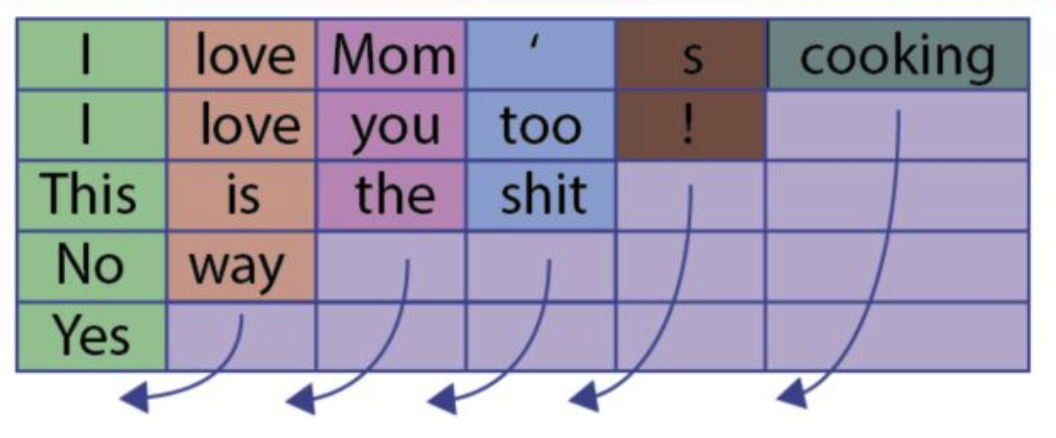
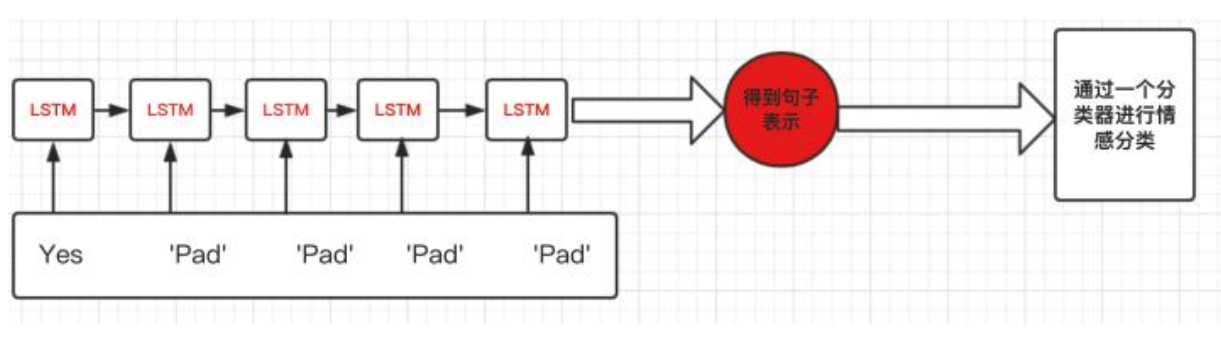
Pytorch中的RNN之pack\_padded\_sequence()和pad\_packed\_sequence()

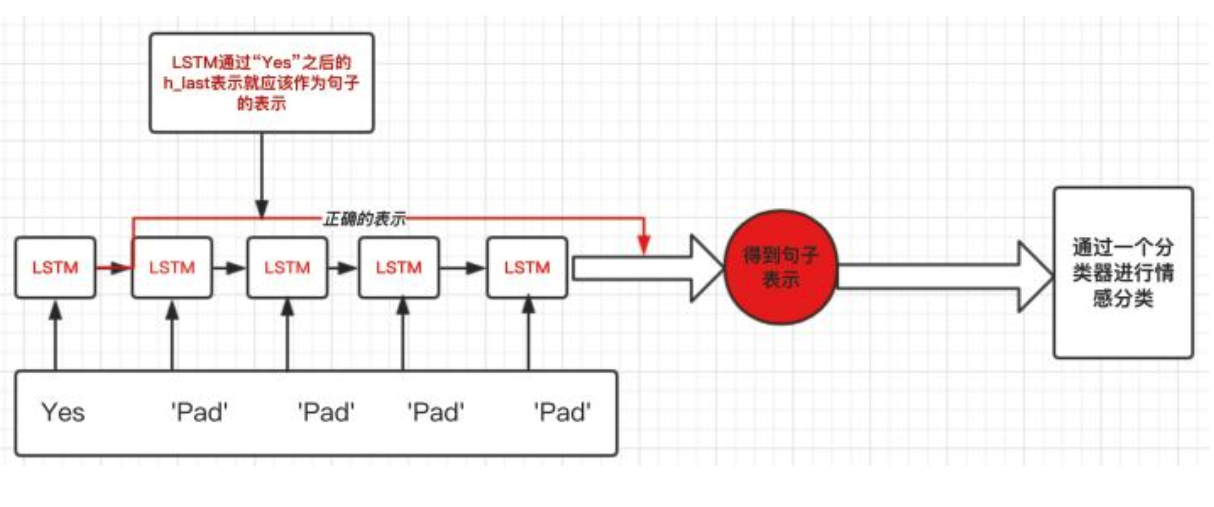
为什么会有pad和pack操作，先看一个例子，这个batch中有5个sample。



如果不用pack和pad操作会有一个问题，比如上图中的句子“Yes”只有一个单词，但是padding了多余的pad符号，这样会导致LSTM对它的表示通过了非常多的误用符号，这样得到的句子表示就会有误差，更直观的如下图：



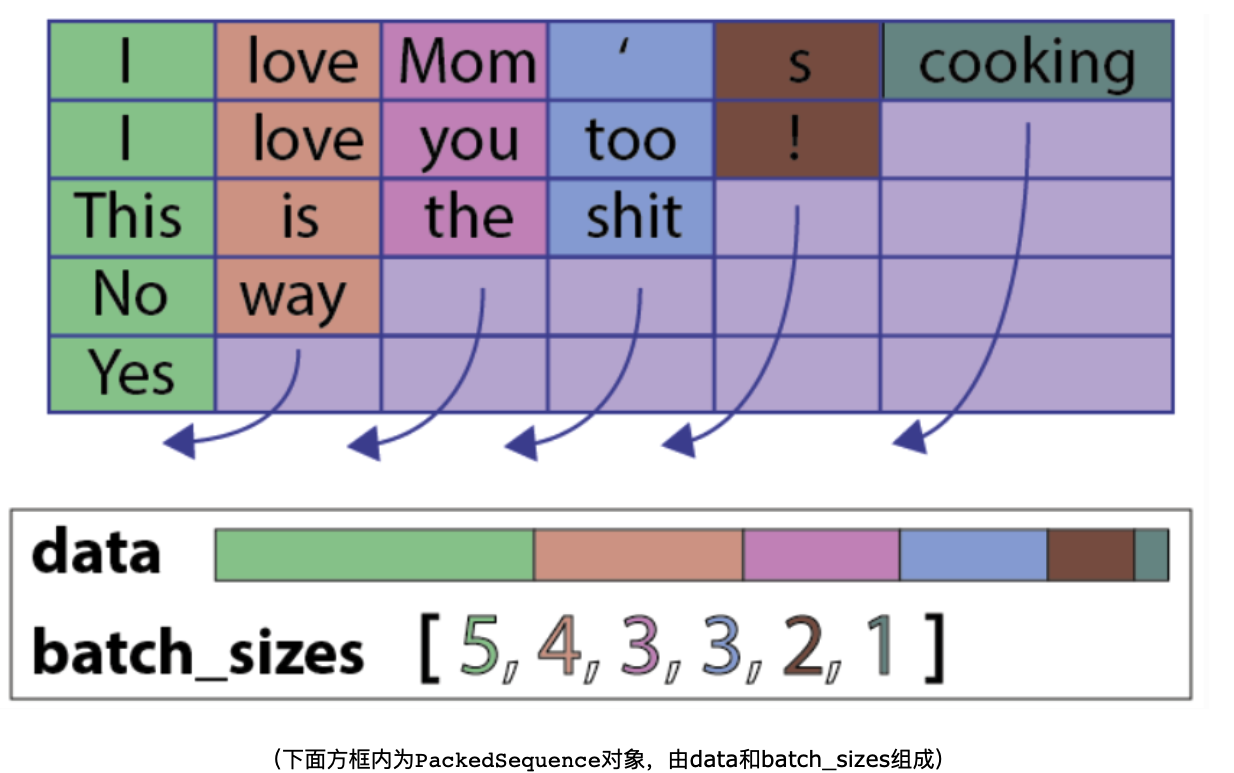
那么我们应该怎么做呢？在上面的这个例子，我们想要得到的表示仅仅是LSTM过完单词“Yes”之后的表示，而不是通过了多个无用的“Pad”得到的表示，如下图：



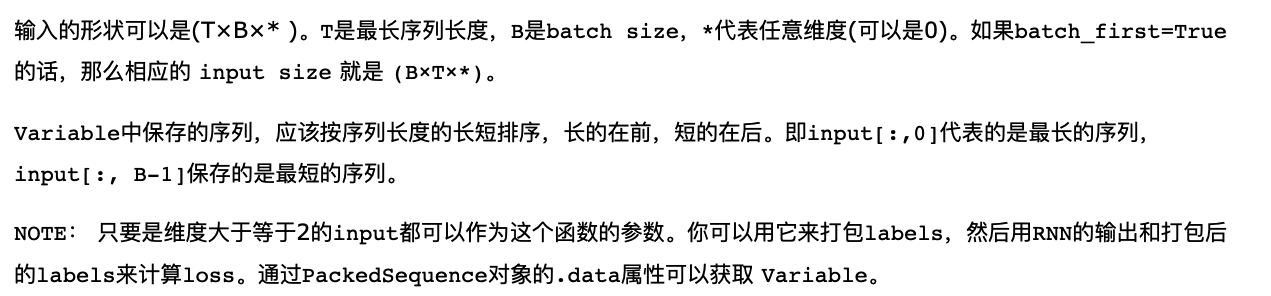
**Torch.nn.utils.rnn.pack\_padded\_sequence()**

这里的pack理解为压紧，将一个填充的变长序列压紧。（填充的时候会有冗余，所以压紧一下）。

其中pack的过程为：注意pack的形式，不是按行压，而是按列压。



Pack之后，原来填充的PAD(一般初始化为0)占位符就被删掉了。



参数说明：

* Input(Variable)—变长序列，被填充后的每个batch
* Lengths(list[int])—Variable中每个序列的长度
* Batch\_first(bool, optional)—如果是True，Input的形状应该是B\*T\*size

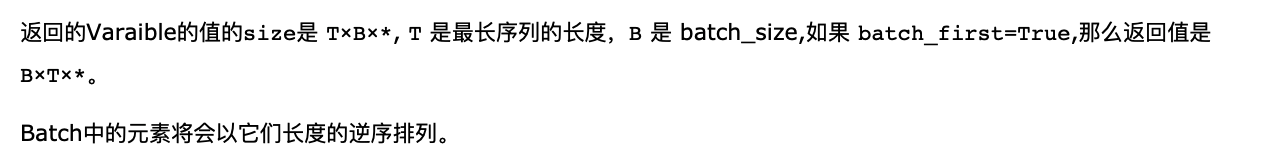
返回值：

* 一个PackedSequence对象

**Torch.nn.utils.rnn.pad\_packed\_sequence()**

实现填充packed\_sequence。

上面提到的函数的功能是将一个填充后的变长序列压紧。这个操作和pack\_padded\_sequence()是相反的。把压紧的序列再填充回来，填充时会初始化为0.



参数说明：

* Sequence(PackedSequence)—将要被填充的batch
* Batch\_first(bool, optional)—如果为true，返回的数据的格式为B\*T\*

返回值：

* 一个tuple,返回被填充后的序列，和batch中序列的长度列表

一个例子：

