大规模语言模型

陆俞因

2023.05.30

Language Model (LM)
语言模型

➤标准定义：

给定词典V，语言模型能够计算出任意单词序列$\{w_1, w_2,\ldots, w_n\}$是一句话的概率 $p(w_1, w_2,\ldots, w_n)$ ，其中， $p \geq 0$ 。

***p* = 0.98**

✔今天上课的内容是大规模语言模型

***p* = 0.08**

✘猫机器天气

➤目标：计算概率 $p(w_1, w_2,\ldots, w_n)$

N-gram Language Model

N-gram语言模型

➢根据概率论中的链式法则(chain rule)将 ***p*** 展开

$$p(w_1, w_2, \ldots, w_n) = p(w_1) \prod_{i=2}^{n} \boxed{p(w_i | w_1, \ldots, w_{i-1})}$$

➢文本生成角度，定义语言模型：

给定一个短语（或句子），语言模型能够生成/预测接下来的一个词

N-gram Language Model

N-gram语言模型

- 为简化 $p(w_i | w_2,\ldots, w_{i-1})$ 的计算，引入一阶马尔可夫假设(first-order Markov assumption)

每个词只依赖前一个词，即 $p(w_i | w_2,\ldots, w_{i-1}) \approx p(w_i | w_{i-1})$

$$
\begin{aligned}
& p(w_1, w_2, \ldots, w_n) \\
& = p(w_1) \prod_{i=2}^{n} p(w_i | w_1, \ldots, w_{i-1}) \\
& \approx p(w_1) \prod_{i=2}^{n} p(w_i | w_{i-1})
\end{aligned}
$$

N-gram Language Model

N-gram语言模型

➢也可以引入二阶马尔可夫假设(second-order Markov assumption)

每个词只依赖前两个词，即 $p(w_i \mid w_2,\ldots, w_{i-1}) \approx p(w_i \mid w_{i-2}, w_{i-1})$

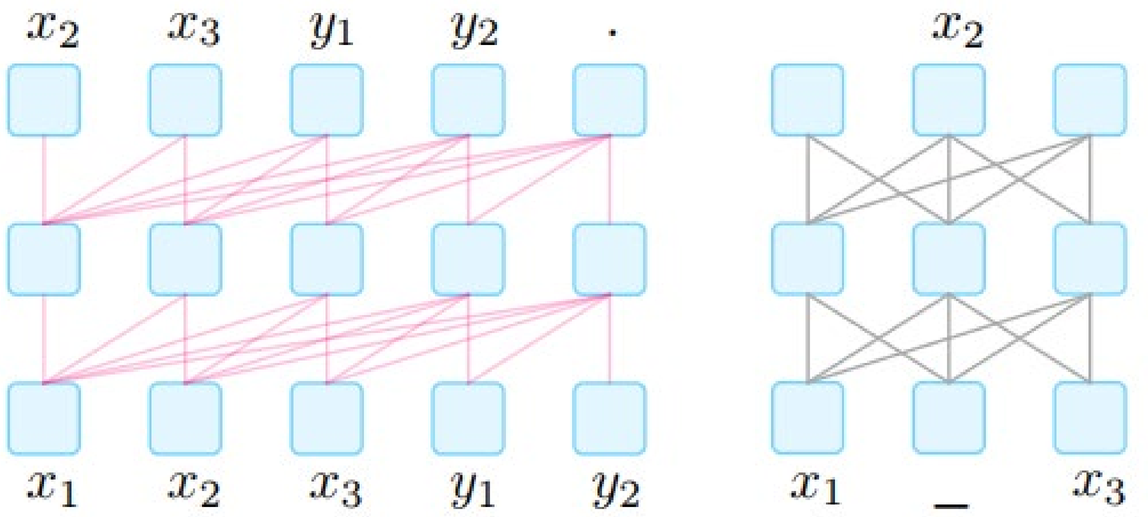
$$p(w_1, w_2, \ldots, w_n)$$
$$= p(w_1) \prod_{i=2}^{n} p(w_i | w_1, \ldots, w_{i-1})$$
$$\approx p(w_1) p(w_2 | w_1) \prod_{i=3}^{n} \boxed{p(w_i | w_{i-2}, w_{i-1})}$$

统计方法：$\boxed{p(w_i | w_{i-2}, w_{i-1}) = \frac{count(w_{i-2}, w_{i-1}, w_i)}{count(w_{i-2}, w_{i-1})}}$

Neural Network Language Model

神经网络语言模型

➢通过神经网络预测下一个单词

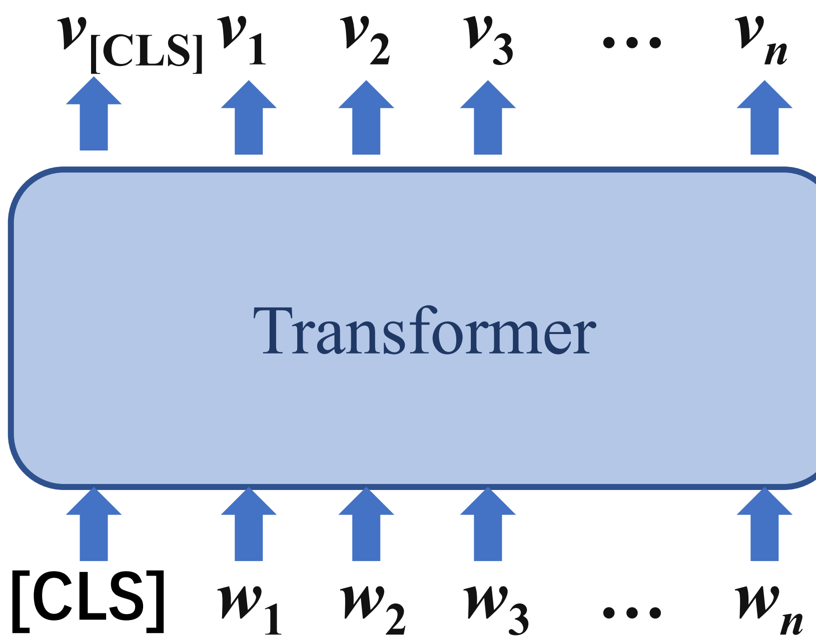


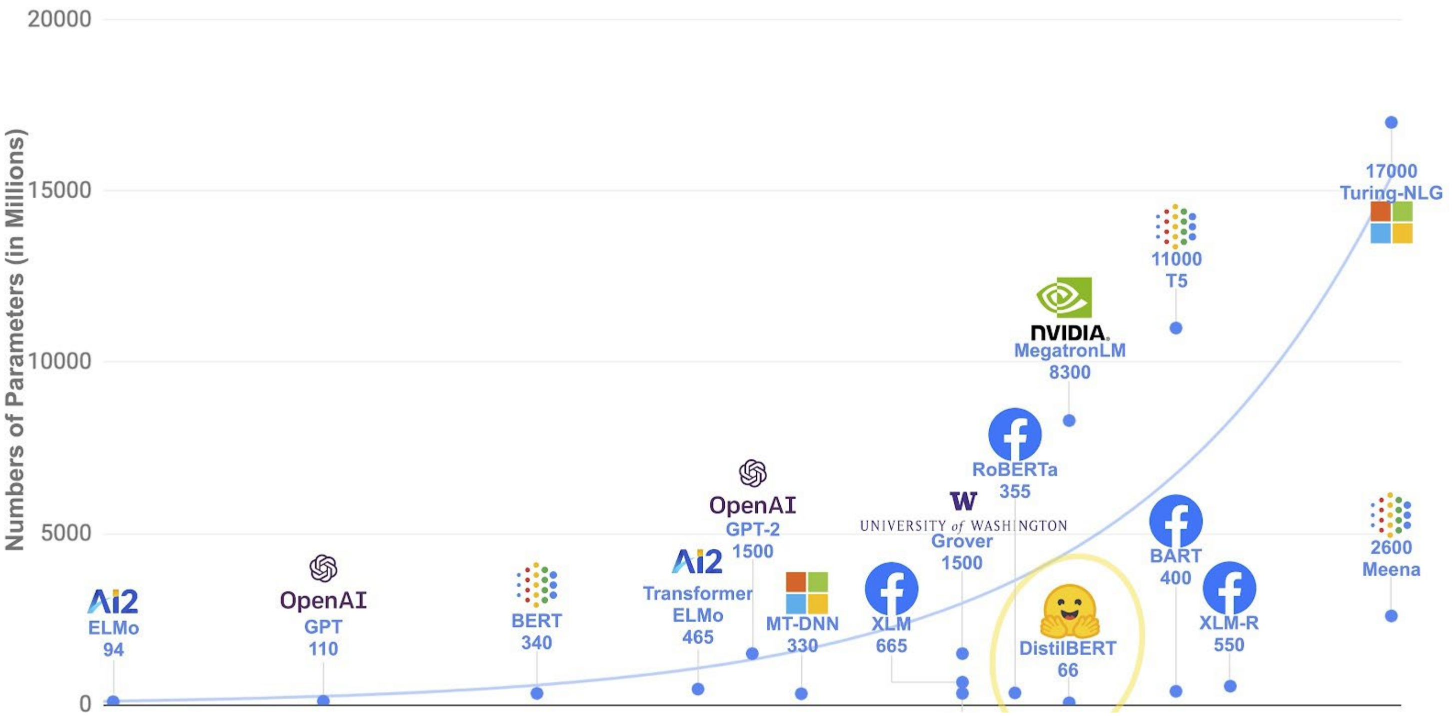
(a) Left-to-right LM. (b) Masked LM.

代表模型： GPT-3 BERT、ERNIE

Large Language Model (LLM)
大规模语言模型

➢通常采用Transformer作为网络架构



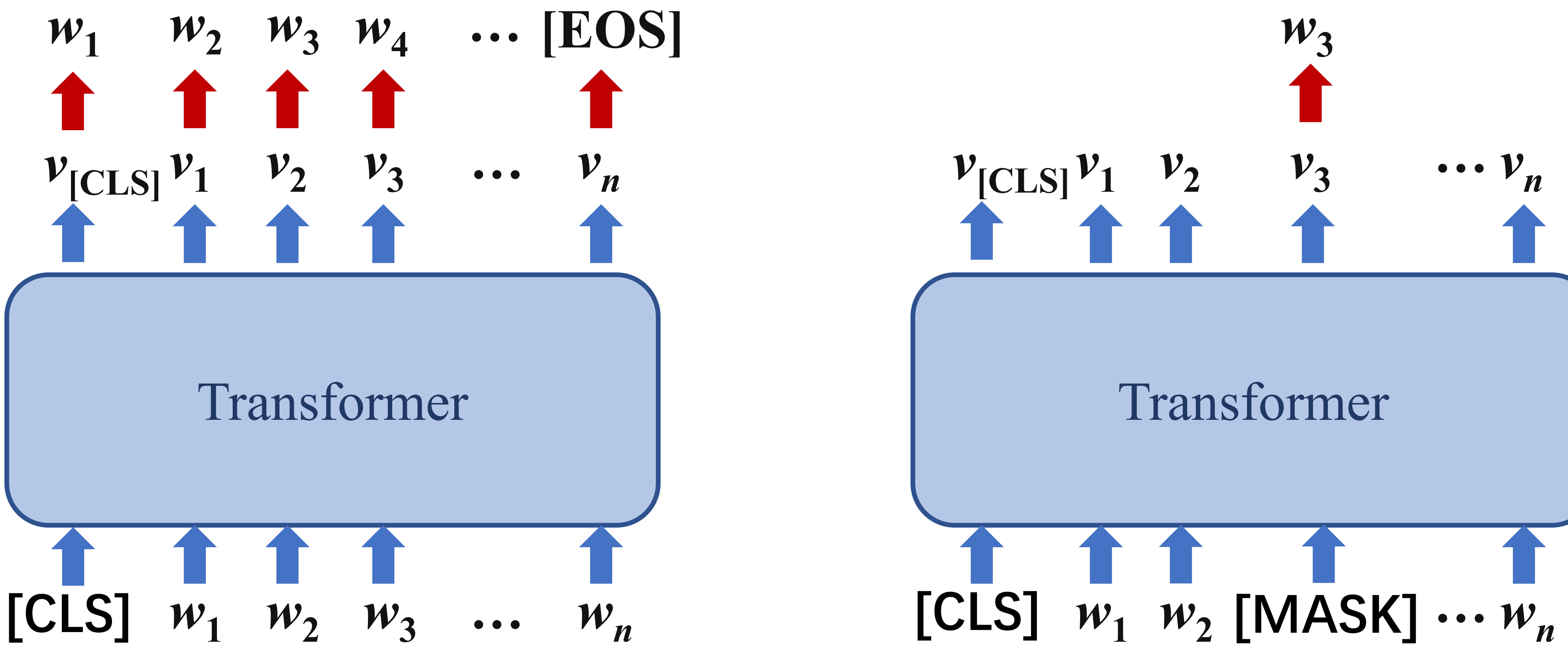


参数量

Pretrain

预训练

- ➤在**无标记语料**上通过**自监督任务**进行预训练



Left-to-Right LM　　　Masked LM

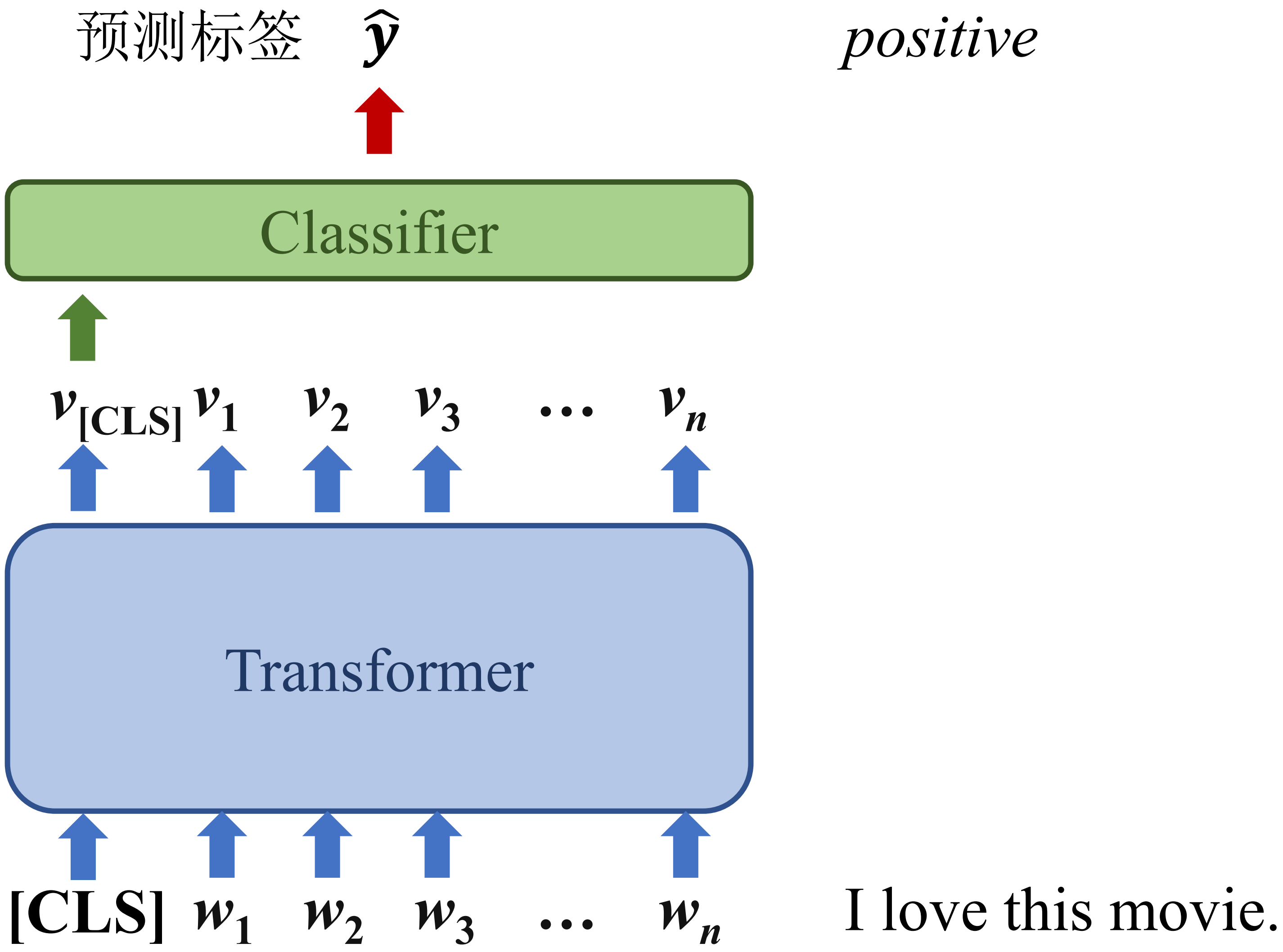
Fine-tune

微调

➤在**有标记语料**上进行微调

以文本分类为例：

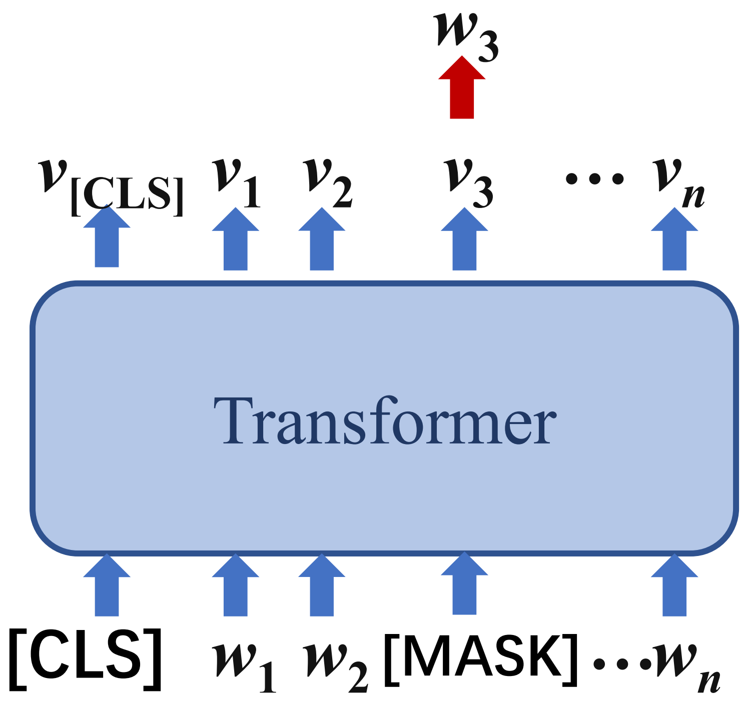
通常采用$\boldsymbol{v}_{[\mathrm{CLS}]}$ 作为文档嵌入

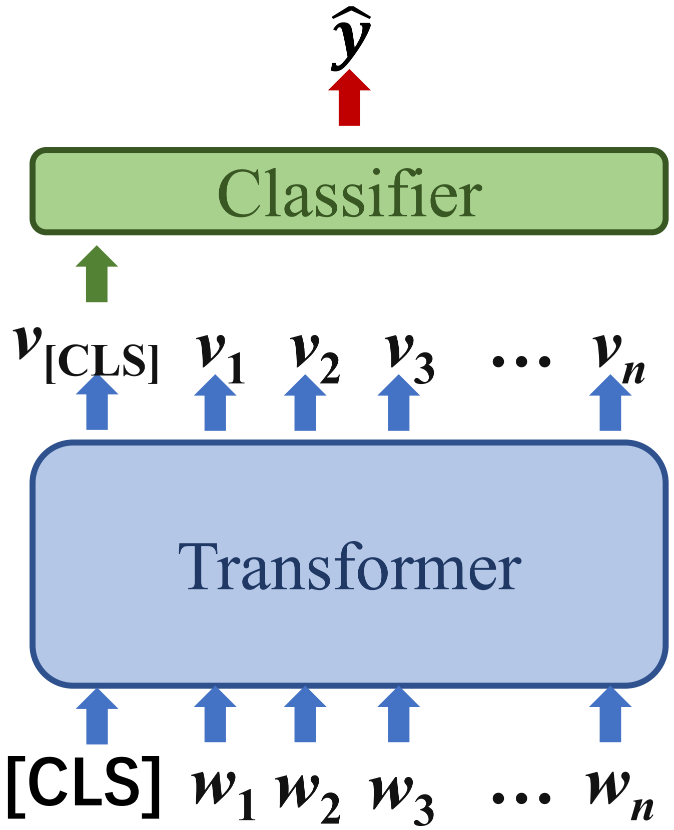


Prompt Learning
提示学习

- 预训练-微调范式的存在问题：

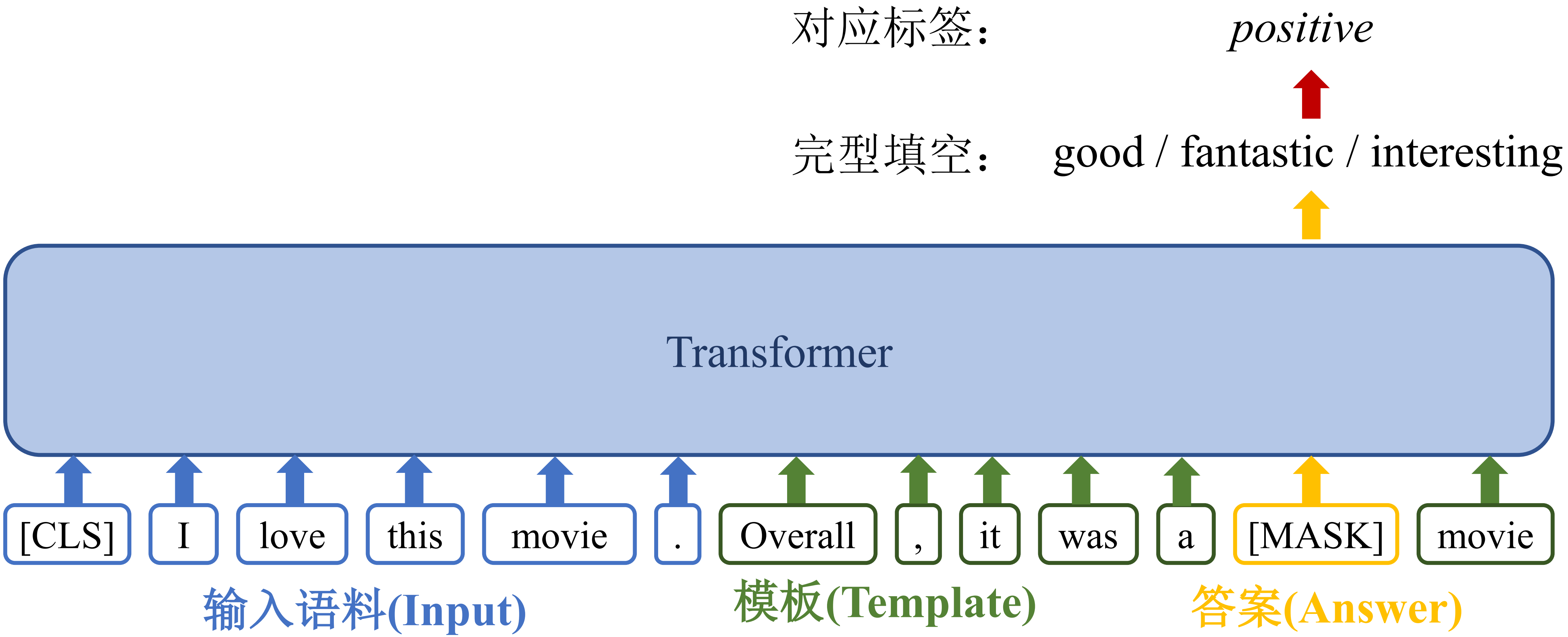
预训练阶段的自监督任务与微调阶段的有监督任务存在较大的差距





Prompt Learning
提示学习

- 目标：拉近预训练与目标任务的距离



Prompt Learning

提示学习

➢形式化：

[CLS] I love this movie . Overall , it was a [MASK] movie

输入语料(Input) 模板(Template) 答案(Answer)

[X] [Z]

提示(Prompt): [X] [Template] [Z]

Prompt Learning
提示学习

➤形式化：

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills `[Z]`

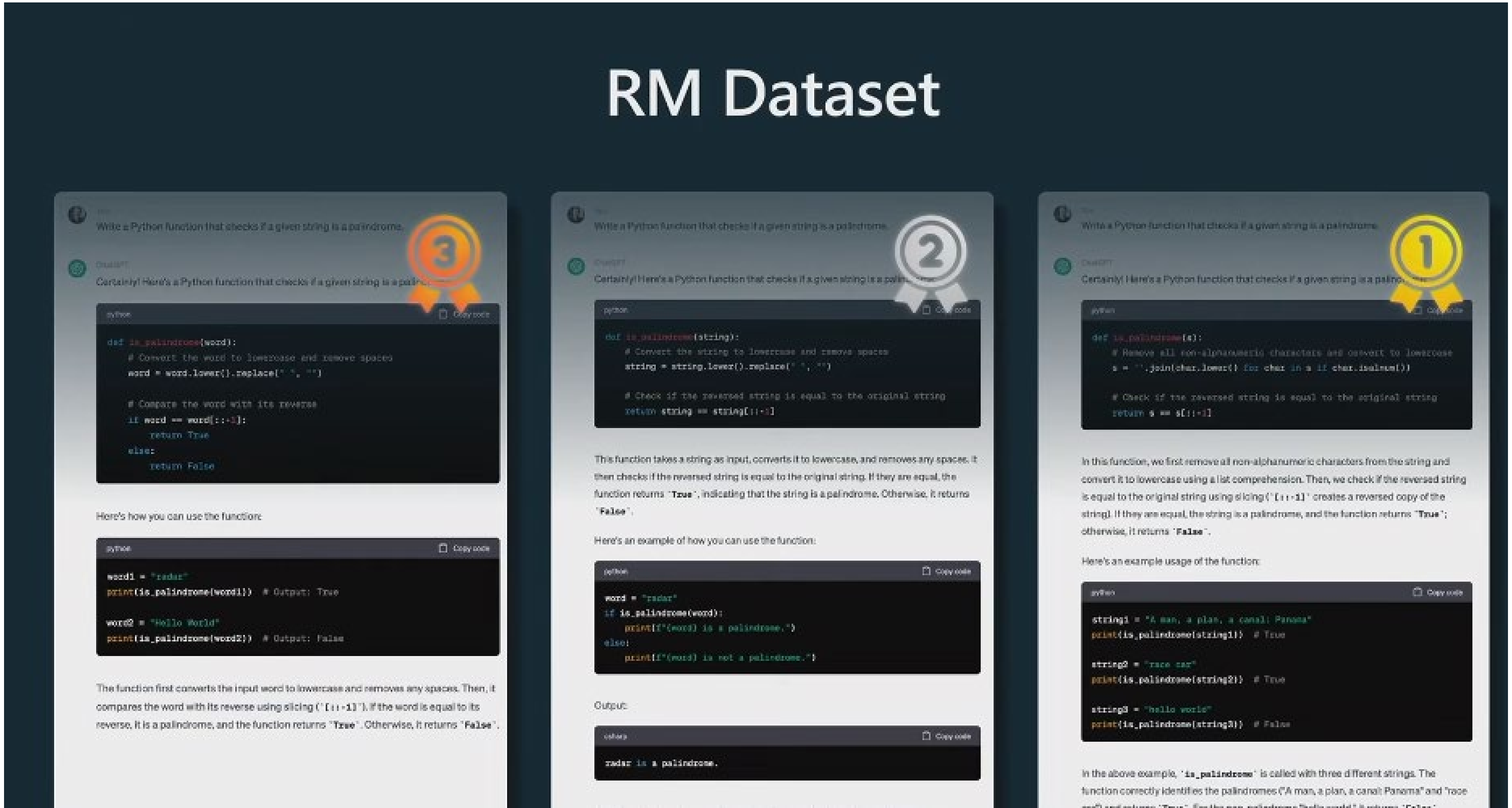
GPT Assistant training pipeline

Stage	Pretraining	Supervised Finetuning	Reward Modeling	Reinforcement Learning
Dataset	**Raw internet** text trillions of words low-quality, large quantity	**Demonstrations** Ideal Assistant responses, ~10-100K (prompt, response) written by contractors low quantity, high quality	**Comparisons** 100K –1M comparisons written by contractors low quantity, high quality	**Prompts** ~10K-100K prompts written by contractors low quantity, high quality
Algorithm	**Language modeling** predict the next token	**Language modeling** predict the next token	**Binary classification** predict rewards consistent w preferences	**Reinforcement Learning** generate tokens that maximize the reward
		init from	init from	init from SFT use RM
Model	**Base model**	**SFT model**	**RM model**	**RL model**
Notes	1000s of GPUs months of training ex: GPT, LLaMA, PaLM can deploy this model	1-100 GPUs days of training ex: Vicuna-13B can deploy this model	1-100 GPUs days of training	1-100 GPUs days of training ex: ChatGPT, Claude can deploy this model

https://www.bilibili.com/video/BV1ts4y1T7UH/?vd_source=5ef3b9db5a34764c84c73611f33aeafc

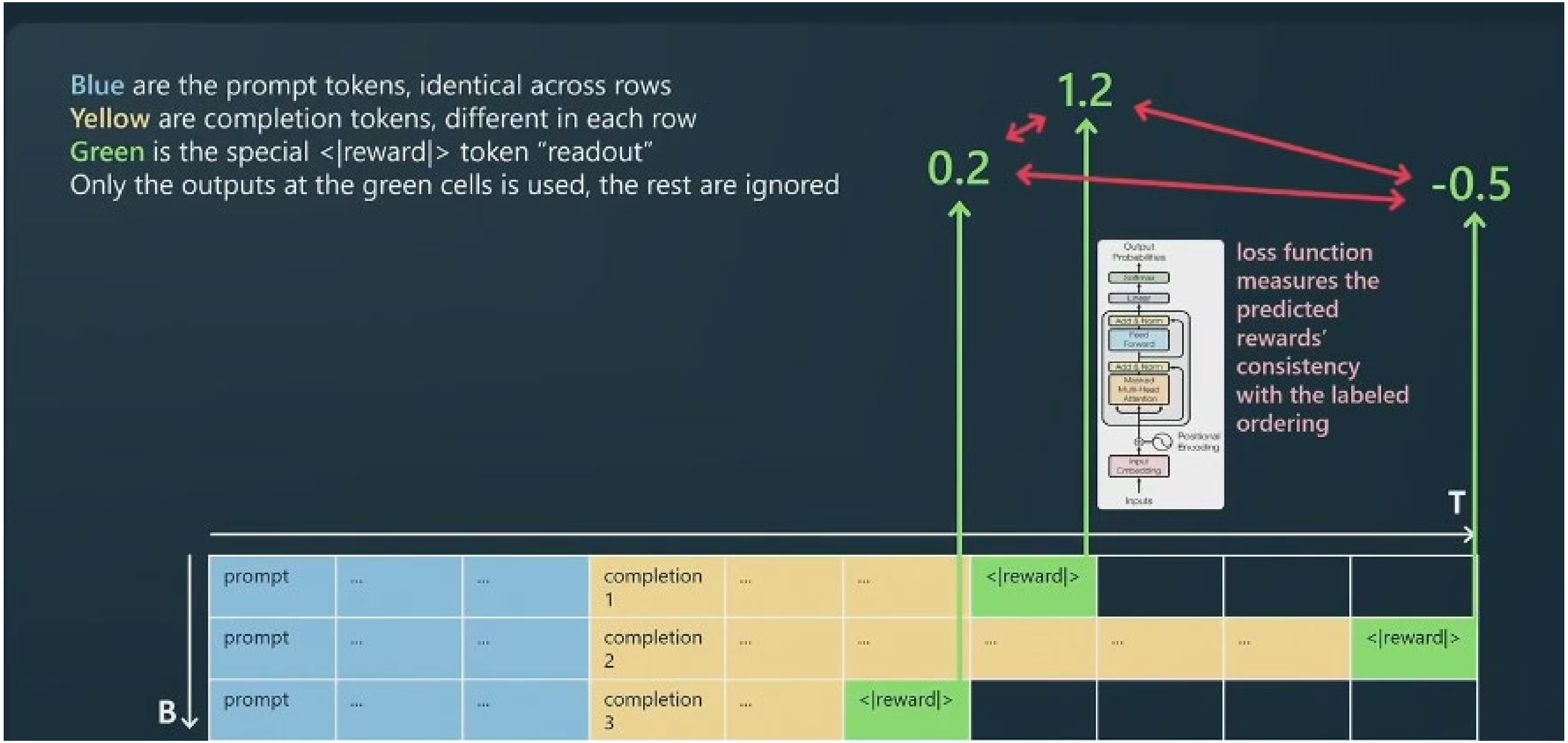
Reinforcement Learning from Human Feedback (RLHF)
根据人类反馈进行强化学习

➢奖励建模 (Reward Modeling)



Reinforcement Learning from Human Feedback (RLHF)
根据人类反馈进行强化学习

➢奖励建模 (Reward Modeling)



Reinforcement Learning from Human Feedback (RLHF)
根据人类反馈进行强化学习

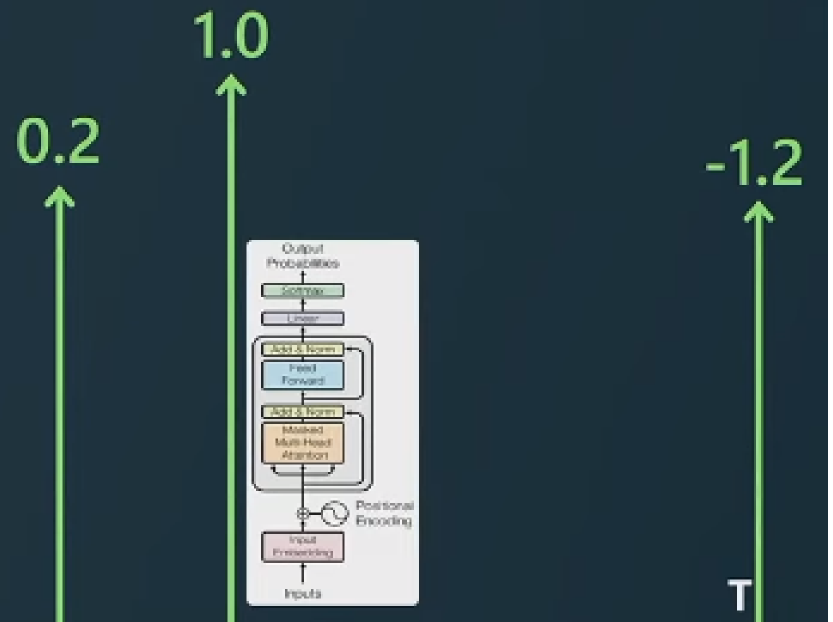
➢强化学习(Reinforcement Learning)

Blue are the prompt tokens, identical across rows
Yellow are completion tokens by the model (initialized with **SFT** model)
Green is the special <|reward|> token “readout”, **RM** now predicts these
Only the **yellow** cells are trained on, the rest are ignored.

The sampled tokens become labels, but the training objective is weighted by the “advantage” (normalized rewards)

In this example:

- Row #1 tokens were great. These get their probabilities boosted.
- Row #2 tokens were bad. These get their probabilities decreased.
- Row #3 tokens were ~ok. These get their probabilities slightly boosted.



<table>
<tr><td>prompt</td><td>...</td><td>...</td><td>completion 1</td><td>...</td><td>...</td><td><|reward|></td><td></td><td></td><td></td></tr>
<tr><td>prompt</td><td>...</td><td>...</td><td>completion 2</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td><|reward|></td></tr>
<tr><td>prompt</td><td>...</td><td>...</td><td>completion 3</td><td>...</td><td><|reward|></td><td></td><td></td><td></td><td></td></tr>
</table>