# 《BERT 与逻辑的结合》

YKY

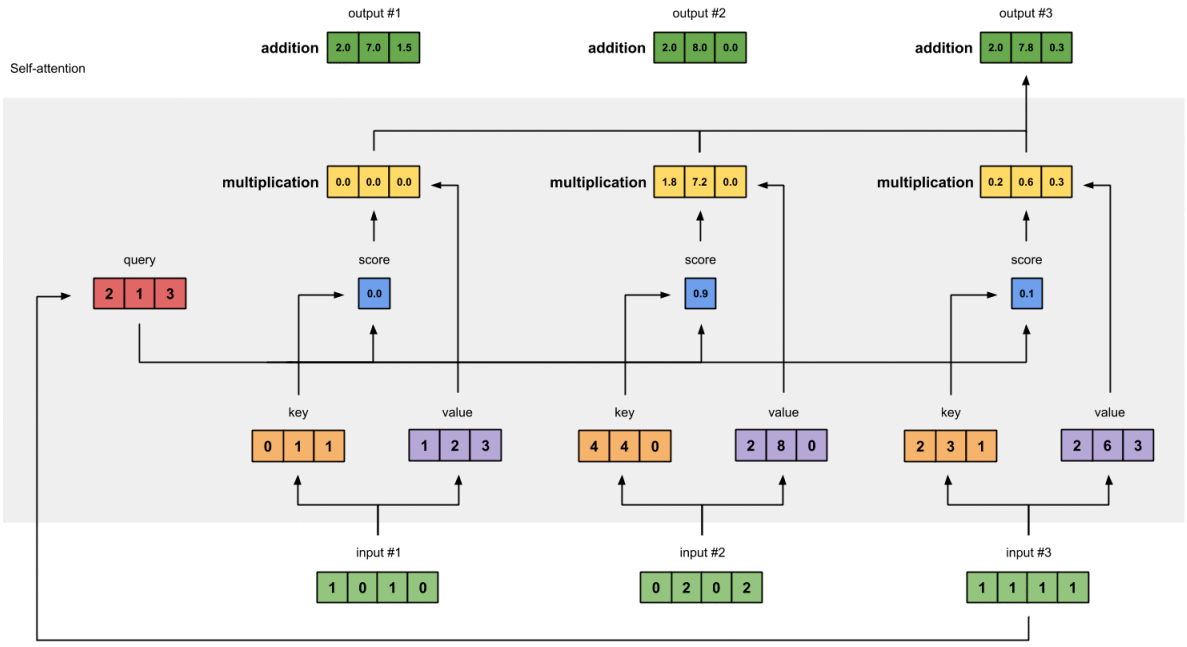
September 27, 2021

- 我比较熟悉 经典逻辑 AI，写过 逻辑引擎
- 但我没有 BERT/GPT 的实战经验
- 今天我们考虑一下 结合 BERT/GPT 和 逻辑引擎 的可能，有什么优势？

## 0. 我们的策略

- 将 BERT/GPT 解释为一种 逻辑 / 符号演算的系统
- 将 逻辑结构 impose 到新的 BERT/GPT 模型
  （它不再是语言模型，而是逻辑模型）
- 利用我们对逻辑 AI 的理解，
  改良这新的模型，
  从逻辑角度理解参数的意义
- 如果不这样做，BERT/GPT 仍然是 “black box”，
  那就很难想出改良的思路

# 1. Transformer 的 equi-variance




(1)

# 2. Logic AI 的基本架构

系统的 **状态 (state)** = 例如：

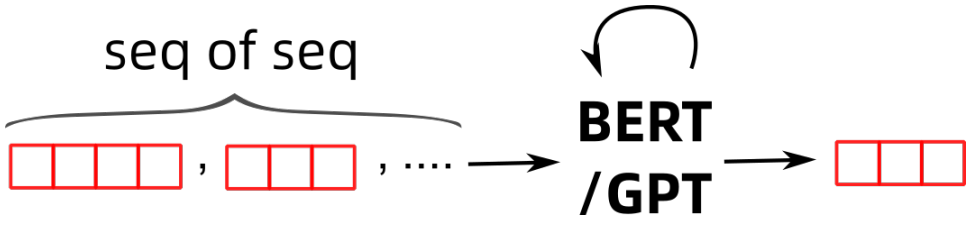
我很肚饿 ∧
冰箱没有食物 ∧
现在是午夜 **3** 点 ∧
商店已经打烊 ∧
.... ∧ ....

换句话说，状态 是 一堆 逻辑命题 的 **集合**

- 一直以来，人们觉得 大脑的 **KR**（**knowledge representation,** 知识表述）跟符号逻辑 肯定是大相迳庭的

# 3. Seq-seq-2-seq

- 逻辑 与 自然语言 之间大约有这样的对应：
  句子 ≈ 命题，词语 ≈ 概念，
  命题 = 多个概念的 **concatenation**
- 从 强化学习的角度看：
  状态 **(state)** = 命题集合，
  **transition function**：命题集合 → 命题集合
- 命题集 = **sequence of** 命题，
  命题 = **sequence of concepts**，
  所以 状态 = 命题集 = **sequence of sequences (seq-seq)**
- **transition fn: seq-seq → seq-seq**

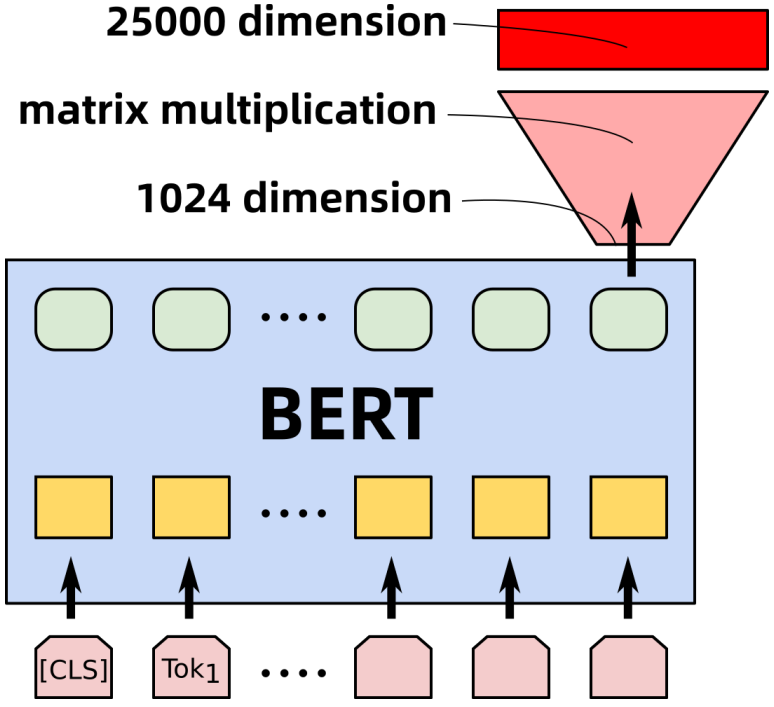



(2)

- 「状态」的另一个名称是 **working memory**
  （借用 **认知科学** 术语）
- 每次状态更新时，我们可以只增添一个命题，
  「**遗忘**」另一个命题
- 因此 **transition fn** 只需是 **seq-seq → seq**
- 而我想说的是：**BERT/GPT** 可能就是一种 **seq-seq → seq**

# 4. 强化学习的考虑

- 从 强化学习 的角度看，
  每个 iteration 要输出一个 **命题** = 几个**词语**
- 这 输出 对应于 强化学习的 **actions**
- 换句话说，每个 **action** = 一个命题 = 几个词语
- 所以，我们需要输出 在 **actions** 之上的 **概率分布**
  （而不仅仅是一个 **action**）
- 数学上 这是 $\{\text{所有可能命题}\} \to \mathbb{R}$ 的空间 $= \mathbb{R}^{|X|}$
- 这个空间异常大，我初时觉得 没有希望在计算机上表达
- 但 **Dr** 肖达 解释了一个很有效率的方法，
  用 矩阵乘法 将输出 由 **1024** 维 **扩张**到 **25000** 维：




(3)

- 但这个做法，其实输出的 只有 **1024** 个 **独立**的份量

例如，「天气很热，我在家中整天_____」

- 流汗
- 吃冰淇淋
- 喝冰水
- 不穿衣服
- 开冷气....

「女朋友说分手，我觉得_____」

- 很伤心
- 如释重负
- 很气愤
- 很妒忌....

「电脑的键盘没反应，可能是因为_____」

- 未插线
- 电线断了
- 档机了
- 视窗未 active ....

考虑这些例子，我暂时不清楚 1024 维 够不够用。
以 1024-dim 表示所有 **概念** 是足够的（cf. Word2Vec）
但未知它能不能够 表示所有常见的 **multi-modal** 概率分布。

# 5. BERT/GPT 是符号演算系统

**Few-shot generalization.**

# 6. Relation algebra

Relation algebra 似乎是一种更 接近 自然语言 的 逻辑形式：

$$\begin{matrix} F & \circ & F & = & G \\ \text{爸爸} & \text{的} & \text{爸爸} & \text{是} & \text{爷爷} \end{matrix} \tag{4}$$

# 7. 自动 产生 / 运行 代码

「计算我生命中的秒数」