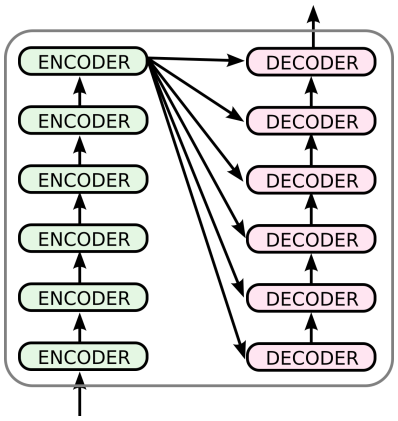
White paper

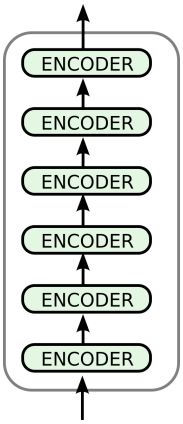
March 31, 2021

This is the original Transformer architecture:



(1)

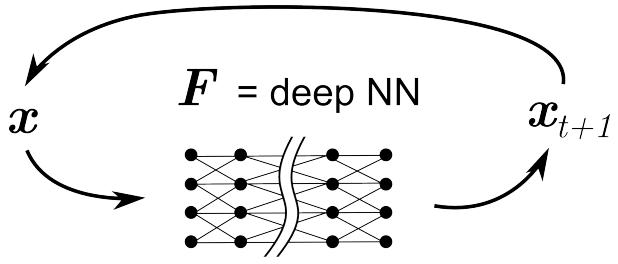
This is the "Encoder" part:



(2)

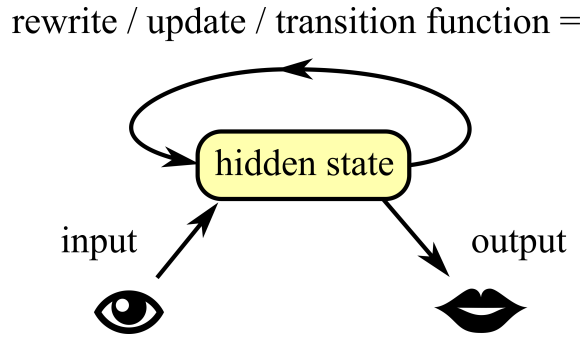
(A friend told me that BERT only uses the Encoder part of Transformer.)

所不同的是，在 强化学习里，状態 is iteratively updated:



(3)

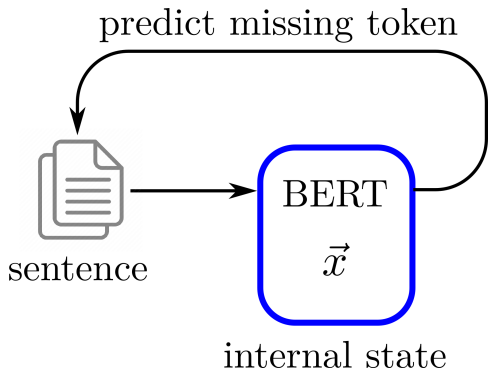
其中，input/output are reserved parts inside the state vector:



(4)

现在问题是我们想用 reinforcement learning 的架构，加到 BERT 上。

首先看看 BERT 是怎样训练的：



(5)

其实我也不太清楚怎样将 BERT 与 RL 结合，其中一篇論文是 **KG-A2C** (Knowledge-Graph Advantage Actor-Critic):

https://openreview.net/forum?id=B1x6w0EtwH

According to the paper, the update is done via an Advantage function A:

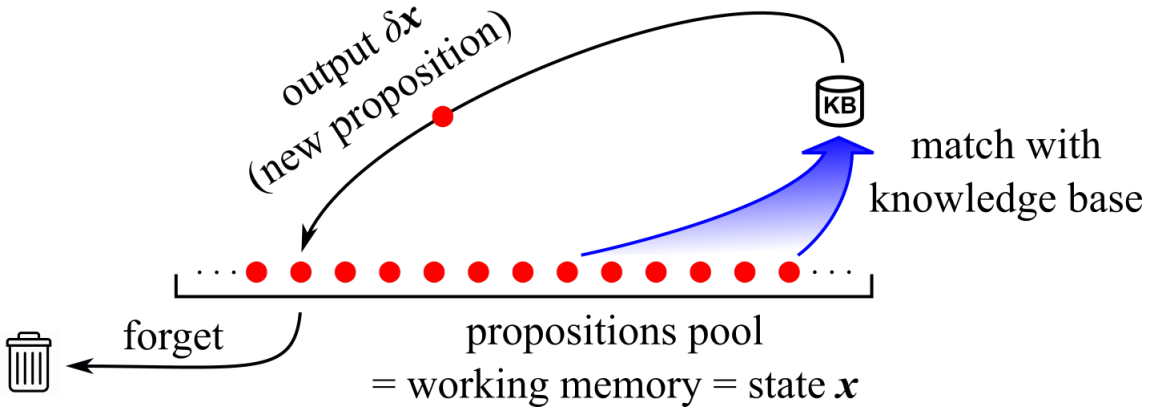
$$A(\boldsymbol{x}_t, a_t) = Q(\boldsymbol{x}_t, a_t) - V(\boldsymbol{x}_t) \tag{6}$$

$$Q(\boldsymbol{x}_t, a_t) = \mathbb{E}[r_t + \gamma V(\boldsymbol{x}_{t+1})] \tag{7}$$

where R is the reward. V is the value function and Q is the value function restricted to action a, as is standard in Q-learning. This part seems pretty standard for A2C.

如果不用 BERT 而用我的 symmetric neural network 方法可能更易：

用 symmetric neural network 的话，系统的 状态 有 **逻辑命题** (red ●) 的结构：



(8)

在每个时间点 t，状态 以如下方式 **update**:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t \oplus \delta \boldsymbol{x} \ominus \text{forget}(\boldsymbol{x}_t) \tag{9}$$

暂时不会 implement forgetting mechanism, 我们只是用一个足够大的 state 装下整个 NL 句子的字（每个 NL 字 用一个逻辑命题），**遗忘** 时间最早的那些命题。