## 实验思路

对训练数据进行处理，计算出单个字出现的次数和二元组出现的次数，考虑到出现在句首的字的特殊性对其单独计算，这里将概率的和转换为对数的加进行处理。对上述文件进行保存，将拼音表处理成字典形式重新存储

维特比算法中对于中间层的节点，我们只需要对前一层节点进行遍历找到前一层节点中到当前节点距离与其到起点距离只和最大值即可，最后对最后一层节点按概率从大到小排序，对每个节点回溯路径并保存

GUI上输入拼音字符串，中间用空格隔开，非拼音字符串直接在输出栏输入进去，在候选的选择上，首先将给出10个当前拼音串的结果，如果没有正确选项，可翻页到单个字的候选，选择之后将把所选字打印到输出栏中，并用剩余拼音更新候选栏，减少翻页次数

## 实验环境

Python3 Pycharm

Pyqt5 QtDesigner

## 代码结构

./init\_hmm.py 数据预处理，独立于gui程序，需要pinyin\_train.txt和pinyin.txt文件

./viterbi.py viterbi算法实现，外部通过调用pinyin2hanzi函数返回可能列表

./Solver.py gui界面调用维特比算法的接口，返回输入拼音串和单个汉字的预测列表

./input\_method.py 输入法gui的.ui文件通过使用PyUIC得到的.py文件

./input\_method.ui 使用QtDesigner得到的输入法界面.ui文件

./input\_method\_slots.py 处理信号和槽的窗口类

## 使用方法

命令行输入：

python init\_hmm.py

python main.py

在输入框输入拼音，中间用空格分开，点击确定，在候选框会出现候选词，使用0-9选择，+-翻页，若需要输入标点和数字需要直接在输出框里输入

## 测试击键次数

约为1500次

## 实验总结

总体上看实验过程比较艰难，而且在最后的测试阶段发现给出的pinyin.txt有很多错误，在结果上还是使用了全拼作为输入，没有进行编码比较遗憾，实验参考了网上的思路，在此表示感谢！