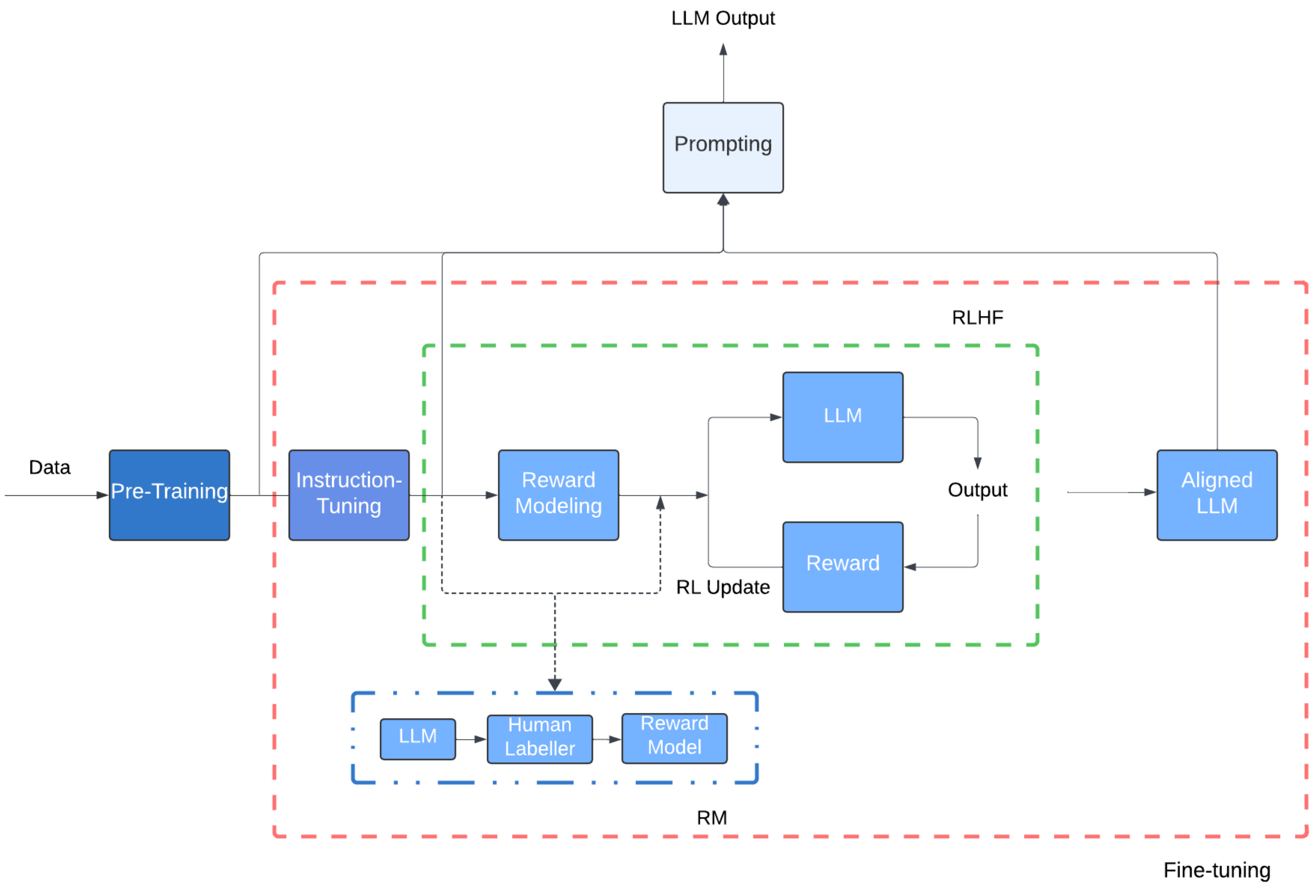

Figure 6: A basic flow diagram depicting various stages of LLMs from pre-training to prompting/utilization. Prompting LLMs to generate responses is possible at different training stages like pre-training, instruction-tuning, or alignment tuning. "RL" stands for reinforcement learning, "RM" represents reward-modeling, and "RLHF" represents reinforcement learning with human feedback.

Masked Language Modeling: In this training objective, tokens or spans (a sequence of tokens) are masked randomly and the model is asked to predict masked tokens given the past and future context. An example is shown in Figure 5.
Unified Language Modeling: Unified language modeling [94] is a combination of causal, non-causal, and masked language training objectives. Here in masked language modeling, the attention is not bidirectional but unidirectional, attending either left-to-right or right-to-left context.

2.11. LLMs Scaling Laws

Scaling laws study the optimal combination of model parameters, dataset size, and computational resources that predict the improvement in the model performance. It has been shown that the loss scales according to the power-law with model size, dataset size, and compute resources [95]. This study suggests larger models are more important than big data for better performance. Another variant of scaling law [96] suggests the model size and the number of training tokens should be scaled equally.

2.12. LLMs Adaptation Stages

This section discusses the fundamentals of LLMs adaptation stages, from pre-training to fine-tuning for downstream tasks and utilization. An example of different training stages and inference in LLMs is shown in Figure 6. In this paper, we refer to alignment-tuning as aligning with human preferences, while occasionally the literature uses the term alignment for different purposes.

2.12.1. Pre-Training

In the very first stage, the model is trained in a self-supervised manner on a large corpus to predict the next tokens given the input. The design choices of LLMs vary from encoder-decoder to decoder-only architectures with different building blocks and loss functions in sections 2.5, 2.4, 2.10.

2.12.2. Fine-Tuning

There are different styles to fine-tune an LLM. This section briefly discusses fine-tuning approaches.
Transfer Learning: The pre-trained LLMs perform well for various tasks [6, 15]. However, to improve the performance for