**语言分析和机器翻译实验报告**

**班 级：计硕2009**

**学 号： 2071845**

**学生姓名： 赵 玉 迪**

### 实验目的

1. 复习课上内容，了解机器翻译相关技术。
2. 查阅资料了解NLP前沿技术，为日后学习做铺垫。

### 实验内容

#### 机器翻译相关技术

机器翻译的发展是一个漫长的过程，先经历了遵循基于规则的词法、句法分析；后随着计算机处理能力的不断提高和大规模语料库的逐步构建，出现了统计型机器翻译；而今天神经机器翻译已成为主流。

随着深度学习框架的不断发展，机器翻译也在不断地迎来变革。2016年谷歌发表的gnmt框架，其主要结构分为三部分：编码网络（encoder），负责将每个输入字符转换成相应的词向量；解码网络（decoder），负责接收词向量并生成翻译文本；注意力模块（attention），编码器和解码器通过注意模块连接，该注意模块允许解码器在解码过程中聚焦于源语句的不同区域。

针对循环神经网络存在的长距离依赖等问题，2017年谷歌右发布了Transfomer架构。Transformer模型仅使用自注意力机制和前馈神经网络，包括自注意力子层、前馈神经网络子层、残差连接和层标准化等模块。其中自注意力机制可以看成对注意力机制的的一种特殊情况，在self-attention中，将当前词直接与序列中其他词的建模，不在像循环神经网络依赖于信息传递建立单词之间的联系。这样一来，单词之间的联系就是相对独立的，从而提高了系统对序列的处理速度。

#### Seq2Seq模型的简易实现

##### 2.1 深度学习框架的选择

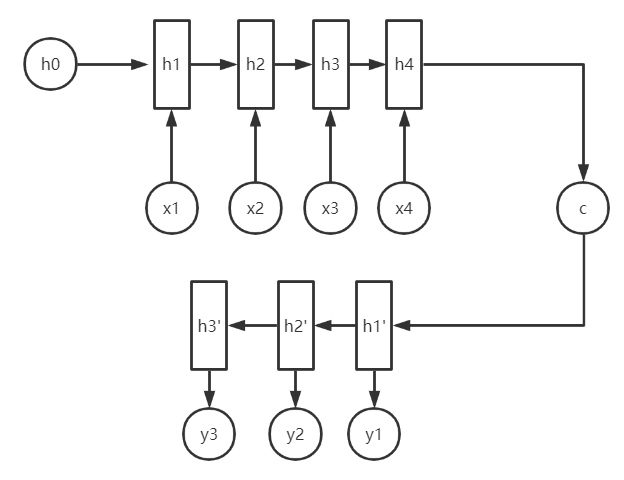
在本次实验中，我使用的深度学习框架为PyTorch。与其他框架不同，PyTorch是一种基于动态图计算的深度学习框架，因为动态图计算是在运行过程中被定义，所以可以多次定义多次运行。

与其他框架相比，PyTorch更容易上手，具有Python编程基础，就不需要像TensorFlow一样，要去学习一些额外的概念。PyTorch具有强大的GPU加速的张量计算，一经发布便得到了广泛使用。出于对未来学习的计划，借本次实验的机会初步使用PyTorch，增强自己的动手能力里。

##### 2.2 Seq2Seq模型的结构

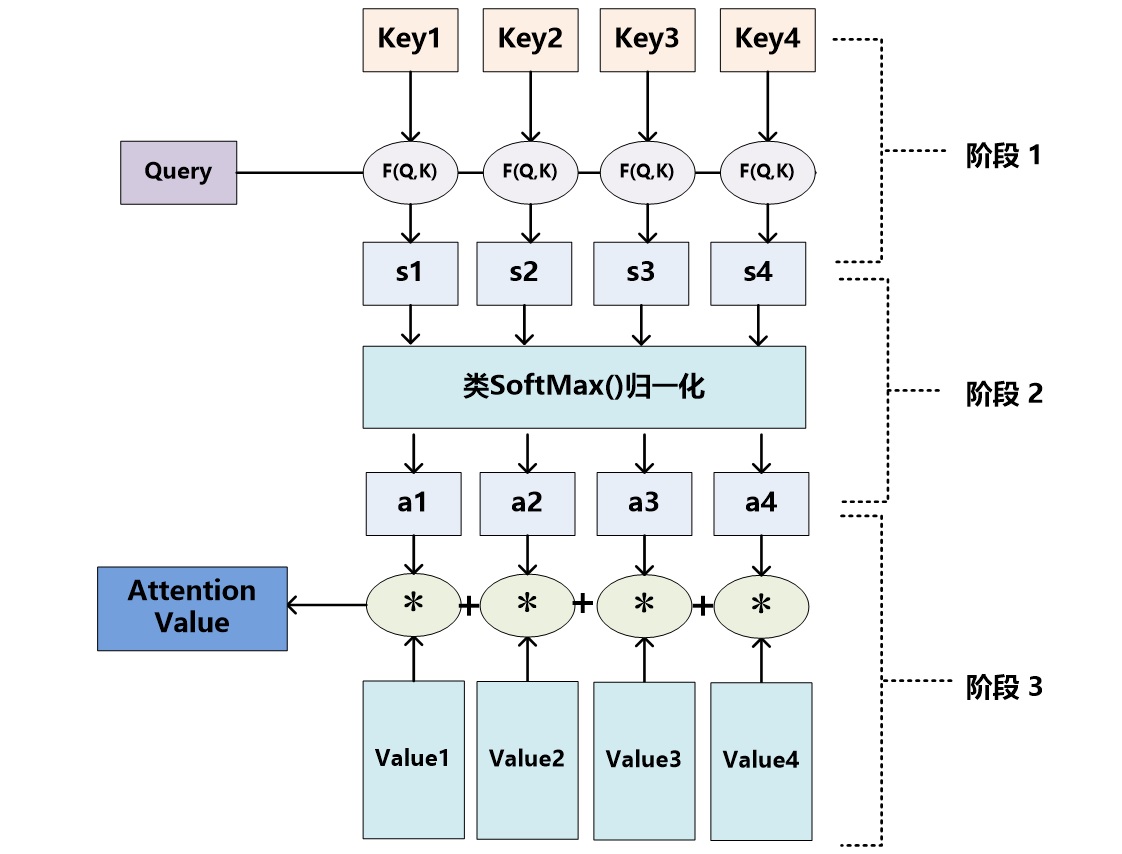
Seq2Seq模型是输出长度不确定时采用的模型，其属于encoder-decoder结构中的一种。故他的结构主要分为Encoder层、Decoder层，而本次实验也加入了Attention层。

Encode和Decoder实际上是一个RNN、LSTM、或GRU的网络，前者负责将输入序列压缩成指定长度的向量，后者则负责根据语义向量生成指定的序列。如图一所示，Encoder结构先将输入数据编码成一个上下文向量c，拿到c之后，就用另一个循环神经网络对其进行解码，这部分被就是Decoder。具体做法就是将c当做之前的初始状态h0输入到Decoder中。



**图一**

在encoder-decoder模型中，每个英文单词对于翻译目标单词的贡献是相同的，在翻译较短的句子时也许没有什么问题，但是当输入句子较长时，仅靠一个中间语义向量来表示，会忽略很多单词自身的细节，使得翻译的结果有很大偏差。所以引入了Attention模型，体现出英文单词对于当前翻译单词不同的影响程度。如图二所示，Attention的计算过程可分为三阶段，第一阶段根据Query和Key计算两者的相似性或相关性，第二阶段利用SoftMax函数对上一阶段的原始分值进行归一化处理，第三阶段根据权重系数对Value进行加权求和。



**图二**

本次实验采用了GRU网络，它是LSTM网络的一种效果较好的变体，所以也可以解决RNN中的长依赖问题。在GRU中引用了更新门和重置门。更新门用于控制前一时刻的状态信息被带入到当前状态中的程度，更新门的值越大说明前一时刻的状态信息带入越多。重置门控制前一状态有多少信息被写入到当前的候选集 上，重置门越小，前一状态的信息被写入的越少。每个GRU单元针对输入进行下面的函数计算：



式中表示sigmoid激活函数，其取值范围为(0,1)，即越接近0表示越容易遗忘，越接近1越容易被保留。结合当前输入和上一节点传递下来的隐藏状态，GRU可以得到当前节点的输出和传递给下一个节点的隐藏状态。

LSTM和GRU都解决了长期记忆和反向传播中的梯度等问题，但是后者比前者少了一个门函数，所以运行速度要快，也就是说二者的实验效果相似，但是GRU更易计算，这也是GRU现在流行的原因之一。

### 实验过程

1. 预处理

英文用nltk分词，中文则使用jieba分词。创建函数load\_data用来逐句读取文本，并在每个句子的开头加BOS，结尾加EOS。创建函数create\_dict用来统计文本中每个词出现的频数，并用出现次数较多的一些词创建词表，其中添加UNK来表示词典中未出现的词，添加PAD表示后续的填补内容，同时建立word2index及index2word的map映射。

创建一个encode函数，进行句子编码，将句子中的词转换为词表中的index，并对按句子长度排序，使得长度相近的句子可以放入同一batch。用句子在原始文本中的行号创建每个batch的数据索引，对每个batch中的句子进行padding操作，使得同一个batch中的句子等长。

2.Encoder层

分别利用torch.nn模块实现以下内容，首先用Embedding将输入映射为词向量，接着用GRU实现双向GRU，最后用Linear将encoder的输出经过线性层和激活层变换为decoder的初始输入。在编码前还要将数据进行还原处理，即去掉先前添补的padding部分，并恢复batch中句子的原始顺序，将句子长度倒排序，以便便pad\_packed\_s-equence的使用。

1. Attention层

这里的匹配度使用双线性变换计算，即source的hidden输出H，target的hidden输出O，那么atten=O \* W \* H，其中W不是方矩阵，这样可以让编码器和解码器拥有不同维度大小的隐层状态向量。接着做mask，对于source和target中的padding部分用负数来代替，以此消除后续对softmax的影响。利用softmax得到概率分布，再将atten与source的hidden state即上下文输出做一个加权求和，然后利用cat函数做堆叠获取融合信息，经Linear转换后，再经tahn激活得到输出。

1. Decoder层

同样Embedding层将target输入查找词向量，建立单向GRU，创建mask并使用Attention层，然后经线性变换输出，再经过log\_softmax求出每个输出的概率分布，最大概率出现的位置就是预测的目标词在词表中的位置。

1. SeqwSeq模块

将上述三层整合，得到完整的模型计算过程：

1. srouce输入Embedding层srouce\_embed  
    (2)srouce\_embed经过双向GRU层，得到srouce\_hidden，srouce\_last\_hidden  
    (3)srouce\_last\_hidden经过线性层、tanh激活得到decoder的初始hidden输入target\_init\_hidden  
    (4)target输入Embedding层target\_embed  
    (5)target\_embed及target\_init\_hidden经过单向GRU层，得到target\_hidden  
    (6)根据srouce及target句子batch中的长度，创建mask  
    (7)srouce\_hidden和target\_hidden做双线性attention得到输出atten  
    (8)atent做mask后softmax归一化为概率分布  
    (9)srouce\_hidden与atten加权求和输出value  
    (10)value与target\_hidden信息融合后输入线性层、tanh激活输出为target\_output  
    (11)target\_output输入线性层、softmax后取对数，得到最终的target vocab size上的对数概率分布
2. 自定义损失函数

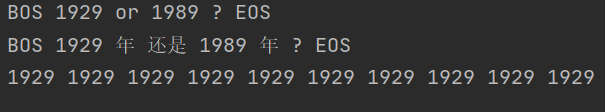
使用mask交叉熵的方法来计算loss，因为采用梯度下降，所以可以通过最小化其相反数的方法获得最大化目标词语的概率，再乘以mask，即句子的有效长度，最终取平均loss=sum(loss)/sum(mask)。

1. 初始值定义

batch\_size设置为8，embed\_size设置为50，enc\_hidden\_size设置为100，dec\_hidden\_size设置为200，模型使用Adam，学习率lr取0.001，总共训练200个epoch。

### 四、实验结果

训练后，设置一个测试函数，将中文数据集中的第一句话翻译为中文，其翻译结果如图三所示。



**图三**

可见结果并不如人意，由于本次实验条件有限，没有GPU支持，只能在CPU上运行，考略训练数据较大，所以我只保留了原数据集的一小部分，而对于神经机器翻译模型训练数量越大，效果越好，因此很可能是因为数据集较小，训练不足导致的。

### 实验总结

通过浏览相关信息，我意识到当前机器翻译仍面临着许多挑战。首先就是数据稀疏的问题，上述提到神经机器翻译依赖于大量有质量的语料，针对这个问题提出了一个多任务学习的多语言翻译模型。在进行多语言翻译的时候，源语言共享编码器，在解码端不同的语言使用不同的解码器。这样在源语言端就会共享编码器的信息，从而缓解数据稀疏问题。其二就是一些新兴的常识和专属词汇的翻译，针对这个问题将知识引入翻译模型，当词语没有被翻译时，就去查阅外部知识获取翻译结果，但是目前引入知识还是比较表层的，仍有待发展。还有就是语篇翻译的挑战，大部分的翻译系统现在所使用的翻译方法都是基于句子，从而导致有时候连起来看就觉得不连贯。针对这个问题提出了一个两步解码的方法。在第一轮解码中单独生成每个句子的初步翻译结果，在第二轮解码中利用第一轮翻译的结果进行翻译内容润色，并且提出使用增强式学习模型来奖励模型产生更流畅的译文。

这里也只是简单说了一些我平时浏览到的有关机器翻译发展的信息。由此可以看出机器翻译仍然面对着许多挑战，而这些挑战也必然会反过来推动翻译技术的不断革新。

通过语言分析和机器翻译这门课，我对NLP以及机器翻译的相关技术有了初步的了解，同时我也意识到其中蕴含的知识有多么的深奥和广泛。也引起了我对NLP的好奇心和兴趣，这无疑是在我开展未来学习的良好开端。