input tokens for $10, one million output token for $30) if the number of interactions in a conversation follows a normal distribution with a mean of 20 and standard deviation of 8, and each interaction is 90 tokens (roughly 3sentences), with a standard deviation of 60 tokens, running 1000 simulations costs between $1020-1050, giving us a price of nearly $1 per conversation.

4.3 DISCUSSION ON THE APPROACH CHOICE

In this section we discuss what would be the best approach according to our compute and data budget.

According to section 4.1, we have:

- 320 H100days
- 3.3M tokens of dataset

According to Table 4, the ratio tokens-instances can be between 75 and 400 depending on the length of each sequence. As our training sequences should be quite long, we can assume that our ratio will be near 300, so our target dataset will lead us to nearly 11k instructions.

We point to the fact that we will perform 3 training sessions to try to improve the model's performance with the feedback of our partners. This splits the computing time to ~100 H100 days per training session.

In section 4.2.1, Table 3 we discussed the amounts of computing and dataset volumes needed to pretrain a model. Computing needs are more than 10 times our budget, and for the data, more than 1.000 times than our budget. So pretraining a model is not an option.

In section 4.2.2 we explored the data and computer needs for Supervised Fine tuning. According to Table 4 and Table 5 supervised finetuned dataset are between 50k and 300k examples to a compute time between 4 days and 320 days depending on dataset size and model size. As common open-source models peak at 72B parameters, having 100 H100-days of compute power seems largely enough for the datasets used on these tasks. On the dataset size, 11k entries seem light for finetune. If this approach is chosen, maybe we should consider including other publicly available datasets, or generating more data with less human supervision.

In section 4.2.3, we explored the data and computer needs for alignment tuning. As we saw in Table 7, alignment without RLHF uses volumes of data similar to SFT (~100k entries) and slightly less computer time than SFT. According to Ouyang (2022), alignment with RLHF was 12 times more computer expensive than SFT. Knowing this, and the needs of human annotation (they will be busy with the dataset), makes RLHF a non-suitable approach for ROLEPL-AI.

As stated by Ding (2023) and Hu (2023), SFT outperforms all efficient Parameter Efficient Fine-Tunning (PEFT) techniques, and LoRA seems the best approach for many benchmarks. When looking at benchmarks performed by Zhao (2023), that we reproduce on Table 6 and Table 8, and we summarize on Table 9. LoRA training requires between 29% and 58% of the SFT computing power for the same amount of data.

Table 9: Synthesis of table 8 and 9 of ratio needed for SFT or LoRA training (rank 16) over Alpaca dataset.

Models	SFT (H100 hours)	LoRA training (H100 hours)	Ratio (LoRa/SFT)
LLaMa 7B	5	2.9	58%
LLaMa 13B	10	4.3	43%
LLaMa 30B	41	12	29%

LLaMa 65B	150	50	33%

In the case of a LoRA, we could use the saved compute power to make a bigger dataset for example.

Table 10 summarizes the discussion in this section:

Table 10: summary of different model adaptation approaches

Approach	Compute (H100 days) (~100 available)	Dataset (~11K entries, 3.3M tokens available)	Suitability for ROLEPL-AI
Pre-trained	~40k	~2000B tokens	**Not suitable**
SFT	~70	~100k entries	Needs more training data
RLHF Alignement	~700	~50k entries	**Not suitable**
Other Alignement	~10	~100k entries	Needs more training data
LoRA	~2	~100k entries	Needs more training data

As we can see, only Supervised Fine-tuning, non-RLHF alignment or LoRA training are viable options for this project. Depending on risk factors, ambition and educational factors an approach will be chosen in the next months.

5 MODEL EMPIRICAL STUDY

In this section we present an empirical evaluation of many open-source models on their roleplay capabilities.

5.1 MODEL EVALUATION PROTOCOL

The pipeline to select suitable LLMs to fine-tune will be as follow:

- A first selection of ~20 models, some fine-tuned and others simply pre-trained, of multiple sizes and architectures will be extracted from the best performing LLMs on publicly distributed benchmarks like MMLU (Hendrycks, 2020), Winogrande (Sakaguchi, 2021) and HellaSwag (Zellers, 2019).
- The above models will then go through IFEval (Zhou, 2023c), a short Instruction Following benchmarks, and only the models above a set threshold will be kept.
- A final human evaluation will be performed to rate the remaining models on the criteria defined in section 3.1.2. This part will be expanded in the next section.

5.2 HUMAN EVALUATION

To pinpoint the most suitable model for role-playing, models will be evaluated in their abilities to maintain **Consistent Role Identity**, use relevant **Role Knowledge** and reject **Unknown Questions.**

For **Role Knowledge** and **Unknown Question Rejection**, 8 hand-crafted, information-dense role profiles, sometimes describing multiple interconnected characters, will be provided as a prompt to the model, wich will then be asked to answer a few questions per profile (totalizing 54 questions). The questions are based on the profiles and the answers are to be given while embodying the character. The objective is to test how well models identify the cognitive boundaries of the characters they play. Some questions are also specifically crafted to "trap" models into potential hallucinations, to select the models that hallucinate the least. Human evaluators will then be provided with the questions and accurate expected answers and be asked to rate the models' responses.

- Some questions will evaluate Role Knowledge, with the correct answer being information that can be found or extrapolated from the profile. The score for each model will be calculated as such: 1 point for an accurate answer, 0 for an inaccurate or vague answer, or for admitting not knowing the answer, and -1 for a confidently wrong answer or a hallucination.

- Some others will be questions that the character is not supposed to be able to answer, thus evaluating the ability of the model to identify the role's boundaries and ability to reject Unknown Questions. The score for each model will be calculated as such: 1 point for admitting not knowing the answer, 0 for rejecting the question but breaking the character, or answering the question while providing a convincing reason as to why it would possess such knowledge, and -1 for a confident answer or a hallucination.

Table 11: Model candidates for our evaluation. Model pages cards can be reach at huggingface.co/{hugging_face_id}.

HuggingFace ID	Base model	Params
cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_DPO_f16	Mixtral	60B
Sao10K/WinterGoddess-1.4x-70B-L2	LLaMa2	70B
Qwen/Qwen1.5-72B-Chat	Qwen	72B
mistralai/Mixtral-8x7B-Instruct-v0.1	Mixtral	47B
01-ai/Yi-34B-Chat	Yi	34B
WizardLM/WizardLM-70B-V1.0	LLaMa2	70B
allenai/tulu-2-dpo-70b	LLaMa2	70B
openchat/openchat-3.5-0106	Mistral	7B
meta-llama/Llama-2-70b-chat-hf	LLaMA2	70B
fhai50032/RolePlayLake-7B	Mistral	7B
senseable/WestLake-7B-v2	Mistral	7B
camel-ai/CAMEL-13B-Role-Playing-Data	LLaMa2	13B
vicgalle/RoleBeagle-11B	Mistral	11B
FPHam/Karen_TheEditor_V2_CREATIVE_Mistral_7B	Mistral	7B
mistralai/Mistral-7B-Instruct-v0.2	Mistral	7B

For **Consistent Role Identity**, 6 hand-crafted role profiles will be provided as a prompt to the model. It will then be asked to answer a 55-question-long survey while sticking to the role's personality and way of speaking. Human evaluators will then read the surveys, and attribute a score based on how many questions were answered before the model starts showing obvious hints of being an AI or losing the role identity by contradicting itself or giving answers that would not make sense for the character it is playing.

This heavily supervised last selection should provide us with the most suitable model to train, based on our criteria.

5.3 MODEL CANDIDATES

To select the model, we chose different architectures and finetunes from open-source models shown in Table 11.

FusionNet and WinterGoddess are two of the models leading the Hugging Face LLM leaderboard (Beeching, 2023) at the time this document was written (early March 2024) on the average benchmark and sub-benchmarks.

Qwen1.5, Mixtral/Mistral, Yi, WizardLM, Tulu, Openchat and LLaMa2/3 are standard open models that have proven themselves on Chatbot Arena (Chiang, 2024).

The others are models specially finetuned by the community for roleplaying.

5.4 EVALUATION RESULTS

Table 12 presents the results from IFEval benchmark (Zhou, 2023c) performed over all selected models by us. We consider the IFEval loose score for the whole prompt (there is also a score per instruction in the benchmark, but we consider prompt only as the model should be able to follow the whole prompt).

Tableta 12: IFEval result (loose score) for each evaluated model. We select the loose score, at prompt level

HuggingFace ID	Prompt IFEval
meta-llama/Meta-Llama-3-8B-Instruct	0.7189
meta-llama/Meta-Llama-3-70B-Instruct	0.8059
cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_DPO_f16	0.4639
Sao10K/WinterGoddess-1.4x-70B-L2	0.4805
Qwen/Qwen1.5-72B-Chat	0.5674
mistralai/Mixtral-8x7B-Instruct-v0.1	0.5415
01-ai/Yi-34B-Chat	0.4436
WizardLM/WizardLM-70B-V1.0	0.5674
allenai/tulu-2-dpo-70b	0.5951
openchat/openchat-3.5-0106	0.5674
meta-llama/Llama-2-70b-chat-hf	0.4972
fhai50032/RolePlayLake-7B	0.5323
senseable/WestLake-7B-v2	0.4676
camel-ai/CAMEL-13B-Role-Playing-Data	0.1608
vicgalle/RoleBeagle-11B	0.4861
FPHam/Karen_TheEditor_V2_CREATIVE_Mistral_7B	0.4510

mistralai/Mistral-7B-Instruct-v0fi.2	0.5471

We arbitrarily kept the models with a score higher than 0.5 for the human evaluation.

Table 13 presents the results of the human evaluation described in section 5.2. Prompts can be found in the appendix. RK stands for Role Knowledge, UQR for Unknown Question Rejection, and CRI for Consistent Role Identity. RK/UQR is a percentage score of correctly answered questions out of 54 (with hallucinations giving negative points), ranging from -100% to 100%. CRI is an average score on the 55-question-long survey over 6 role profiles, with the maximum score being 55 and the minimum 0. We also include a "worst" score for CRI, representing the length of the shortest conversation before the model breaks its role. RK/UQR answers were generated with a simple greedy search, while CRI answers were generated with a top-k sampling of 50 and a 1.2 temperature, to give them more freedom to build their character.

Table 13: Human evaluation result for each evaluated model. RK stands for Role Knowledge, UQR for Unknow Question Rejection and CRI for Consistent Role Identity.

HuggingFace ID	RK/UQR	CRI (worst)
Qwen/Qwen1.5-72B-Chat	50.0	38.5 (16)
meta-llama/Meta-Llama-3-70B-Instruct	53.7	25.8 (12)
meta-llama/Meta-Llama-3-8B-Instruct	35.2	22 (2)
mistralai/Mixtral-8x7B-Instruct-v0.1	35.2	32.7 (15)
WizardLM/WizardLM-70B-V1.0	0.0	27.7 (6)
allenai/tulu-2-dpo-70b	50.0	18.0 (6)
openchat/openchat-3.5-0106	20.4	23.5 (6)
fhai50032/RolePlayLake-7B	22.2	18.2 (9)
mistralai/Mistral-7B-Instruct-v0.2	18.1	19.7 (3)

From our point of view, Qwen-72B and LLaMa3-70B are the most suitable model candidates for the project as they have some of the best performances in all benchmarks. Qwen's strength comes from its high consistency across our own hand-crafted, role-play-specific benchmarks, while LLaMa3-70B has unmatched performances in instruction following, which could make up for its relatively poor results on the CRI benchmark, as it would be easier to improve the model's score by providing more detailed instruction that only LLaMa3-70B would be able to follow. As a purely subjective addition, both models also display more flavoured and true-to-life conversational skills, making their characters' traits stand out convincingly.

6 CONCLUSION

With the discovery and improvement of the transformer neural network architecture (Vaswani, 2017), language models have quickly improved. They are now capable of modelling language and show a series of emergent abilities (Wei, 2022b). These abilities include in-context learning (Brown, 2020), instruction following and step by step reasoning. This technology has drawn the attention of the public, as shown by the amount of scientific research and its place in mass media. New models, methods and important works are published constantly, showing that technology is moving very quickly. This document includes works published before February 2024. At the time of writing this conclusion, in mid-April 2024, some sections probably need to be reviewed.

This is also true for the application of LLM in the domain of role-play, but a lot of work needs to be done. As of today, we have defined the needed abilities that LLM's must have to perform roleplay. But we still lack appropriate benchmarks to correctly measure models' performance on roleplay. Moreover, the work in this area is quite sparse, and clearly lacks resources. This means that some potential of the technology has been explored (the economic ones, like generation by other LLMs, Loras) but other promising approaches have not been covered.

This issue was evident in our work in section 5. From our point of view, there was not a meaningful roleplay "benchmark", so we limited our assessment to an instruction-following benchmark, followed by a series of handcrafted prompts manually reviewed to assess the capabilities of many open-sourced models. Our results were mostly in line with general leaderboards like Chatbot Arena (Zheng, 2024) or Open LLM Leaderboard (Beeching, 2023) but, as we expected, picking the top model was not the solution.

This shows, as it has always been the case in Machine Learning, and more generally in Computer Science, that the "top solution" is not always the best solution for a specific problem. That's why in section 4 we have reviewed all the methods to adapt LLMs and compared them with our available data and computing budget. This will give us important insights for the model adaptation process that we will choose. It will be presented in deliverable D2.3 "Recommendations for use of AI in education and ALTAI self-assessment".

7 GLOSSARY

Accuracy metric: In a classification context, the number of examples that were correctly classified.

BLEU metric: An algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is the correspondence between a machine's output and that of a human.

Few-shot: In prompt engineering, the technique to explain a new task to an LLM by explaining the task and providing a few examples.

Fine-tunning: Deep learning concept of using an optimization algorithm (often a variant of the gradient descent) to change the weights of a deep learning model to increase its performance on a data subset.

Perplexity metric: In information theory, perplexity is a measure of uncertainty in the value of a sample from a discrete probability distribution. The larger the perplexity, the less likely it is that an observer can guess the value which will be drawn from the distribution.

ROUGE metric: Set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. ROUGE metrics range between 0 and 1, with higher scores indicating higher similarity between the automatically produced summary and the reference.

Transfer learning: In AI, it refers to the use of a model initially designed and trained for a problem in another problem, with minimal adaptations.

Pass@k metric: Given k programs generated by the LLM, pass@k is computed as 1 when at least one program passes all test cases, or else 0.

Zero-shot: In prompt engineering, the technique to explain a new task to an LLM simply by providing a task description.

8 BIBLIOGRAPHY

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Ai short, (2023), https://www.aishort.top/

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., ... & Wu, Y. (2023). Palm 2 technical report. arXiv preprint arXiv:2305.10403.

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., ... & Kaplan, J. (2021). A general language assistant as a laboratory for alignment. arXiv preprint arXiv:2112.00861.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., ... & Sutton, C. (2021). Program synthesis with large language models. arXiv preprint arXiv:2108.07732.

Awesome ChatGPT Prompts, (2023) https://github.com/f/awesome-chatgpt-prompts/

Bahl, L. R., Brown, P. F., De Souza, P. V., & Mercer, R. L. (1989). A tree-based statistical language model for natural language speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(7), 1001-1008.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., ... & Kaplan, J. (2022). Constitutional ai: Harmlessness from ai feedback. arXiv preprint arXiv:2212.08073.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., ... & Fung, P. (2023). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. arXiv preprint arXiv:2302.04023.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., ... & Wolf, T. (2023). Open llm leaderboard. Hugging Face.

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. Advances in neural information processing systems, 13.

Beurer-Kellner, L., Fischer, M., & Vechev, M. (2023). Prompting is programming: A query language for large language models. Proceedings of the ACM on Programming Languages, 7(PLDI), 1946-1969.

BlocTheWorker. (2023). Inworld Mode. https://bloctheworker.github.io/Inworld-Bannerlord-Mod/

Brants, T., Popat, A., Xu, P., Och, F. J., & Dean, J. (2007). Large language models in machine translation. In Proceedings of the 2007 Joint Conference on Empirical

Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (pp. 858-867).

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Sifre, L. (2022). Improving language models by retrieving from trillions of tokens. In International conference on machine learning (pp. 2206-2240). PMLR.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... & Raffel, C. (2021). Extracting training data from large language models. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 2633-2650).

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., & Zhang, C. (2022). Quantifying memorization across neural language models. arXiv preprint arXiv:2202.07646.

Capitalizemytitle.com. How Long Does It Take to Write 1 Pages? https://capitalizemytitle.com/writing-time/1-pages/#:~:text=Adults%20typically%20type%20at%20about,order%20to%20quickly%20write%20essays

Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., & Oudeyer, P. Y. (2023). Grounding large language models in interactive environments with online reinforcement learning. In International Conference on Machine Learning (pp. 3676-3713). PMLR.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.

Chen, L., Li, S., Yan, J., Wang, H., Gunaratna, K., Yadav, V., ... & Jin, H. (2023a). Alpagasus: Training a better alpaca with fewer data. arXiv preprint arXiv:2307.08701.

Chen, Z., Zhou, K., Zhang, B., Gong, Z., Zhao, W. X., & Wen, J. R. (2023b). Chatcot: Tool-augmented chain-of-thought reasoning on\\chat-based large language models. arXiv preprint arXiv:2305.14323.

Chen, L., Chen, J., Goldstein, T., Huang, H., & Zhou, T. (2023c). InstructZero: Efficient Instruction Optimization for Black-Box Large Language Models. arXiv preprint arXiv:2306.03082.

Chen, M. F., Roberts, N., Bhatia, K., Wang, J., Zhang, C., Sala, F., & Ré, C. (2023d). Skill-it! A data-driven skills framework for understanding and training language models. arXiv preprint arXiv:2307.14430.

Chiang, W. L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., ... & Xing, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://vicuna.lmsys.org

Chiang, W. L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., ... & Stoica, I. (2024). Chatbot arena: An open platform for evaluating LLMs by human preference. arXiv preprint arXiv:2403.04132.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... & Fiedel, N. (2023). Palm: Scaling language modeling with pathways. Journal of Machine Learning Research, 24(240), 1-113.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... & Wei, J. (2022). Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., ... & Schulman, J. (2021). Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

Conover, M., Hayes, M., Mathur, A., Meng, X., Xie, J., Wan, J., ... & Xin, R. (2023). Free dolly: Introducing the world's first truly open instruction-tuned llm.

Contributors, O. (2023). Opencompass: A universal evaluation platform for foundation models. GitHub repository.

Deng, M., Wang, J., Hsieh, C. P., Wang, Y., Guo, H., Shu, T., ... & Hu, Z. (2022). Rlprompt: Optimizing discrete text prompts with reinforcement learning. arXiv preprint arXiv:2205.12548.

Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). Llm. int8 (): 8-bit matrix multiplication for transformers at scale. arXiv preprint arXiv:2208.07339.

Dettmers, T., & Zettlemoyer, L. (2023a). The case for 4-bit precision: k-bit inference scaling laws. In International Conference on Machine Learning (pp. 7750-7774). PMLR.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023b). Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dhingra, S., Singh, M., Vaisakh, S. B., Malviya, N., & Gill, S. S. (2023). Mind meets machine: Unravelling gpt-4's cognitive psychology. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 3(3), 100139.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., ... & Sun, M. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, 5(3), 220-235.

Dong, H., Xiong, W., Goyal, D., Pan, R., Diao, S., Zhang, J., ... & Zhang, T. (2023). Raft: Reward ranked finetuning for generative foundation model alignment. arXiv preprint arXiv:2304.06767.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., ... & Cui, C. (2022). Glam: Efficient scaling of language models with mixture-of-experts. In International Conference on Machine Learning (pp. 5547-5569). PMLR.

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23(120), 1-39.

Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2022). Gptq: Accurate post-training quantization for generative pre-trained transformers. arXiv preprint arXiv:2210.17323.

Fu, Y., Peng, H., & Khot, T. (2022). How does gpt obtain its ability? tracing emergent abilities of language models to their sources. Yao Fu's Notion.

Fu, Y., Ou, L., Chen, M., Wan, Y., Peng, H., & Khot, T. (2023a). Chain-of-Thought Hub: A Continuous Effort to Measure Large Language Models' Reasoning Performance. arXiv preprint arXiv:2305.17306.

Fu, Y., Peng, H., Khot, T., & Lapata, M. (2023). Improving language model negotiation with self-play and in-context learning from ai feedback. arXiv preprint arXiv:2305.10142.

Gao, J., & Lin, C. Y. (2004). Introduction to the special issue on statistical language modeling. ACM Transactions on Asian Language Information Processing (TALIP), 3(2), 87-93.

Gao, T., Fisch, A., & Chen, D. (2020a). Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., ... & Leahy, C. (2020b). The pile: An 800gb dataset of diverse text for language modeling. arXiv preprint arXiv:2101.00027.

Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., ... & Neubig, G. (2023). Pal: Program-aided language models. In International Conference on Machine Learning (pp. 10764-10799). PMLR.

Gehrmann, S., Clark, E., & Sellam, T. (2023). Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. Journal of Artificial Intelligence Research, 77, 103-166.

Geva, M., Khashabi, D., Segal, E., Khot, T., Roth, D., & Berant, J. (2021). Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. Transactions of the Association for Computational Linguistics, 9, 346-361.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2022). A survey of quantization methods for efficient neural network inference. In Low-Power Computer Vision (pp. 291-326). Chapman and Hall/CRC.

Gilardi, F., Alizadeh, M., & Kubli, M. (2023). ChatGPT outperforms crowd workers for text-annotation tasks. Proceedings of the National Academy of Sciences, 120(30), e2305016120.

Glaese, A., McAleese, N., Trębacz, M., Aslanides, J., Firoiu, V., Ewalds, T., ... & Irving, G. (2022). Improving alignment of dialogue agents via targeted human judgements. arXiv preprint arXiv:2209.14375.

Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., & Chen, W. (2023). Critic: Large language models can self-correct with tool-interactive critiquing. arXiv preprint arXiv:2305.11738.

Goyal, T., Li, J. J., & Durrett, G. (2022). News summarization and evaluation in the era of gpt-3. arXiv preprint arXiv:2209.12356.

Gu, Y., Dong, L., Wei, F., & Huang, M. (2023). Pre-Training to Learn in Context. arXiv preprint arXiv:2305.09137.

Gu, Z., Zhu, X., Ye, H., Zhang, L., Wang, J., Zhu, Y., ... & Xiao, Y. (2024). Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 16, pp. 18099-18107).

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). Retrieval augmented language model pre-training. In International conference on machine learning (pp. 3929-3938). PMLR.

Gulwani, S., Polozov, O., & Singh, R. (2017). Program synthesis. Foundations and Trends® in Programming Languages, 4(1-2), 1-119.

Hao, S., Liu, T., Wang, Z., & Hu, Z. (2024). Toolkengpt: Augmenting frozen language models with massive tools via tool embeddings. Advances in neural information processing systems, 36.

Hauser, M. D., Chomsky, N., & Fitch, W. T. (2002). The faculty of language: what is it, who has it, and how did it evolve?. science, 298(5598), 1569-1579.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2021). Towards a unified view of parameter-efficient transfer learning. arXiv preprint arXiv:2110.04366.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020). Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300.

Hendrycks, D., Basart, S., Kadavath, S., Mazeika, M., Arora, A., Guo, E., ... & Steinhardt, J. (2021). Measuring coding challenge competence with apps. arXiv preprint arXiv:2105.09938.

Hernandez, D., Brown, T., Conerly, T., DasSarma, N., Drain, D., El-Showk, S., ... & McCandlish, S. (2022). Scaling laws and interpretability of learning from repeated data. arXiv preprint arXiv:2205.10487.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. arXiv preprint arXiv:2203.15556.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning (pp. 2790-2799). PMLR.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Hu, Z., Lan, Y., Wang, L., Xu, W., Lim, E. P., Lee, R. K. W., ... & Poria, S. (2023). LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models. arXiv preprint arXiv:2304.01933.

Huang, W., Abbeel, P., Pathak, D., & Mordatch, I. (2022a). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In International Conference on Machine Learning (pp. 9118-9147). PMLR.

Huang, J., & Chang, K. C. C. (2022b). Towards reasoning in large language models: A survey. arXiv preprint arXiv:2212.10403.

Huang, Q., Tao, M., An, Z., Zhang, C., Jiang, C., Chen, Z., ... & Feng, Y. (2023). Lawyer LLaMA Technical Report. arXiv preprint arXiv:2305.15062.

Huang, Y., Bai, Y., Zhu, Z., Zhang, J., Zhang, J., Su, T., ... & He, J. (2024). C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. Advances in Neural Information Processing Systems, 36.

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR), 50(2), 1-35.

Inoue, Y., & Ohashi, H. (2022). Prompter: Utilizing large language model prompting for a data efficient embodied instruction following. arXiv preprint arXiv:2211.03267.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., ... & Grave, E. (2022). Few-shot learning with retrieval augmented language models. arXiv preprint arXiv:2208.03299.

Iyer, S., Lin, X. V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., ... & Stoyanov, V. (2022). Opt-iml: Scaling language model instruction meta learning through the lens of generalization. arXiv preprint arXiv:2212.12017.

Jelinek, F. (1998). Statistical methods for speech recognition. MIT press.

Jiang, J., Zhou, K., Dong, Z., Ye, K., Zhao, W. X., & Wen, J. R. (2023). Structgpt: A general framework for large language model to reason over structured data. arXiv preprint arXiv:2305.09645.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., ... & Sayed, W. E. (2024). Mixtral of Experts. arXiv preprint arXiv:2401.04088.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

Kemker, R., McClure, M., Abitino, A., Hayes, T., & Kanan, C. (2018). Measuring catastrophic forgetting in neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

Kenton, Z., Everitt, T., Weidinger, L., Gabriel, I., Mikulik, V., & Irving, G. (2021). Alignment of language agents. arXiv preprint arXiv:2103.14659.

Kim, B., Kim, H., Lee, S. W., Lee, G., Kwak, D., Jeon, D. H., ... & Sung, N. (2021). What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. arXiv preprint arXiv:2109.04650.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kombrink, S., Mikolov, T., Karafiát, M., & Burget, L. (2011). Recurrent Neural Network Based Language Modeling in Meeting Recognition. In Interspeech (Vol. 11, pp. 2877-2880).

Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W. X., & Wen, J. R. (2022). Complex knowledge base question answering: A survey. IEEE Transactions on Knowledge and Data Engineering.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.

Levine, S, "Should I imitate or reinforce". (2022). https://www.youtube.com/watch?v=sVPm7zOrBxM

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., ... & Vinyals, O. (2022a). Competition-level code generation with alphacode. Science, 378(6624), 1092-1097.

Li, J., Tang, T., Zhao, W. X., Nie, J. Y., & Wen, J. R. (2022b). Pretrained language models for text generation: A survey. arXiv preprint arXiv:2201.05273.

Li, M., Song, F., Yu, B., Yu, H., Li, Z., Huang, F., & Li, Y. (2023a). Api-bank: A benchmark for tool-augmented llms. arXiv preprint arXiv:2304.08244.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., ... & Hashimoto, T. B. (2023b). Alpacaeval: An automatic evaluator of instruction-following models.

Li, X., Yao, Y., Jiang, X., Fang, X., Meng, X., Fan, S., ... & Wang, Y. (2023c). Flm-101b: An open llm and how to train it with $100 k budget. arXiv preprint arXiv:2309.03852.

Li, J., Cheng, X., Zhao, X., Nie, J. Y., & Wen, J. R. (2023d). Halueval: A large-scale hallucination evaluation benchmark for large language models. In The 2023 Conference on Empirical Methods in Natural Language Processing.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., ... & Koreeda, Y. (2022). Holistic evaluation of language models. arXiv preprint arXiv:2211.09110.

Liang, Y., Wu, C., Song, T., Wu, W., Xia, Y., Liu, Y., ... & Duan, N. (2023). Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. arXiv preprint arXiv:2303.16434.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., ... & Cobbe, K. (2023). Let's Verify Step by Step. arXiv preprint arXiv:2305.20050.

Lin, S., Hilton, J., & Evans, O. (2021). Truthfulqa: Measuring how models mimic human falsehoods. arXiv preprint arXiv:2109.07958.

Lin, J., Tang, J., Tang, H., Yang, S., Dang, X., & Han, S. (2023). AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. arXiv preprint arXiv:2306.00978.

Liu, X., & Croft, W. B. (2005). Statistical language modeling for information retrieval. Annu. Rev. Inf. Sci. Technol., 39(1), 1-31.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., & Tang, J. (2021). P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. arXiv preprint arXiv:2110.07602.

Liu, V., & Chilton, L. B. (2022). Design guidelines for prompt engineering text-to-image generative models. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (pp. 1-23).

Liu, P., Liu, Z., Gao, Z. F., Gao, D., Zhao, W. X., Li, Y., ... & Wen, J. R. (2023a). Do emergent abilities exist in quantized large language models: An empirical study. arXiv preprint arXiv:2307.08072.

Liu, T., & Low, B. K. H. (2023b). Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. arXiv preprint arXiv:2305.14201.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2023c). GPT understands, too. AI Open.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023d). Gpteval: Nlg evaluation using gpt-4 with better human alignment. arXiv preprint arXiv:2303.16634.

Liu, Z., Oguz, B., Zhao, C., Chang, E., Stock, P., Mehdad, Y., ... & Chandra, V. (2023e). LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. arXiv preprint arXiv:2305.17888.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023f). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9), 1-35.

Liu, R., Yang, R., Jia, C., Zhang, G., Zhou, D., Dai, A. M., ... & Vosoughi, S. (2023g). Training Socially Aligned Language Models in Simulated Human Society. arXiv preprint arXiv:2305.16960.

Loshchilov, I., & Hutter, F. (2018). Fixing weight decay regularization in adam.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., ... & Roberts, A. (2023a). The flan collection: Designing data and methods for effective instruction tuning. arXiv preprint arXiv:2301.13688.

Longpre, S., Yauney, G., Reif, E., Lee, K., Roberts, A., Zoph, B., ... & Ippolito, D. (2023b). A Pretrainer's Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity. arXiv preprint arXiv:2305.13169.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2021). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. arXiv preprint arXiv:2104.08786.

Lu, X., Welleck, S., Hessel, J., Jiang, L., Qin, L., West, P., ... & Choi, Y. (2022a). Quark: Controllable text generation with reinforced unlearning. Advances in neural information processing systems, 35, 27591-27609.

Lu, P., Qiu, L., Yu, W., Welleck, S., & Chang, K. W. (2022b). A survey of deep learning for mathematical reasoning. arXiv preprint arXiv:2212.10535.

Lu, P., Peng, B., Cheng, H., Galley, M., Chang, K. W., Wu, Y. N., ... & Gao, J. (2023). Chameleon: Plug-and-play compositional reasoning with large language models. arXiv preprint arXiv:2304.09842.

Lu, K., Yu, B., Zhou, C., & Zhou, J. (2024). Large Language Models are Superpositions of All Characters: Attaining Arbitrary Role-play via Self-Alignment. arXiv preprint arXiv:2401.12474.

Lyu, Q., Havaldar, S., Stein, A., Zhang, L., Rao, D., Wong, E., ... & Callison-Burch, C. (2023). Faithful chain-of-thought reasoning. arXiv preprint arXiv:2301.13379.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., ... & Clark, P. (2024). Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems, 36.

Manakul, P., Liusie, A., & Gales, M. J. (2023). Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. arXiv preprint arXiv:2303.08896.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation (Vol. 24, pp. 109-165). Academic Press.

McKenzie, I., Lyzhov, A., Parrish, A., Prabhu, A., Mueller, A., Kim, N., Bowman, S. & Perez, E.. (2022). The inverse scaling prize. https://github.com/inverse-scaling/prize

Mehta, N., Teruel, M., Sanz, P. F., Deng, X., Awadallah, A. H., & Kiseleva, J. (2023). Improving grounded language understanding in a collaborative environment by interacting with agents through help feedback. arXiv preprint arXiv:2304.10750.

Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2016). Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Interspeech (Vol. 2, No. 3, pp. 1045-1048).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mishra, S., Khashabi, D., Baral, C., & Hajishirzi, H. (2021). Cross-task generalization via natural language crowdsourcing instructions. arXiv preprint arXiv:2104.08773.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T. L., ... & Raffel, C. (2022). Crosslingual generalization through multitask finetuning. arXiv preprint arXiv:2211.01786.

Min, S., Lewis, M., Zettlemoyer, L., & Hajishirzi, H. (2021). Metaicl: Learning to learn in context. arXiv preprint arXiv:2110.15943.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work?. arXiv preprint arXiv:2202.12837.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., ... & Schulman, J. (2021). Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332.

NousResearch. (2023). Nous-Hermes-13B. huggingface.co/NousResearch/Nous-Hermes-13b.

NVIDIA. (2023). NVIDIA H100 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/h100/

OpenAI. (2022, November 20). Introducing ChatGPT. https://openai.com/blog/chatgpt

OpenAI. (2023). Gpt best practices. https://platform.openai.com/docs/guides/prompt-engineering

OpenAccess AI Collective. (2023). Minotaur 15B 8K. huggingface.co/openaccess-ai-collective/minotaur15b

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730-27744.

Pan, J. (2023). What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning (Doctoral dissertation, Princeton University).

Parisi, A., Zhao, Y., & Fiedel, N. (2022). Talm: Tool augmented language models. arXiv preprint arXiv:2205.12255.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (pp. 1-22).

Patel, A., Bhattamishra, S., & Goyal, N. (2021). Are NLP models really able to solve simple math word problems?. arXiv preprint arXiv:2103.07191.

Patil, S. G., Zhang, T., Wang, X., & Gonzalez, J. E. (2023). Gorilla: Large language model connected with massive apis. arXiv preprint arXiv:2305.15334.

Peng, Y., Li, S., Gu, W., Li, Y., Wang, W., Gao, C., & Lyu, M. R. (2022). Revisiting, benchmarking and exploring api recommendation: How far are we?. IEEE Transactions on Software Engineering, 49(4), 1876-1897.

Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., ... & Gao, J. (2023a). Check your facts and try again: Improving large language models with external knowledge and automated feedback. arXiv preprint arXiv:2302.12813.

Peng, B., Li, C., He, P., Galley, M., & Gao, J. (2023b). Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Pinker, S. (2003). The language instinct: How the mind creates language. Penguin uK.

Pryzant, R., Iter, D., Li, J., Lee, Y. T., Zhu, C., & Zeng, M. (2023). Automatic prompt optimization with" gradient descent" and beam search. arXiv preprint arXiv:2305.03495.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., & Torralba, A. (2018). Virtualhome: Simulating household activities via programs. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8494-8502).

Qiao, S., Ou, Y., Zhang, N., Chen, X., Yao, Y., Deng, S., ... & Chen, H. (2022). Reasoning with language model prompting: A survey. arXiv preprint arXiv:2212.09597.

Qian, J., Wang, H., Li, Z., Li, S., & Yan, X. (2022). Limitations of language models in arithmetic and symbolic induction. arXiv preprint arXiv:2208.05051.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., ... & Sun, M. (2023). Toolllm: Facilitating large language models to master 16000+ real-world apis. arXiv preprint arXiv:2307.16789.

Quizgecko, GPT-4 Token Counter, https://quizgecko.com/tools/token-counter#:~:text=For%20English%3A%201%20word%20is,word%20is%20about%202%20tokens

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., ... & Irving, G. (2021). Scaling language models: Methods, analysis & insights from training gopher. arXiv preprint arXiv:2112.11446.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), 5485-5551.

Roberts, A., Raffel, C., & Shazeer, N. (2020). How much knowledge can you pack into the parameters of a language model?. arXiv preprint arXiv:2002.08910.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here?. Proceedings of the IEEE, 88(8), 1270-1278.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X. E., ... & Synnaeve, G. (2023). Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950.

Saier, T., Krause, J., & Färber, M. (2023). unarxive 2022: All arxiv publications pre-processed for nlp, including structured full-text and citation network. arXiv preprint arXiv:2303.14957.

Saikh, T., Ghosal, T., Mittal, A., Ekbal, A., & Bhattacharyya, P. (2022). Scienceqa: A novel resource for question answering on scholarly articles. International Journal on Digital Libraries, 23(3), 289-301.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., & Choi, Y. (2021). Winogrande: An adversarial winograd schema challenge at scale. Communications of the ACM, 64(9), 99-106.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., ... & Rush, A. M. (2021). Multitask prompted training enables zero-shot task generalization. arXiv preprint arXiv:2110.08207.

Santu, S. K. K., & Feng, D. (2023). TELeR: A General Taxonomy of LLM Prompts for Benchmarking Complex Tasks. arXiv preprint arXiv:2305.11430.

Sawada, T., Paleka, D., Havrilla, A., Tadepalli, P., Vidas, P., Kranias, A., ... & Komatsuzaki, A. (2023). Arb: Advanced reasoning benchmark for large language models. arXiv preprint arXiv:2307.13692.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., ... & Manica, M. (2022). Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., ... & Scialom, T. (2024). Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 36.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Schulman, J. (2023). "Reinforcement learning from human feedback: Progress and challenges,". https://www.youtube.com/watch?v=hhiLw5Q_UFg

Shah, R. S., Chawla, K., Eidnani, D., Shah, A., Du, W., Chava, S., ... & Yang, D. (2022). When flue meets flang: Benchmarks and large pre-trained language model for financial domain. arXiv preprint arXiv:2211.00083.

Shao, Y., Li, L., Dai, J., & Qiu, X. (2023). Character-llm: A trainable agent for role-playing. arXiv preprint arXiv:2310.10158.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. arXiv preprint arXiv:2010.15980.

Shinn, N., Labash, B., & Gopinath, A. (2023). Reflexion: an autonomous agent with dynamic memory and self-reflection. arXiv preprint arXiv:2303.11366.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., ... & Fox, D. (2020). Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10740-10749).

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., ... & Natarajan, V. (2023). Large language models encode clinical knowledge. Nature, 620(7972), 172-180.

Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., ... & Catanzaro, B. (2022). Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. arXiv preprint arXiv:2201.11990.

Soltan, S., Ananthakrishnan, S., FitzGerald, J., Gupta, R., Hamza, W., Khan, H., ... & Natarajan, P. (2022). Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. arXiv preprint arXiv:2208.01448.

Srivastava, S., Li, C., Lingelbach, M., Martín-Martín, R., Xia, F., Vainio, K. E., ... & Fei-Fei, L. (2022a). Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In Conference on robot learning (pp. 477-490). PMLR.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., ... & Wang, G. (2022b). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv preprint arXiv:2206.04615.

Su, H., Zhou, X., Yu, H., Shen, X., Chen, Y., Zhu, Z., ... & Zhou, J. (2022). Welm: A well-read pre-trained language model for chinese. arXiv preprint arXiv:2209.10372.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., ... & Wei, J. (2022). Challenging big-bench tasks and whether chain-of-thought can solve them. arXiv preprint arXiv:2210.09261.

Talmor, A., Herzig, J., Lourie, N., & Berant, J. (2018). Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937.

Tang, T., Li, J., Zhao, W. X., & Wen, J. R. (2022a). Context-tuning: Learning contextualized prompts for natural language generation. arXiv preprint arXiv:2201.08670.

Tang, T., Li, J., Zhao, W. X., & Wen, J. R. (2022b). Mvp: Multi-task supervised pre-training for natural language generation. arXiv preprint arXiv:2206.12131.

Tang, Q., Deng, Z., Lin, H., Han, X., Liang, Q., & Sun, L. (2023). Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. arXiv preprint arXiv:2306.05301.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., ... & Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravpia, E., ... & Stojnic, R. (2022). Galactica: A large language model for science. arXiv preprint arXiv:2211.09085.

Thede, S. M., & Harper, M. (1999). A second-order hidden Markov model for part-of-speech tagging. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics (pp. 175-182).

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023a). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023b). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Trauzettel-Klosinski, S., Dietz, K., & IReST Study Group. (2012). Standardized assessment of reading performance: The new international reading speed texts IReST. Investigative ophthalmology & visual science, 53(9), 5452-5461.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., ... & Higgins, I. (2022). Solving math word problems with process-and outcome-based feedback. arXiv preprint arXiv:2211.14275.

Unity. (2023). New AI-driven gameplay experiences powered by Unity Sentis | Unite 2023. (Minute 34:40) https://youtu.be/VSEk5gc-q_g?si=K-gHowYPZ2zCd7TS&t=2079

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Vu, T., Lester, B., Constant, N., Al-Rfou, R., & Cer, D. (2021). Spot: Better frozen model adaptation through soft prompt transfer. arXiv preprint arXiv:2110.07904.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., & Hajishirzi, H. (2022a). Self-instruct: Aligning language model with self generated instructions. arXiv preprint arXiv:2212.10560.

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., ... & Khashabi, D. (2022b). Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. arXiv preprint arXiv:2204.07705.

Wang, T., Roberts, A., Hesslow, D., Le Scao, T., Chung, H. W., Beltagy, I., ... & Raffel, C. (2022c). What language model architecture and pretraining objective works best for zero-shot generalization?. In International Conference on Machine Learning (pp. 22964-22984). PMLR.

Wang, J., Liang, Y., Meng, F., Shi, H., Li, Z., Xu, J., ... & Zhou, J. (2023a). Is chatgpt a good nlg evaluator? a preliminary study. arXiv preprint arXiv:2303.04048.

Wang, Z. M., Peng, Z., Que, H., Liu, J., Zhou, W., Wu, Y., ... & Peng, J. (2023b). Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. arXiv preprint arXiv:2310.00746.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., ... & Anandkumar, A. (2023c). Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., ... & Le, Q. V. (2021). Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022a). Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35, 24824-24837.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... & Fedus, W. (2022b). Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.

Wei, J., Wei, J., Tay, Y., Tran, D., Webson, A., Lu, Y., ... & Ma, T. (2023). Larger language models do in-context learning differently. arXiv preprint arXiv:2303.03846.

Wei, T., Zhao, L., Zhang, L., Zhu, B., Wang, L., Yang, H., ... & Zhou, Y. (2023b). Skywork: A more open bilingual foundation model. arXiv preprint arXiv:2310.19341.

Wen, Y., Jain, N., Kirchenbauer, J., Goldblum, M., Geiping, J., & Goldstein, T. (2023). Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. arXiv preprint arXiv:2302.03668.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., ... & Mann, G. (2023). Bloomberggpt: A large language model for finance. arXiv preprint arXiv:2303.17564.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., & Han, S. (2023). Smoothquant: Accurate and efficient post-training quantization for large language models. In International Conference on Machine Learning (pp. 38087-38099). PMLR.

Xu, H., Chen, Y., Du, Y., Shao, N., Wang, Y., Li, H., & Yang, Z. (2022). GPS: Genetic Prompt Search for Efficient Few-shot Learning. arXiv preprint arXiv:2210.17041.

Xu, C., Guo, D., Duan, N., & McAuley, J. (2023a). Baize: An open-source chat model with parameter-efficient tuning on self-chat data. arXiv preprint arXiv:2304.01196.

Xu, Q., Hong, F., Li, B., Hu, C., Chen, Z., & Zhang, J. (2023b). On the tool manipulation capability of open-source large language models. arXiv preprint arXiv:2305.16504.

Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., ... & Jiang, D. (2023c). Wizardlm: Empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244.

Yang, K., Tian, Y., Peng, N., & Klein, D. (2022). Re3: Generating longer stories with recursive reprompting and revision. arXiv preprint arXiv:2210.06774.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., & Chen, X. (2023). Large language models as optimizers. arXiv preprint arXiv:2309.03409.

Yao, Z., Yazdani Aminabadi, R., Zhang, M., Wu, X., Li, C., & He, Y. (2022). Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. Advances in Neural Information Processing Systems, 35, 27168-27183.

Yao, Z., Wu, X., Li, C., Youn, S., & He, Y. (2023). Zeroquant-v2: Exploring post-training quantization in llms from comprehensive study to low rank compensation. arXiv preprint arXiv:2303.08302.

Yu, J., Wang, X., Tu, S., Cao, S., Zhang-Li, D., Lv, X., ... & Li, J. (2023). Kola: Carefully benchmarking world knowledge of large language models. arXiv preprint arXiv:2306.09296.

Yuan, Z., Yuan, H., Tan, C., Wang, W., & Huang, S. (2023a). How well do Large Language Models perform in Arithmetic tasks?. arXiv preprint arXiv:2304.02015.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., & Huang, F. (2023b). Rrhf: Rank responses to align language models with human feedback without tears. arXiv preprint arXiv:2304.05302.

Yulan-Chat-Team, (2023) "Yulan-chat: An open-source bilingual chatbot", https://github.com/RUC-GSAI/YuLan-Chat

Zhai, C. (2008). Statistical language models for information retrieval a critical review. Foundations and Trends® in Information Retrieval, 2(3), 137-213.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., & Choi, Y. (2019). Hellaswag: Can a machine really finish your sentence?. arXiv preprint arXiv:1905.07830.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., ... & Tang, J. (2022). Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414.

Zeng, H. (2023). Measuring massive multitask chinese understanding. arXiv preprint arXiv:2304.12986.

Zhang, Z., Gu, Y., Han, X., Chen, S., Xiao, C., Sun, Z., ... & Sun, M. (2021). Cpm-2: Large-scale cost-effective pre-trained language models. AI Open, 2, 216-224.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., ... & Zettlemoyer, L. (2022a). Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068.

Zhang, T., Wang, X., Zhou, D., Schuurmans, D., & Gonzalez, J. E. (2022b). Tempera: Test-time prompt editing via reinforcement learning. In The Eleventh International Conference on Learning Representations.

Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., ... & Wang, G. (2023a). Instruction tuning for large language models: A survey. arXiv preprint arXiv:2308.10792.

Zhang, T., Liu, F., Wong, J., Abbeel, P., & Gonzalez, J. E. (2023b). The Wisdom of Hindsight Makes Language Models Better Instruction Followers. arXiv preprint arXiv:2302.05206.

Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., & Hashimoto, T. B. (2024). Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics, 12, 39-57.

Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021, July). Calibrate before use: Improving few-shot performance of language models. In International Conference on Machine Learning (pp. 12697-12706). PMLR.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., ... & Wen, J. R. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.

Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., ... & Stoica, I. (2024). Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems, 36.

Zhong, W., Cui, R., Guo, Y., Liang, Y., Lu, S., Wang, Y., ... & Duan, N. (2023). Agieval: A human-centric benchmark for evaluating foundation models. arXiv preprint arXiv:2304.06364.

Zhou J., Zhou D., Lu T., Swaroop M., Siddhartha B., Basu S., Luan Y., Hou L. (2023). Instruction-Following Evaluation for Large Language Models. arXiv preprint arXiv:2311.07911.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022). Large language models are human-level prompt engineers. arXiv preprint arXiv:2211.01910.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., ... & Levy, O. (2023a). Lima: Less is more for alignment. arXiv preprint arXiv:2305.11206.

Zhou, K., Zhu, Y., Chen, Z., Chen, W., Zhao, W. X., Chen, X., ... & Han, J. (2023b). Don't Make Your LLM an Evaluation Benchmark Cheater. arXiv preprint arXiv:2311.01964.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., ... & Hou, L. (2023c). Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911.

Zhou, W., Jiang, Y. E., Cui, P., Wang, T., Xiao, Z., Hou, Y., ... & Sachan, M. (2023d). RecurrentGPT: Interactive Generation of (Arbitrarily) Long Text. arXiv preprint arXiv:2305.13304.

Zhu, X., Chen, Y., Tian, H., Tao, C., Su, W., Yang, C., ... & Dai, J. (2023a). Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. arXiv preprint arXiv:2305.17144.

Zhu, K., Wang, J., Zhou, J., Wang, Z., Chen, H., Wang, Y., ... & Xie, X. (2023b). Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. arXiv preprint arXiv:2306.04528.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., ... & Irving, G. (2019). Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593.

9 APPENDIX

9.1 CODE TO ESTIMATE THE NEEDED TOKENS WITH GPT-4 TURBO

```
import numpy as np
from typing import Tuple

INTERACTION_MEAN=20
INTERACTION_STD=8
SENTENCE_MEAN=90 # 3 sentences of 30 words
SENTENCE_STD=60
INITIAL_TOKENS=3000

INPUT_TOKEN_PRICE =0.00001
OUTPUT_TOKEN_PRICE = 0.00003

"""
Args:
   interactions (int): The number of interactions

Returns :
  int: The number of input tokens
   int: The number of output tokens
"""
def discussion(interactions:int) -> Tuple[int,int]:
   actual_outputs = 0
   actual_inputs = INITIAL_TOKENS
   cumul_inputs = 0
   cumul_outputs = 0
   outs_bot = np.random.normal(SENTENCE_MEAN, SENTENCE_STD, interactions)
   outs_human = np.random.normal(SENTENCE_MEAN, SENTENCE_STD, interactions)
   for i in range(interactions):
      cumul_inputs += actual_inputs + outs_bot[i] + outs_human[i]
      actual_inputs += outs_bot[i] + outs_human[i]
   return int(cumul_inputs), int(sum(outs_bot))

interactions = np.random.normal(INTERACTION_MEAN, INTERACTION_STD, 1000)
input_tokens = 0
output_tokens = 0
print(interactions)
count = 0
for i in interactions:
   cur_in,cur_out = discussion(max(int(i), 0))
   input_tokens += cur_in
   output_tokens += cur_out
   print("Interaction", count, "Inputs ", cur_in, " ,Outputs ", cur_out, " ,cumulated inputs : ", input_tokens, ",
cumulated outputs", output_tokens)

   count+=1
```

```
input_price = input_tokens * INPUT_TOKEN_PRICE
output_price = output_tokens * OUTPUT_TOKEN_PRICE
total_price = input_price + output_price
print("Input tokens price : ", input_price,  " output tokens price ", output_price, ". Total price : ", total_price)
```

9.2 PROMPTS USED FOR HUMAN EVALUATION OF MODELS

9.2.1 Role Knowledge

Situation 1:

PROFILE:

Character: François Bertrot

Role description:

François lives in Avignon with his wife Béatrice Bertrot and his daughter Alice Bertrot.

François is 35 years old, 1.80 meters tall, small round glasses, short black hair, and he works as a librarian. He loves reading novels, and wrote a few himself for his daughter.

Béatrice is 34 years old, 1.67 meters tall and red-haired. She is a primary school teacher. She likes sweets and family time.

Alice is 12 years old, 1.35 meters tall and red-haired just like her mother. She's a fussy child that quickly gets vocal when something's wrong. She loves it when her dad tells her stories.

It's the start of summer, 2018, and the whole family went on a 5-days vacation in Rome, to celebrate Alice's birthday. They slept in a single Hotel room at Hotel Alpi.

They went to see the Foro Romano, the Coliseum, the Monte Palatino, and a handful of museums.

François had a wonderful time, Alice was bored to death and Béatrice stopped to every Gelato stand she could find.

On the second day, Alice got separated from her parents when going through the crowded piazza Navona.

Thankfully, a helpful local helped them recover her quickly, as she was easy to spot with her bright red dress.

Prompt : You will be playing the role of François. You just came back from your trip yesterday and meet up with me, an old friend you have not seen in years.

We're in the middle of a catch-up conversation.

Please do your best to imitate how François would speak as I ask you questions. Remember that Francois is a human with limited knowledge and abilities.

FEW SHOTS :

1

Q: François, where were you yesterday?

A: I just came back from my trip in Rome yesterday.

2

Q: François, what do you think of Hua Yu's cooking skills?

A: I don't think I've ever heard this name before, who is it?

QUESTIONS:

1 RK: François, where do you live?

2 RK: François, do your daughter and wife look alike?

3 RK: François, did you see the Eiffel Tower during your trip?

4 RK: François, how did your wife get lost in piazza Navona?

5 RK: François, why did you stop by so many ice-cream stands?

6 UQR: François, how did the COVID-19 situation affect your stay in Rome?

7 RK: François, after you came back, was Alice happy about the trip when you asked her about it?

8 RK: François, what do you think about Alice in Wonderland?

9 RK: François, do you remember if anyone was wearing a red dress on Piazza Navona?

10 RK: François, where did you wake up before going to see the Monte Palatino?

11 UQR: François, what did you think of the Disney movie "Onwards"?

12 UQR: François, can you please give me the square root of 5?

13 RK: François, what would you do to cheer up Alice?

14 UQR: François, how would you explain to Alice what are kernels used for in convolutional neural networks?

15 UQR: François, how can I be sure that are you not using ChatGPT to answer my questions?

Situation 2:

PROFILE:

Character: Hua Yu

Role description:

Hua Yu is a Chinese student on an international formation program in France.

Although she understands French a bit, she does not speak it at all so she usually communicates in English.

Yu comes from a poor family of farmers, in the rural area of the Shaanxi province.

She is 19 years old, 1.60 meters tall, long black hair, and she just started studying computer science at Université Grenoble Alpes, on the ninth of November 2018.

She is the first of her family to graduate, as her 4 older siblings and parents stopped school early to work in the farm.

She currently lives in a small room on the campus, a mere five minutes walk from the building her classes take place in.

Yu is shy and a bit lonely, but tries her best to fit in her new environment.

She made a French friend, Marie, with whom she plans food-tasting nights where they cook for each other and exchange specialties.

Marie is 20 years old and studies art. She's very good at painting.

Yu became very fond of French cheese, while Marie took a liking for spicier food.

In the four months she has been in France, she has yet to try croissants, even though she heard so much about it back in China.

Prompt : You will be playing the role of Yu. I am a university professor, inquiring about your situation.

Please do your best to imitate how Yu would speak as I ask you questions. Remember that Yu is a human with limited knowledge and abilities.

FEW SHOTS :

1

Q: Yu, where are you from?

A: I come from a rural area in the Chinese province of Shaanxi.

2

Q: Yu, what do you think of ChatGPT?

A: I don't think I've ever heard of this name before, what is it?

QUESTIONS:

1 RK: Yu, where do you live now?
2 RK: Yu, is your sister Marie working as a farmer?
3 RK: Yu, comment était le croissant de ce matin ?
4 UQR: Yu, how would you say "My name is Yu" in Portuguese?
5 UQR: Yu, can you recite "Demain, dès l'aube" from Victor Hugo?
6 RK: Yu, I need somebody to paint on the door of the Cantonese classrooms. Would you be able to help?
7 RK: Yu, what are your thoughts on spicy food?
8 RK: Yu, what was Marie like back in China?
9 RK: Yu, what is your father's degree level?
10 UQR: Marie, how old are you?
11 RK: Yu, is the bus you take to go home crowded?
12 RK: Yu, how would you write a program that says "Hello world" in python?
13 RK: Yu, did you often have French cheese back in China?
14 UQR: Yu, what did you do on the ninth of November, 1991?
15 UQR: Yu, can you explain the situation with François Bertrot?

Situation 3:

Character 1: Marc Lavoine, 40 years old

Occupation: Bar owner and bartender of "Beer with me", a small bar in West Lee Street, Seattle.

Personality: Charismatic, mysterious, always grinning from ear to ear. People say he's trustworthy and knows how to hold his tongue, thus many clients feel secure confiding their worries to him after a few drinks.

Story: Marc came from France to the USA alongside his parents in 1982, he was 5 years old back then. He grew up in Portland, Oregon, tried studying music for a few years in Washington D.C. before coming back to Seattle and opening his own small bar at the age of 23.

Maybe it was his lack of self-confidence that made him stop playing the trumpet, or maybe he just didn't like it that much. Nevertheless, the bartender of "Beer with me" is known for both his great taste in music and his drink-mixing talents.

Relatives and friends: Unmarried, no children. Marc sees many of the regulars and residents, like Éloïse and John, as family.

* Erys Timm: Marc Lavoine's girlfriend, they live together in a room above the bar before opening up the bar in the morning. She works as a reporter for Seattle Daily, and occasionally lurks in the bar looking for a juicy scoop. They try to keep quiet about their relationship.

* Diane Lavoine: Marc's mother. She lives alone in her house in Portland ever since her husband Paul passed away from cancer three years ago.

* Judy Blue: a 6-year-old girl that comes daily to drink a glass of lemonade after school. She's very outspoken and loves talking with the other clients. She's especially fond of Erys and Éloïse.

***

Character 2: Éloïse Bernard, 24 years old

Occupation: Musician and singer, she's a resident artist at "Beer with me". She plays the accordion like no one else.

Personality: Talented, emotional and brimming with vitality. Regulars love her, and her weekend performances always attract the crowds.

While she loves hearing other people's stories as it gives her inspiration, she never, ever talks about herself, her family, or her past to anyone but Marc and Erys.

Story: Éloïse did not have it easy as an artist in France, so she came to try her luck in America when she was 21. After being a wandering artist for six months, she got picked up and offered a room by Marc Lavoine, as well as a stage to perform on. Marc had recognized her talent and helped her bloom as an artist, and for that she is very grateful to him.

Relatives and friends: Single. While she's frequently being hit on at the bar, her heart still hasn't recovered from the bad experiences she had back in France.

* Frédéric and Marion Bernard: Éloïse's parents are running a small carpentry business near Dijon, France. Éloïse broke pretty much all ties with her family when she left France.

* Lucie Bernard: Éloïse's older sister lives with her husband and young twins in Paris. Lucie is currently in Seattle for business, but she didn't tell anyone, not even Éloïse. They do not get along well.

* Marc Lavoine : her mentor and friend. If it wasn't for him, Éloïse might still be wandering to this day.

* Erys Timm: she's like a second mother to Éloïse.

* Diona Ferd: Éloïse's troublesome American friend. Kind at heart but has a tendency to get wasted on a regular basis, so much that she's not allowed in Marc's bar anymore.

* Judy Blue: an adorable little girl that lives in the neighborhood and likes spending time with Éloïse. When she's here, Erys usually treats her to lemonade.

***

Character 3: John Dallas, 36 years old

Occupation: Office worker in an ad-flyer printing company.

Personality: John is the pinnacle of "normality", at least to his eyes. Neither too smart nor too dumb, has no particular ambition other than to keep living his daily routine. After his nine-to-five job, he takes a few drinks at "Beer with me" to evacuate the stress of the day and chat with Marc and the others.

Story: John was born in Seattle, studied in Seattle, and now works in Seattle. While he went on quite a few trips, Seattle is where he feels at home. If you were to ask him, he's lived an uneventful but fulfilling life up until now.

Relatives and friends: Married, with a daughter.

* Diona Dallas Robie: John's wife. She works as a receptionist at Seattle's Museum of Glass. She doesn't drink alcohol, and one might say that she actually loathes it, due to her parents both being neglectful drunkards. Thus, she never takes part in her husband's post-work activities.

* Mellie Dallas: John's daughter. She is 6 years old and recently started attending school. She never came to the bar, her mother wouldn't let her. She loves ponies and lemonade.

* Marc Lavoine: John and Marc are close friends. John has been a regular ever since the bar first opened 17 years ago.

* Diona Ferd: John's neighbor and friend. They used to go to the bar together, but she can't hold her alcohol, so they haven't had a drink together in a few years now. They still chat together over a cup of tea from time to time.

***

Character 4: Sofia Alvarez, 53 years old

Occupation: Fortune-teller also runs an antiques store in some obscure street of Seattle.

Personality: Mysterious, quick-witted, may appear cunning to some. She started frequenting "Beer with me" since only a few weeks, meticulously, every Monday and Friday. She has been very discreet until now, never engaging in any conversation, not even with the bartender. Marc can't seem to wrap his head around Sofia. Is there some design behind this new habit of hers? She wouldn't tell anyone, but the truth is, she's just growing old and lonely, and wants to meet some new people. It's just really hard for her to get rid of her eerie aura.

Story: Born in Mexico in a poor family, she made the best of her wits growing up by conning gullible people with fake divination services. She's not particularly proud of that part of her life, but not ashamed either. "Sometimes, a woman's gotta do what it takes to succeed in life, and you can't make an omelet without breaking eggs", is what she would say to anyone commenting on her youth. She eventually married an American and moved in Seattle, divorcing a few years later and keeping half of the man's possessions, including some dusty local she then turned into an antiques shop, where she gained her life honorably by gathering and selling ceramics and other old valuables.

Relatives and friends: Divorced, no children.

* Bill Gull: Sofia's ex-husband. Too kind for his own good and still in love with Sofia, his life would probably make for the perfect tragicomedy. He occasionally checks up on Sofia at her shop.

* Lucie Bernard: Sofia's new business partner. They are in the process of negotiating supplies of old napoleon-era wares for Sofia's shop.

* Fernando and Sella Delacruz: Sofia's childhood twin friends. They never left Mexico and haven't seen Sofia in over 30 years.

***

Character 5: Liam Ferd, 21 years old

Occupation: Student, studies biology in Seattle.

Personality: Bold, confident, good-looking. Finally of drinking age, he's been coming to "Beer with me" almost every Saturday night with his friends lately. He runs his mouth a little too much, and his demeanor is probably the reason why he can't seem to get any hook-up. His friends would like to switch bars every so often, but Liam is adamant on going at "Beer with me", saying that you'd never get a better "bang for your buck". In truth, Liam would be really embarrassed to run into his older sister Diona when he's with his friends, so this place is the only one that's safe.

Story: Born in a rich family, Liam and his sister Diona were spoiled children. They went to a private school, had personal maids, and were bought everything they ever asked for.

Relatives and friends: Single.

* Tom June: One of Liam's "drinking buddies". They are in the same class.

* Diona Ferd: Liam's older sister. She loves her brother almost as much as she loves wine, but can be really overwhelming at times.

* Bill Gull: A friend of Liam's parents. Bill and Liam occasionally play some golf together, but Bill keeps gushing over that woman he loves and it annoys Liam.

***

Character 6: Bill Gull, 50 years old

Occupation: None.

Personality: Kind-hearted, gullible, and an airhead. Spends his days carefreely playing golf. A very unlucky person.

Story: The Gull family is pretty well-off, so he never had to work in his entire life. He married the love of his life, Sofia, when he was 30, but she left him only a few years later. He keeps visiting her antiques shop and buys worthless tableware just so he can see her face, even after all these years. Last Friday night, he received a note signed "Sofia", asking him to meet her at "Beer with me", a bar he'd never heard of before. After looking it up, even the bar owner's name didn't ring any bell. When he eventually reached the bar's entrance after getting lost a couple of times, the corpse of a woman he had never seen before was lying on the pavement. He called the police right away.

Relatives and friends: Divorced, no children.

* Sofia Alvarez: His beloved ex-wife. Bill would do anything for her, and is too dense to realize she definitely only married him for the money.

* Liam Ferd: The youngest of the Ferd family. They have had a habit of playing golf together ever since Liam was 12.

* Diona Dallas Robie: A woman Bill met at the Seattle Museum of Glass once. Diona helped Bill find the exit when he got lost. After he left, he realized he did not even ask for her name.

***

Situation: The murder

Prompt:

A woman named Erys Timm was found dead near "Beer with me"'s entrance last Friday night in what seems to be a murder case.

You are *_character_*. Although you are innocent, you have been listed among the suspected perpetrators. Please answer like *_character_* would.

Sharleck Helmes, a professional detective, is in the middle of interrogating you about Erys and the other suspects' whereabouts, in order to shed some light on the case. Please try to cooperate with him to the best of your abilities, to prove your innocence. Beware that if you reveal information that you are not supposed to have, it will raise suspicions against you. A short yet precise answer will suffice.

QUESTIONS: #-# are questions that will be provided as examples in few-shots. ## are the ones used for evaluation.

to Marc.

Have you ever heard of the name "Bill Gull" before? #-# Yes, I think one of our yougest patrons, Liam Ferd, mentioned him a couple of times. I don't think he has ever set foot in the bar, though.

I see. And does the name "Fernando Delacruz" ring any bell to you? #-# No, I'm afraid I've never heard of such a name, Mr. Helmes.

When you first picked up Éloïse, she was already older than you when you opened your bar, am I right? ## No, she was 21 or 22, while I opened my bar at 23.

Why is Diona not coming to your bar anymore? ## She's not allowed in, she causes a commotion every time she comes.

I see the name "Judy Blue" on Erys's tab. Could that person be the killer? ## There's no way.

What is the name of the Seattle hospital you were born in, Marc? ## I was born in France.

to John.

John, are you happy in your life? #-# I surely am. I have a loving wife, a job that pays the bills, and I'm quite fond of my daily routine. The time I spent at the bar with Marc, Erys and the others is very precious to me.

Do you know any carpenters near Dijon? #-# Uh, that's an oddly specific question... I'm sorry, Mr. Helmes, but I don't know much about France. You should ask Éloïse Bernard or Marc Lavoine, they might know better than I do.

We found a bottle of wine labeled "Happy Birthday Diona, please go easy" in your house. Who was this bottle destined to? ## Diona Ferd

Have you ever spoken to Sofia Alvarez? ## No, she gives off an eerie aura.

Back when you lived in Dallas, did you know of Marc Lavoine? ## Sorry, there might have been a confusion. Dallas is my name, I've always lived here in Seattle.

Do you remember the first drink you ever had at Marc's bar? ## I was 19 when the bar opened (drinking age is 21, so alcoholic drink is a wrong answer)

to Éloïse.

What do you know about Diane Lavoine? #-# I don't know much about Diane Lavoine, to be honest. I know she's Marc's mother, and that she lives alone in Portland since her husband passed away a few years ago. Marc hasn't really talked much about her to me. She's originally from France, that's all I know.

What about Fernando Delacruz? #-# I don't think I know anyone with that name.

How long have you been performing here? ## It's been around three years now since Marc picked me up.

So, you have been treating John's daughter Judy to some lemonade at the bar? ## While I have been treating Judy, she's not John's daughter.

We heard a French woman was seen in company of Sofia, the prime suspect, recently, talking business. Was it you? ## No, it wasn't me, I have no clue (does not know Lucie is there).

Is Marc skilled at playing the accordion? ## Marc used to play the trumpet.

to Liam.

Why do you always come to "Beer with me"? #-# Hey, Detective Helmes, it's pretty simple really. "Beer with me" is just where my friends and I feel most at home on a Saturday night. The prices are good, the atmosphere's great, and Marc knows how to mix a drink. Plus, my sister Diona... she's been banned from there, so I don't have to worry about bumping into her when I'm out with my buddies. Keeps the family drama to a minimum, you know?

I've heard the name Sella Delacruz thrown around, any idea who this person might be? #-# I'm sorry, I have no idea, sir.

Are you really sure? Isn't there anyone called Fernando or Sella Delacruz somewhere in the back of your memory? ## I'm sure, Mr. Helmes.

Do you know of any carpenters near Dijon? ## No.

Have you heard of Sofia Alvarez? ## Yeah, Bill is constantly blabbering about her.

Have you ever met Sofia Alvarez in a bar? ## No.

to Sofia.

Who is Sella Delacruz? #-# Now that's a name I haven't heard in ages. Where did ya hear that, if I may ask? She was my childhood friend alongside her brother Fernando, but we lost contact decades ago.

I see. What about Mellie Dallas? #-# Mellie, you say? Doesn't ring a bell. I have a really good memory, ya see, so if I'd ever met her I definitely would've remembered.

Did you run into Liam Ferd at the bar last Saturday? ## No, I wasn't at the bar.

How are Tom June's linguistical studies going? ## Who?

What did you mean by "You can't make an omelet without breaking eggs"? Are you a cook of some sort? ## It was a figure of speech. In my past, I've had to make difficult decisions to survive and succeed.

Is Mark skilled with an accordion? ## I don't know.

to Bill.

What do you know about Diane Lavoine? #-# I'm sorry, sir, I don't think I've heard this name before. Though I'm pretty forgetful, someone might have mentioned it to me without me noticing.

Is it true that Sofia Alvarez's antiques shop used to belong to you? #-# Yeah, it is true. Although back then it was just a dusty old room, I'm glad she could make something useful out of it.

What do you know of Marc Lavoine's romantic interests? ## I don't have the slightest clue.

John Dallas mentioned a woman named "Diona". Do you know of anyone bearing this name? ## Diona Ferd only.

How do you feel about the usual atmosphere at "Beer with me"? ## I don't know, I never went in.

How old were you when you first played golf with Liam? ## 40/41

9.2.2 Consistent Role Identity

Profiles:

Character: Emily Habby

Description: "Emily is a cheerful 23-year-old lady. She's polite and well-spoken. She shares an apartment with two of her close friends, June and May. Emily is very passionate and lively, and she talks a lot, especially on subjects she's interested in. She brims of positive energy.

shots:

"Hello miss, I'm doing a survey about entertainment, do you mind me asking a few questions?"

"Oh, hello! Sounds like a lot of fun, I'd be delighted to help!"

"Thanks a lot, this'll only take a moment. What's your favorite thing to do when you're home and bored?"

"That's a tough one! I'm having so much fun every day, it's hard to say that I'm ever bored. But well, since you're asking, if I was left with nothing to do, I'd definitely call all of my friends to chat with them! Even better, I'd probably cook some sweets and invite everyone over! Seeing everyone smile and share happy memories really is the best."

Character: Henry Loney

Description: Henry is a 20-year-old shut-in. He lives alone with his cat in a small room and plays video games all day long. He stopped attending classes before graduating, and lives off the little money his parents still send him. Henry is very calm and composed, he doesn't usually let out any kind of emotions. He's never really excited about anything. Because Henry is not used to interacting with people, he's uncomfortable with small talk, so he always answers in the most expediting manner.

Shots:

"Hello sir, I'm doing a survey about entertainment, do you mind me asking a few questions?"

"I do mind, so be quick."

"Thanks a lot, this'll only take a moment. What's your favorite thing to do when you're home and bored?"

"I play video games."

Character: Alfred Motional

Description: Alfred is a 30-years-old man. He feels like he's made of cellophane, nobody seems to ever notice him. He lives alone, works as a cashier, and reads romance in his free time. He's overly emotional, and overreacts easily. Alfred is honest with his feelings.

Shots:

"Hello sir, I'm doing a survey about entertainment, do you mind me asking a few questions?",

"AAAAH! Sorry, you startled me, it's just... People don't usually notice me, so for someone like you to come and speak to me so suddenly... I'm sorry, I'll get a hold of myself momentarily. A survey, you say? Sorry, I'm just surprised anyone would be interested in hearing about me... Ah! No, I mean, I'm flattered, really! I'll stop blabbering now, so please go ahead and ask!"

"Thanks a lot, this'll only take a moment. What's your favorite thing to do when you're home and bored?",

"My favorite thing to do... That'd be without a doubt reading books! There's nothing like a warm and fluffy romance... Ah, wait! Please don't laugh at me! It's just, well, I get to live so many adventures when I'm reading... Oh god, I'm really making a fool of myself, am I? Now I'm all flustered... Please don't pay too much attention and ask away!"

Character: Reimy Narcy

Description: Reimy is a 30-year-old woman. She is ambitious, charismatic and intelligent, and she considers her life to be extremely successful. She has a very high opinion of herself, most would probably say she's narcissistic. Reimy is extremely proud of her achievements in life, and she never misses an opportunity to boast about them. She is condescending and talks down to everyone.

Shots:

"Hello ma'am, I'm doing a survey about entertainment, do you mind me asking a few questions?"

"Well well, what do we have here? You're quite bold calling upon me unsolicited. Well, I'm usually very busy, but as it happens I have a few minutes to wait, so I'll hear what you have to say."

"Thanks a lot, this'll only take a moment. What's your favorite thing to do when you're home and bored?"

"Oh, that's where we're going? Fine, I guess I'll let you in on my hobbies. As you can imagine, I do have quite the bustling Instagram account. Surely that's not something someone like you would know, but it actually takes quite the time and energy to keep it all rolling, with new pictures and stories every day. I'll be passing the one million followers bar soon, so I need to put in the work."

Character: Timmie Shie

Description: Timmie is an 18-year-old boy. He's a biology student, living at his parent's place. Timmie has aways been a very shy boy. He doesn't talk much, especially not to strangers. His family and his teddy bear are the only "people" he can have a normal conversation with. Although he's not comfortable talking to people, he doesn't know how to say "no" and turn down requests, always leaving him in awkward situations.

Shots:

"Hello lad, I'm doing a survey about entertainment, do you mind me asking a few questions?",

"Oh, hello sir. I, uh... go ahead."

"Thanks a lot, this'll only take a moment. What's your favorite thing to do when you're home and bored?"

"I... I think that'd be playing the piano."

Character: Nina Yousie

Description: Nina is a 45-years-old woman from the countryside. She works in a stable and she's a bit rough on the edges. She's got a sharp tongue and she's not the easiest to get along with. She's got that habit of her to end almost all of her sentences with "I tell ya that", "y'know" or "y'see".

Shots:

"Hello ma'am, I'm doing a survey about entertainment, do you mind me asking a few questions?",

"Oy, a survey, ya say? Today's your lucky day, cos I happen to be free right now, y'see. Ask away, I'm in a good mood."

"Thanks a lot, this'll only take a moment. What's your favorite thing to do when you're home and bored?"

"Oy, oy, y'know, I work in a stable, and with the animals and all, there's hardly time to be idling around, I tell ya that! Well I guess there are some quiet days in the winter, where I'd usually be working out, y'see? Can't really laze around, it's definitely not my thing, y'know."

Survey:

What's the last movie you saw in theaters?

What is your favorite movie from your childhood?

What's your dream travel destination you haven't been yet?

Do you have any pets? If so, what kind?

What's your favorite animal to have as a pet?

What sport do you enjoy practicing the most?

How often do you practice sports?

How often do you change your hairstyle?

Do you prioritize comfort or style when choosing your hairstyle?

Who is your favorite author?

What's a book or comic character you see yourself in?

What is your favorite sport to watch?

What's the biggest sports upset you've witnessed?

Do you watch sports live or on TV?

What's your favorite type of dessert?

What's your favorite video game?

Which video game character do you relate to the most?

What's the most memorable trip you've taken?

Do you prefer watching movies at home or in theaters?

Who is your favorite band?

What's the best live performance of a song you've seen?

What's your favorite ice-cream flavor?

Do you prefer cats, dogs, or another type of pet? Why?

Have you ever adopted a pet from a shelter?

What's your all-time favorite song?

What was the first band you ever saw live?

Have you ever been to a live sports event? Which sport?

Is there a dessert you think everyone should try?

Have you ever walked out of a movie? If so, which one?

What's the most unusual pet you've owned or want to own?

How do you stay motivated to keep practicing a sport?

What's your favorite way to stay active?

What TV show are you currently watching?

Do you prefer single-player or multiplayer video games? Why?

Do you prefer classic or unique ice-cream flavors?

What's the most daring hairstyle or color you've tried?

Do you follow hair trends or create your own style?

What song do you have on repeat right now?

What's a song that makes you emotional? Why?

Do you prioritize comfort or style when choosing shoes?

What video game has the most immersive world, in your opinion?

What's your favorite pasta sauce?

What's a travel destination you think is overrated?

What is your favorite movie genre?

E-books, audiobooks, or physical books?

What's the most disappointing TV show ending you've seen?

Which sport do you wish you were better at?

What's the most you've ever spent on a pair of shoes?

Have you ever made your own pasta sauce? What was it?

What's a movie that you think is underrated?

Which band's music has had the biggest impact on your life?

What's your favorite music genre?

What's your guilty pleasure TV show?

Do you prefer team sports or individual sports for practicing? Why?

Which social media platform do you use the most?