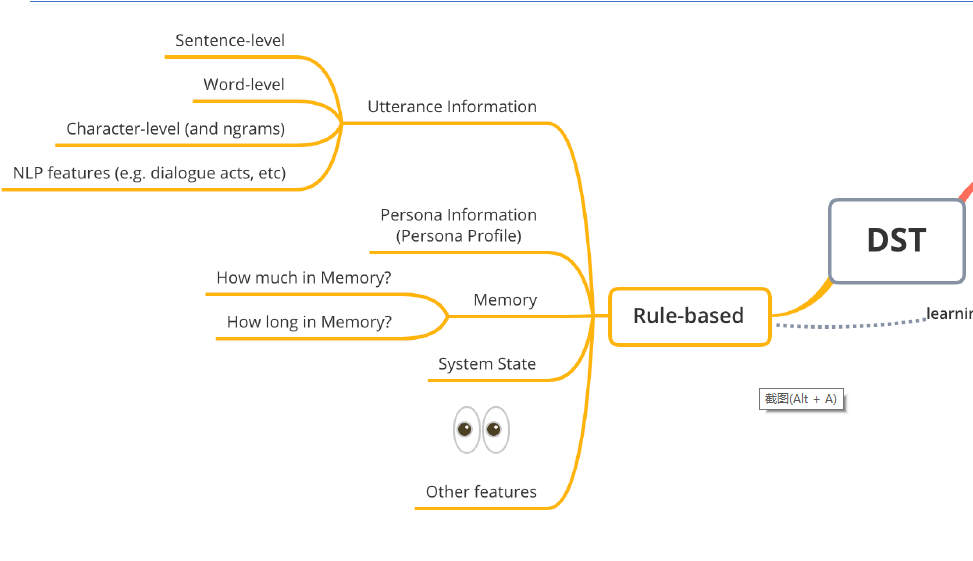
基于Model的DST和Rule的DP以及RASA的Policy理解

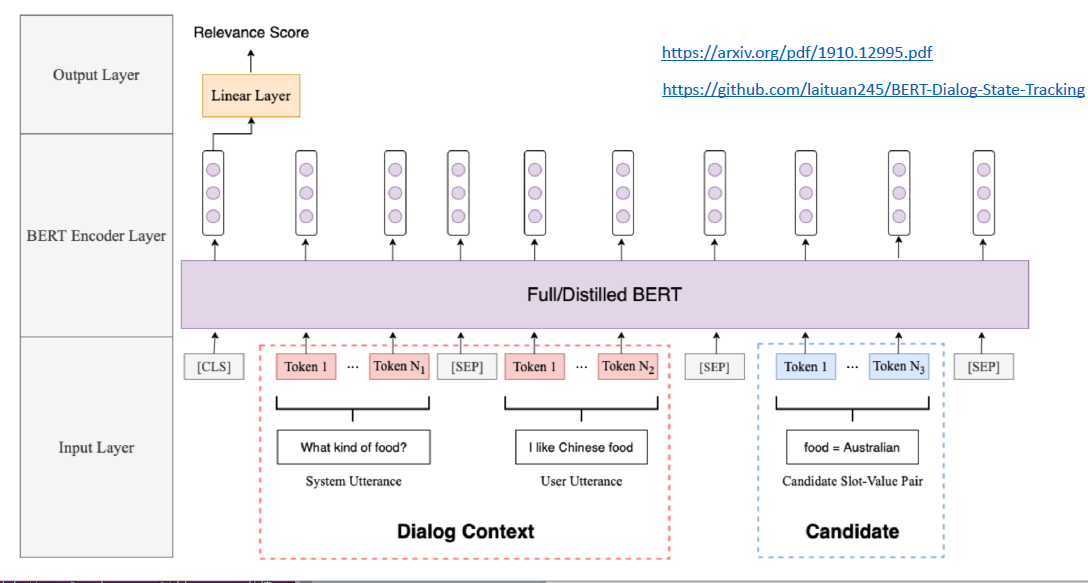
DST的目的在于从上文的多轮历史对话中找到符合的intent和entity。

上节课我们学会了基于Rule的DST



我们可以收集Utterance信息，用户个人信息，内存信息，系统状态等等其他特征放入基于Rule的DST中。

基于Model的DST



模型

它使用的是基于BERT模型或者其衍生模型来做DST。整体结构如上图，变动在于其输入部分，它包括两部分一个是内容对话和candidate。它将对话内容按照bert的形式作为输入，candidate是以键值对的形式，经过bert，CLS中代表是全局信息，然后接一个全连接层，得到一个分数，然后与一个阈值进行比较，超过则将candidate存储起来。也就是说这个模型的目的也是为了找到合适的intent和entity，这与DST的目的是一致的。所以说对话中有多少intent和entity，就有多少次训练，所以整个来说还是比较费资源的。

Further DST

上述训练没有考虑用户的特征等等其他因素。所以我们也可以将用户的信息做一个用户画像，

使用一种类似星空算法的方式来进行比较。我们还可以将响应的长度，句子是否重复等等其他信息也可以作为一个特征作为输入。

Rule的DP

DP的目的在于找到action,action是作为对对话的一种反应，它同时也包括了intent和是sloty.

它的输入是我们DST中找到的intent和entity，目的是为了找到action。

我们还有需要考虑其他的情形

Change Slot

用户在对话中更换了Slot,我们需要将其捕获然后放入词槽中

Repeat Intent

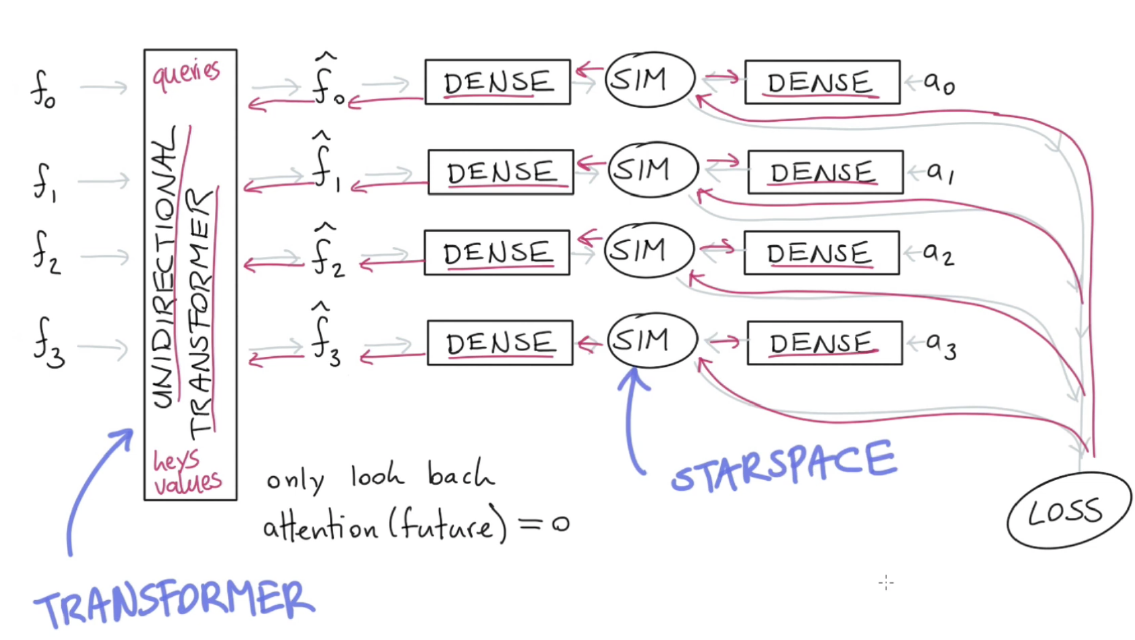
用户在对话中反复提及的Intent，我们也需要理解

SYStem State

在一些任务型机器人，当我们需要更换任务，此时系统仍在处理原来的事务，我们也需要智能化的提醒用户，同时还不能忘记用户的需求。

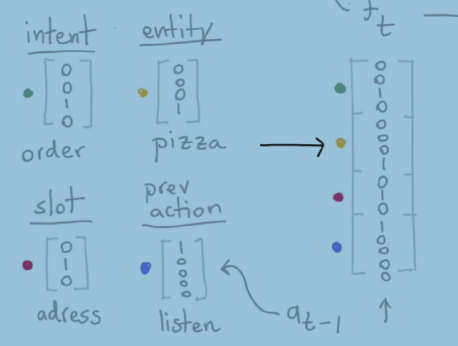
RASA Policy

在RASA中它是怎么处理的呢



它是将对话作为一个个特征作为Transformer的输入，然后得到一个前面上下文的信息，然后经过一个全连接层，与此同时action也会经过一个全连接层，然后他们进行相似度比较，这是一个星空算法的思想。得到的LOSS,然后反向传播训练。

用户特征是来自这句的捕获到的intent entity slot以及之前的action特征进行合并得到的。



用户特征的数目这是可以由用户系统设置的。值得注意的是用户的对话是一个无法预知下一句的过程，这是一个单向过程，所以在上述模型中transformer是一个单向的。这个我的理解是在计算attention的时候使用一种三角矩阵的思想。