期中作业报告

0. 概述

模块1-6的代码分别保存在part1.py-part6.py中，而master.py负责调度所有模块的执行。
segphrase文件夹下保存了使用segphrase算法对中文分词所需的文件（其实是直接照搬了自己的国庆作业）。
giza-pp文件夹下保存了giza++算法的源文件（其实是直接从github上clone下来的），linux环境中make后得到了几个可执行文件：GIZA++，plain2snt.out，snt2cooc.out，拷贝到了父文件夹中。
c2e文件夹中则保存了giza++的运行结果。
前5模块都使用模块化编程，而模块6使用了面向对象编程。

1. 数据预处理

- 层次化编码使用了两种DFS：
  第一种是对文件目录进行DFS，得到所有文件对应的树形结构；第二种是对单个文件内部进行DFS，即使用正则表达式去匹配单个文件内所有的一级，二级和三级标题，并判断标题下是否存在回答：如果有回答，就对该回答进行层次化编码（output.txt）。再对所有的标题建立树形结构，得到json文件（output.json）。
- 句对齐的方法：
  将所有的回答按'.'与'。'切分，如果在某个回答中，中英文切分结果句子数量不同，那么就将整个回答段落扔掉;否则就按照切分结果将中英文句子一一对齐。再加上所有的标题，就能得到大约27000个句对。

2. 双语词典生成

- 使用segphrase方法对所有的中文句子进行分词：
  先将所有的中文句子输出到segphrase/input.txt中，然后调用segphrase/main.py对中文句子分词，分词结果保存在cuts.out中，再将分词前的中文句子和分词后的文本做汉字层面的对齐（因为我自己之前实现的segphrase只支持纯汉字的分词。。非汉字都被过滤掉了。。所以才需要一些额外的对齐操作）这样就得到了中英文的平行语料，分别输出到chinese和english中。
- 使用giza++生成词典：
  直接使用make生成的三个可执行文件来跑giza++算法，使用chinese.vcb，english.vcb，以及c2e/*.t3.final，按照最大概率来选择每个中文词汇对应的英文词汇，保存在dic中。

3. 计算词转移概率矩阵

- 直接使用英文语料计算词转移概率矩阵（包括BOS和EOS），因为矩阵很稀疏，所以可以使用dict保存。

4. 机器翻译

- 先使用jieba对输入的中文字符串进行分词，在根据模块2生成的词典得到对应的中文词汇。然后使用一种近似搜索算法来找到近似的概率最大排列：即从BOS开始进行BFS，每轮保留概率最大的K个（K取10000），一直搜索到EOS为止，选择概率最大的排列输出。特别地，设置了一个平滑因子R=0.85，表示有R的概率按照原矩阵转移，（1-R）的概率随机转移到任一英文单词。

5. 双语检索

- 类似之前的作业，直接使用正则表达式对所有的问题和回答进行匹配（中文先翻译再匹配英文），并返回所有答案。

6. 界面设计

- 先使用place方法在界面的左右两侧各放置一个Frame：左侧的Frame用于放置输入的控件，如输入框，复选框，搜索按钮，清空输出的按钮等，此外还有一个文本框用于显示翻译的结果。右侧的Frame则用于存放检索结果：点击搜索后，会先显示包含了检索内容的标题（按钮）。单击相应的标题即可查看详细的答案，同时可以使用滚动条进行上下滚动。

7. 结果与分析

机器翻译效果较差

- 首先，直接使用概率最大的候选词作为翻译结果，而丢掉了概率小一些的翻译，这样可能导致翻译得到的候选词只有单一时态，从而使正确的短语的单词之间转移概率很低，导致在最大化总概率的过程中，可能会选择一些错误的排列顺序，导致最后翻译结词序混乱。
- 其次，这种翻译模型无法翻译句中的虚词（如介词，冠词等等），导致翻译质量进一步下降。
- 再次，我认为直接使用词转移概率矩阵来计算概率本身就有问题。因为英文单词数量很多，而本次作业的语料又不大，故得到的转移矩阵其实十分稀疏，并不能反映真正的成句情况；除此之外，词转移概率矩阵并不能反映英文的"语法规律"，如果一些有固定结构的短语没有在语料中出现过，那么就很难去确定排列顺序（例如动宾短语，介词短语等等，显然不是每个同类短语都在语料中出现过，但每个同类短语的排列规律是相同的，与单词的词性有关）。
- 改进的思路：
  使用隐马尔可夫模型来代替词转移概率矩阵，即先对语料中的英文单词标注词性，得到相邻词的词性的转移概率矩阵，并得到每种词性对应每个单词的概率；再在翻译时取出概率前5大的候选词去搜索得到概率近似最大的排列，这相当于是利用了英文词性的语法，应该能够得到更好的翻译结果
  （如果隐状态能不仅仅是词性特征。。例如根据词的统计信息以某种方式去划分词的类别作为隐状态（对词向量化后进行聚类？）。。。应该能最大限度地挖掘并利用英语的语法对词进行排列）。

8. 心路历程（吐槽）

- 好不容易把模块四写完了。。然后一看。。这效果甚至还不如直接把英文单词按原序输出。。感觉仅仅靠调参的话也没什么救。。
- 然后想实现搜索的时候会显示所有标题的按钮，点进去之后能显示相应回答的功能。然后发现因为对于每个按钮都需要配置不同的命令，直接用一个lambda表达式似乎是不行的。然后经过一番思索，想起了闭包：在lambda表达式之外套一层函数就可以返回不同的函数，也就可以对不同的按钮实现不同的功能。
- 还有一个问题就是Frame不支持添加滚动条，导致按钮可能列不下。然后谷歌了一下，发现可以通过在Canvas上套Frame的方式对Frame进行滚动。但仍然有一个小bug：貌似Canvas的高度不能超过32767个像素，平均每个按钮会占35个像素的高度，按钮太多的话就可能会溢出然后导致循环。。。。。。
- 还有。。英文文档中有两个文件在中文文档里没有对应的文件。。。我就直接给删了（