**拼音输入法实验报告**

张锦黔 201604014013

**一、问题分析**

**1.1 问题描述**

汉语拼音输入法的编码是依据汉语拼音方案（汉字的读音）进行输入的一类中文输入法。本次作业要求自己编程实现一个简单的汉语拼音输入法，即实现从拼音（全拼）到汉字（字串）内容的转换。其输入为存储在文本中的拼音串（每串一行，以空格分隔）；其输出为根据转换后的汉字串；问题提供国标一二级汉字和基于新浪新闻的九份语料库，其它训练资源可自备。

**1.2 问题抽象和模型建立**

此问题可视为一类通信问题，输入的拼音串为信号，所求输出为由拼音串推出的汉字串。

设,,,…,为用户输入拼音串；

设,,,…,为用户希望得到的汉字串；

由此建立基于字的二元模型：对于,,,…,，其中每相邻的两个汉字作为一个词，通过训练可获得汉字库中任意两个汉字组成一词的频数，进而可以得到该词出现的概率。其意义为：当前一个字为时，后一个字为的概率。

则拼音输入问题抽象为：

即求整体概率最大的句子，整体概率为各相邻字出现概率之积。

**二、算法设计**

**2.1 概率计算**

要求整体概率最大的句子，需要获得每个可能出现的组合的概率，即。

整个训练数据中，记、出现的频数分别为、，词出现的频数为，根据贝叶斯公式，将定义为：

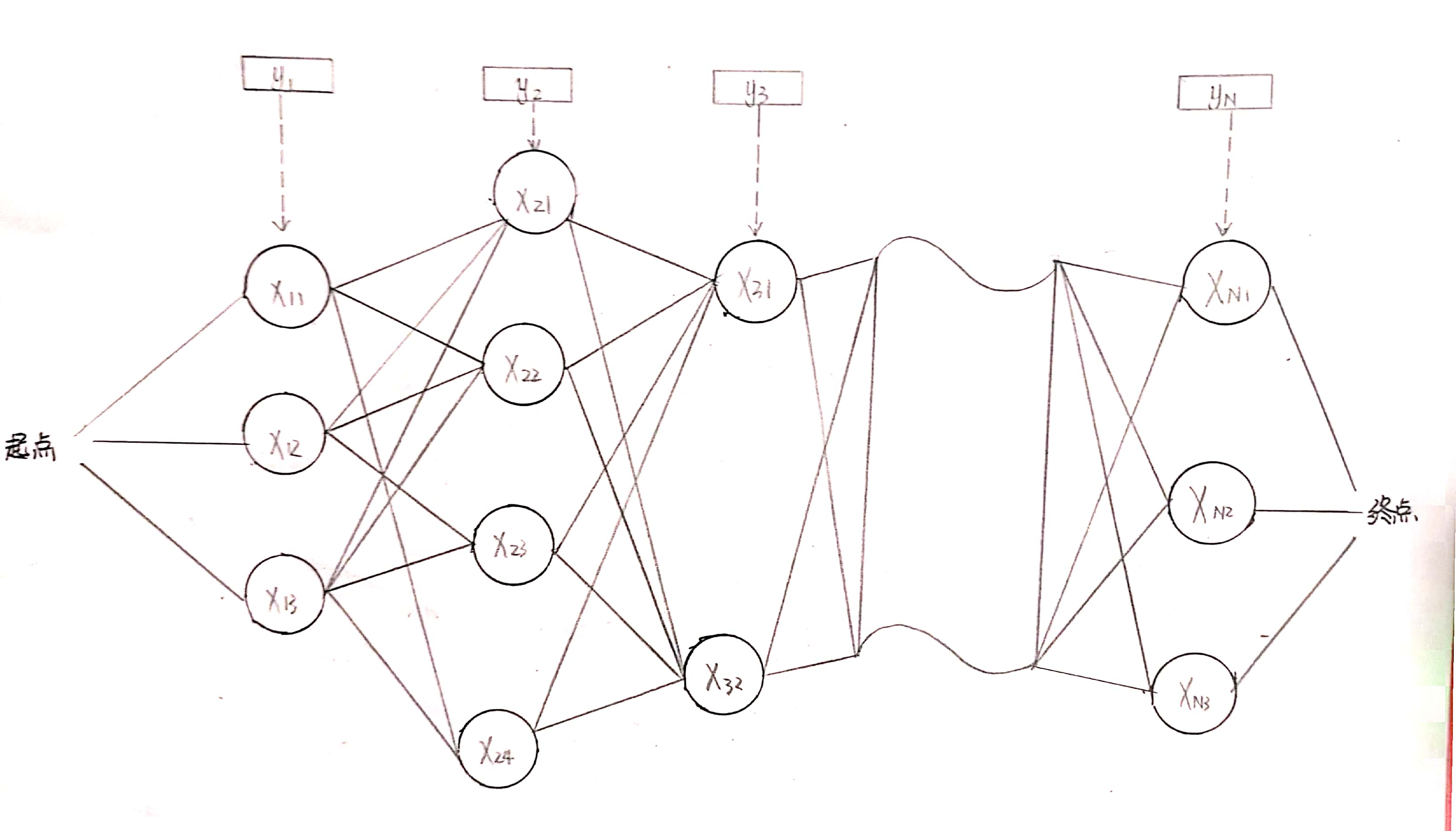
记拼音对应的所有可能的汉字在训练数据中出现的频数，则当第个拼音为时第个字为的概率：

为使数据平滑化，避免的情况，引入参数，将重新定义为：

该表达式表明，取决于词在所有中所占比例和在其同音字中所占比例。

**2.2 求最优序列的Veterbi算法**

基于,,,…,为用户输入拼音串，可通过拼音-汉字库查找到每个拼音对应的若干个候选汉字，从而建立如下有向无环图，此类图有明显的分层特点，每一层为一个拼音对应的若干汉字构成的节点，这种图成为晶格网络（Lattice Network）。



该晶格网络中，每个节点对应一个汉字，前后两层的节点之间有边连接。边的长度由两节点间的决定，则求输出的最优序列问题转化为求晶格网络中最短（长）路径的问题，可采用动态规划求解，具体算法为Veterbi算法，流程如下：

（1）.取当前层的当前节点为，计算其与前一层每个节点的；

（2）.查看前一层每个节点中记录的最短距离，并与上一步计算出的对应值相加，取和最小者记录在本节点内，即为从起点到本节点的最小距离。同时在本节点内记录与所取前层一节点的连接关系；

（3）.对当前层每一节点重复（1）、（2）操作；

（4）.对每一层重复上述操作；

（5）.取最后一层中存储值最小的节点即可得到最小路径长度和最小路径。

注：第（1）步计算第一层节点与前一层的时，由于不存在前一层，的表达式有所调整，见下文。

**2.3晶格网络边长的计算**

在2.2中所述晶格网络中，链接前后层节点的边的长度由两节点间的决定。且满足越大，越小。

选用负对数函数，将区间内的映射到，得到的表达式：

当所计算的是计算第一层节点与前一层的时，不存在前一层。这时表达式中的第一项无意义，因此计算时表达式仅采用第二项。修改后的概率表达式的物理意义为字在同音字中出现的概率。

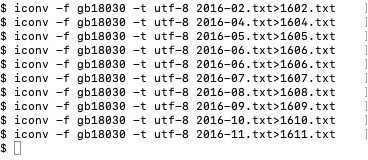
**三、程序设计**

**3.1 预处理工作**

（1）文件的预处理

编码转换：

程序不可避免地需要大量使用文件操作，使用txt格式文本文件作为程序输入、输出操作的对象，编码为utf-8。因为涉及汉字，选用locale库进行字符编码国际化，用wchar\_t和wstring作为存储字符、字符串的数据类型。C++的locale库对utf-8的支持较好，但在C++程序内部进行gbk与utf-8之间的编码转换较为困难，因此在MacOS操作系统下使用系统命令对sina\_news\_gbk中的文件进行编码转换。MacOS控制台中提供了iconv命令进行该操作：



（2）读取和存储汉字-拼音库

读取：

题目提供了存储国标一二级汉字及其拼音的txt文件，对其进行编码转化之后放置于程序目录下。该文件内部组织结构为每行开头一个拼音，之后以空格分隔记录此读音对饮的若干汉字。因此先使用getline()函数获得每一行数据，之后建立宽字符串流对象（wistringstring），即可按空格对拼音和单个汉字进行拆分。

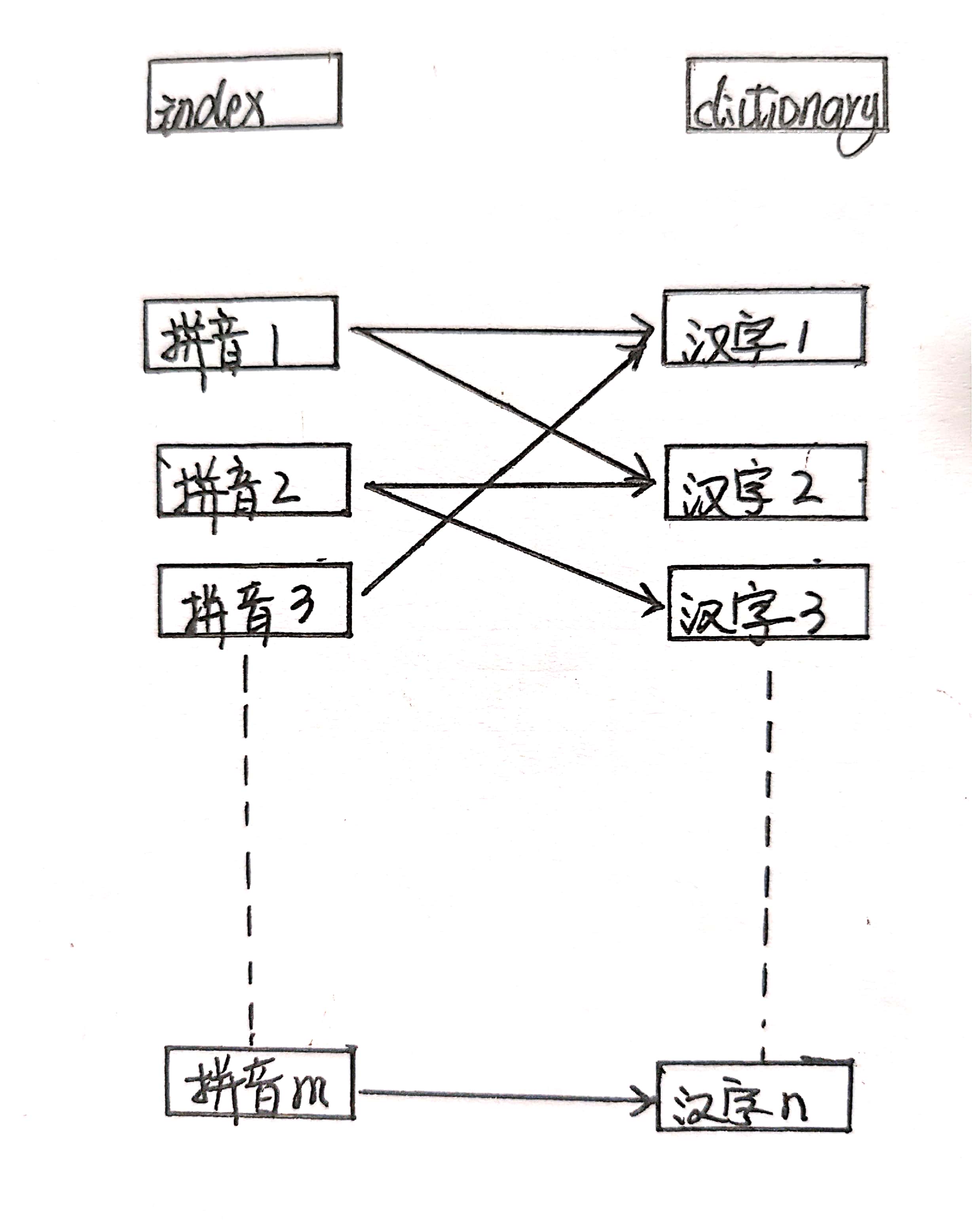
存储：

跟据读取流程，程序每getline()一次得到的信息依次为：一个拼音，若干个汉字。分别声明拼音类（class prn）和汉字类（class chr），其中分别包含wstring类型和wchar\_t类型成员用于存储拼音和汉字。

使用STL容器建立两个向量（vector）：index、dictionary，分别存储拼音类和汉字类对象，每当出现一个新拼音或汉字，将其加入相应的容器。拼音与汉字的对应关系由拼音类中的指针数组存储。

容器中的汉字应保持唯一性，而实际上汉字种有很多多音字，因此每从汉字-拼音库中获得一个新的汉字字符，应首先在dictionary中查找此汉字是否已经存在。向量index中的拼音亦同理。

即可建立如下图所示存储结构：



**3.2 训练**

模型训练的目的是得到汉字两两组合的频数，其具体流程为：

（1）打开指定的训练文本；

（2）读取文本中第一个字符a和其后一个字符b，分别存储于wchar\_t类型变量内；

（3）在3.1中已经预加载好的dictionary内查找a和b，若两个字符都存在，即两个字符都为汉字，则得到了一个词。进行（4）操作，否则跳转到（5）操作；

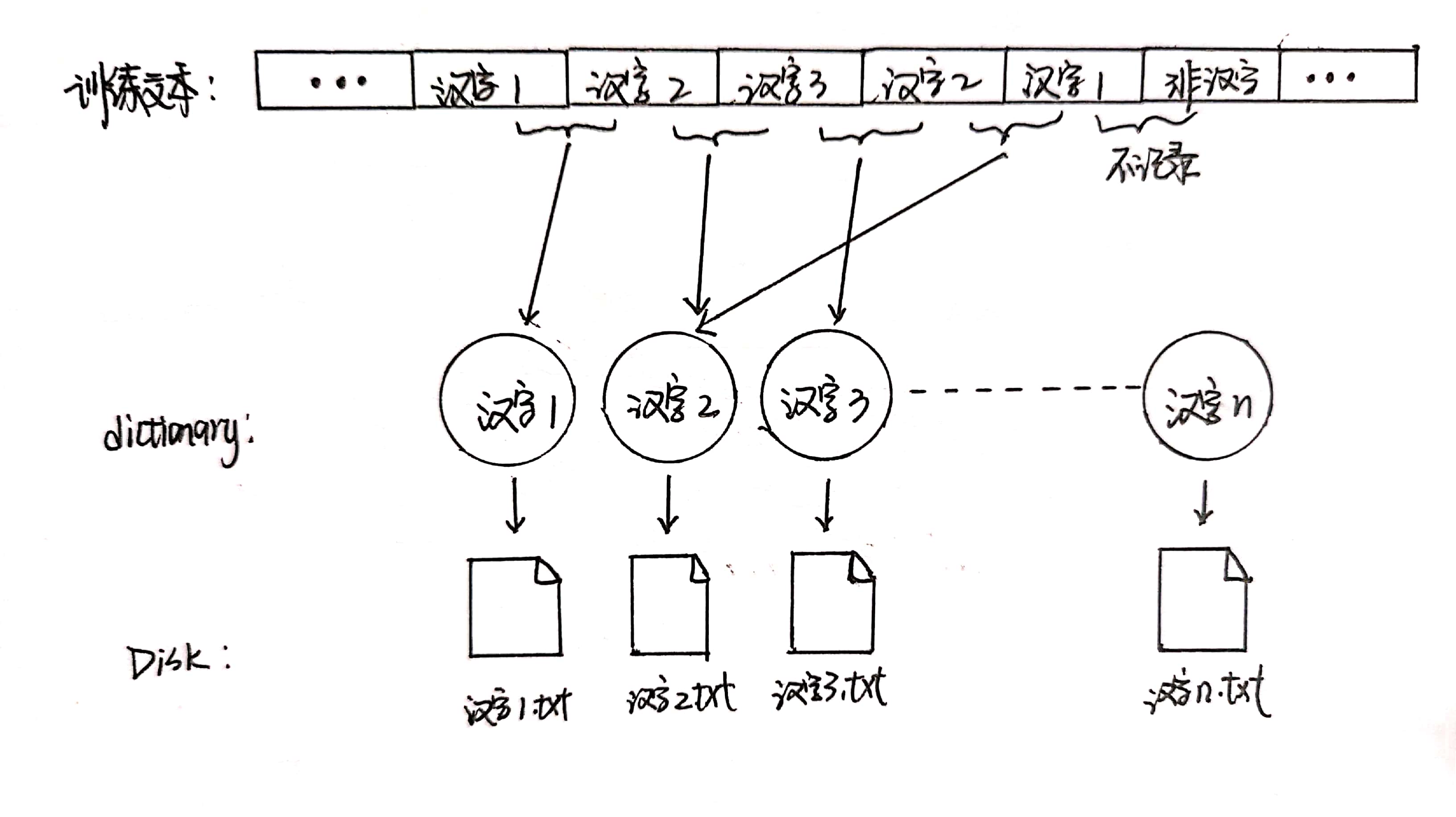
（4）在dictionary中汉字a对应的汉字类对象中，有专门记录其后续出现的汉字的链表，链表的节点记录汉字和出现次数。在链表中查找b，若存在，其次数增加1，若不存在，创建新节点，次数初始化为1。

（5）读取文本中下一个字符和其后一个字符（以此类推），转到（3）操作，从而遍历训练文本。当训练文本全部读完时，流程结束。

多文件存储：

训练结束时会产生大量的数据，可以将这些数据存储在磁盘中，省去每次均需要训练的麻烦。当日后读取训练结果时，往往不需要所有汉字的训练结果，只需要加载输入的拼音涉及到的汉字，因此训练结束后可以创建多个文件对训练结果进行存储。本程序采用了每个汉字一个文件的存储方式。

至此，完成了下图所示训练流程和存储结构：



**3.3 拼音串向汉字的转化**

转化流程为：

（1）从输入文件中用getline()读入一行拼音串，用wistringstring将其按空格拆分为单个拼音；

（2）根据每个拼音建立一个层类（class level）对象，根据此拼音对应的每个汉字建立若干节点类（class node）对象存储于层类对象中。至此完成了晶格网络的创建。

（3）按Veterbi算法（见2.2）求得最短路径并输出。求最短路径的过程中，节点类对象会从磁盘读取训练数据以计算。

（4）回到（1）重复上述操作。直至读完输入文件中所有行。

**四、实验结果及分析**

**4.1 实验结果**

改变的值，观察拼音-汉字转换的准确率发现，值越小，准确率越低。当取0.95时，题给测试数据转化的准确率达到66.62%。

**4.2 错误原因分析**

（1）多音字问题：汉字中存在大量多音字，一个含a、b两种读音的字A在音a的频数可能很高而在另一个音b的频数很低，这时音b的译出会受到字A的干扰。例如“ju you”（具有）有时被误译为“车有”。

（2）模型过简：二元模型无法充分体现汉语语句中更长的词和句子成分间的特征，使用三元或更高级模型可以改善准确率。

（3）测试拼音串内容与训练样本偏离：观察译文可发现，程序对于“走出清华大学特色的世界一流大学道路”“取消或听证的涉及个人事项的行政事业收费包括”等长串的转换依然保持高准确率，但很容易在转换“斧王战死沙场”（网络游戏配音）、“花谢花飞花满天”（诗句）等短句时大量出错，这与训练样本来源于新闻稿有关。

**4.3 参数影响分析**

参数决定表达式中前后两项的权值，表达式第一项代表词的出现概率，表达式第二项代表单字的出现概率。第二项的存在可以避免训练中一些极端情况干扰数据，但若第二项占据越来越高的比例时，算法将向“列举每个拼音对应的出现概率最大的汉字”退化。