

AGI Prototype Road Map

可以分拆成以下较细问题：

- ① **选择机制**：Transformer 输出 N 个 tokens，每个 token 有它的 Q-value，我们选择最高 Q-值 的 tokens 放进 工作记忆。这似乎没有什么困难。
- ①B **encode Q-值**：token 通常是一个 512-维向量，我认为应该增加维数（因为命题空间明显应该比 词汇空间更大）。透过矩阵乘法，一个 token 可以变成词汇之上的概率分布。
- ①C **工作记忆 cache**：因为 Transformer 每次输出一大堆 candidates，工作记忆 遗忘的速度不够快，似乎需要 caching。
- ② **「找关系」函数**：设计一个 M -in N -out c 函数 $c = \text{“chooser”}$ ， M 和 N 是 tokens 个数， $M \gg N$ 。它的作用是从 M 个 tokens 里找出最有可能是某个逻辑 rule 的前提的 N 个 tokens，作为 Transformer 的输入。注意：每个 rule 的前提都是不同的，所以 c 的输出 可以有很多可能。这个问题其实在 LLM 里已经有研究，因为我们不需要 Transformer 有太长的遥距相关。
- ③ **遗忘机制**：在工作记忆里，如何选出那些要被遗忘的 tokens？
- ④ **输入/输出 编码方法**：工作记忆里面，所有 tokens = 命题 都是没有次序的。那么需要一个位置编码的方法，这个跟以前 Transformer 的要求是一样的。
- ⑤ **RL 的训练方法**：在 强化学习下，输入新的句子后，我们的 RL 系统会在 n 步之后输出 词语，组成句子。我们要用 GPT 给予 RL 系统奖励。