

MYIME 项目设计文档

181830044 董宸郅

2020.12

需求分析

- 完成一个功能基本完备的中文拼音输入法
 - 支持全拼与简拼
 - 支持加入用户自定义的词语
 - 对于有歧义的情形，采取**最长匹配**的策略
 - 对结果进行词频排序，且用户输入过的词语**优先级**最高
 - 用户可以进行**多次选择**，直至得到期望的结果

需求分析

- 语料库需要自行从pinyin.txt与data.txt中提取
 - 过滤汉字外的无效内容
 - 对记录到的词语进行词频统计
 - 设计合理高效的方式存储数据，方便后续取用
 - 注意对于多音字的处理

模块与功能划分：整体

- 本次项目采用过程式程序设计，分为以下两个文件/模块：
- `preproc.cpp`:
 - 预处理时，从pinyin.txt与data.txt中提取语料库
 - 正常启动时，读取部分信息，完成初始化
- `type.cpp`:
 - 完成具体打字操作

模块与功能划分：PREPROC.CPP

预处理存储结构设计：

- 将所有的字/词通过统一的树状目录结构进行存储
- 单级目录名为对应的声母
- 目录内含有：文本文件与子目录
 - 文本文件代表拼音至此结束，文件名为全拼.txt，里面存储了对应的字/词及词频
 - 子目录代表还存在更长的拼音序列，目录名即为下一个声母

模块与功能划分：PREPROC.CPP

预处理存储结构设计：

- 举例说明：
- 现在要存一个词语“我爱你”，全拼为 *woaini*
- 则这个词语储存的路径为 `~/w/a/n/woaini.txt`
- 打开这个文件，里面就有所有全拼为 *woaini* 的词语（即可能也有“窝哎泥”），以及对应的词频
- 后续用户输入的词语也用这个规则存储，词频直接设为 `INT_MAX`，以保证优先性

模块与功能划分：PREPROC.CPP

- void trans()
 - 读取pinyin.txt中的内容
 - 将“拼音->汉字”转化为“汉字->拼音”，用 `map<wchar_t, vector<string>> pronunc` 存储
 - 存储pinyin.txt中的汉字到对应目录

模块与功能划分：PREPROC.CPP

- void proc_data()
 - 读取并过滤data.txt中的信息
 - 利用regex: `L"[\u4e00-\u9fa5]+"`来判断是否为汉字词语
 - 此处真正的难点在于读取和存储汉字类型（属于宽字符），对于pinyin.txt中的单个汉字尚可采取读入2个char再转化为w_char的方式，但此处只能以wstring读入
 - 根据踩过的坑，wcin、wcout、wfstream等均需经过类似以下操作才能正确处理中文字符：
 - `setlocale(LC_ALL, "chs");`
 - `wfstream wfout;`
 - `wfout.imbue(std::locale("chs"));`

模块与功能划分：PREPROC.CPP

- void store_phrase(wstring phrase)
 - 将从data.txt中提取到的字词 `phrase` 存入对应目录，并添加词频
 - 通过 `pronunc` 获得字词的读音，并获取其声母
 - 利用regex “[zcs]h.+” 来判断是不是zh、ch、sh
 - 遇到pinyin.txt中不存在的生僻字（往往是繁体字），忽略
 - 遇到含多音字的词语，先存到duoyinci.txt中，可借助 `pypinyin` 这个python工具确定其读音，之后再统一存储

模块与功能划分：PREPROC.CPP

- void load_info()
 - 从已有的一些记录文件（ignore.txt, duoyinzi.txt, duoyinci.txt等）中读取信息，避免重复处理
- void store_duoyinzi()
 - 存储多音字方便使用，同样为了避免重复处理

这两者只是辅助性程序，对应文件也都是临时文件...

模块与功能划分：TYPE.CPP

- 打字分为全拼与简拼两种模式
 - 通过 regex “(((([csz]h|[abcdefghijklmnopqrstuvwxyz])?)+)” 判断输入的拼音是否全为声母，若匹配则默认进入简拼模式
 - 这里需要注意，有些拼音，如 *fang*，本身既可解释为全拼，也可解释为简拼，存在歧义
 - 为应对上述情形，可选择输入enable-qp/jp以强制开启全拼/简拼，否则优先简拼；再输入disable-qp/jp以撤销
- type_main()是打字入口程序，也是给main函数的接口
 - 判断当前是简拼还是全拼
 - 全拼时（简拼同理），调用 quanpin(input, pinyin, full_pinyin) 与 qp_choose(pinyin, full_pinyin) 完成拼音解析

模块与功能划分：TYPE.CPP

- 全拼涉及函数：
 - void quanpin()
 - void qp_choose()
 - bool qp_translate()
- 简拼涉及函数：
 - void jianpin()
 - void jp_choose()
 - bool jp_translate()

程序与算法实现

- 以下介绍全拼的完整算法过程
- FYI: 由于我的数据存储特点，简拼与全拼的算法几乎完全一致，只在个别地方有细微差别，也会一并指出...

程序与算法实现

- 首先调用 `quanpin()`，通过 `regex_search()` 分割词语中每个字的拼音，存储在 `vector<string> pinyin` 中，顺便得到整个词语的全拼 `string full_pinyin`
- 要实现拼音字母的按字分割，且遵循**最长匹配原则**，难点在于写出正确的 `regex`，（可以搜到很多表示拼音的正则，但不保证满足最长匹配原则）此处给出我的实现：
 - `(ang?|a[io]?|ou?|e[inr]?|pou|m[io]u|[bmp](ia[no]|[aei]ng?|a[io]?|ei|ie?|o|u)|me|fou|wai|[fw]([ae]ng?|a|ei|o|u)|dei|diu|[dt]([aeio]ng|an|a[io]?|e|ia[no]|ie?|ou|uan|u[ino]?))|lun|[nl](i?ang?|a[io]?|[eio]ng|ei|iao|in|i[eu]|ou|uan|u[eo]?|ve?)|ni|nen?|le|lia?|[ghk]([ae]ng?|a[io]?|ong|ou|uang?|uai|u[aino]?)|ke|[gh]ei?|[jqx](i(ang?|ao?|e|ng?|ong|u)?|uan|u[en]?)([csz]h?|r)([ae]ng?|ao|e|i|ou|uan|u[ino]?)([csz](ai?|ong)|[csz]h(ai?|uai|uang)|zei|[sz]hua|([cz]h|r)ong|y([ai]ng?|ao?|e|i|ong|ou|uan|u[en]?))`
- 对于简拼，就要简单得多，只要每次取一个声母即可

程序与算法实现

- 接下来调用 qp_choose() ，伪代码如下：

```
void qp_choose(pinyin)
    left = 0 //当前起点
    right = pinyin.size() - 1 //当前终点
    cur_end = 0 //当前已完成的终点
    while cur_end < pinyin.size() - 1:
        qp_translate(pinyin, res, left, right, 0, cur_end)
        sort results based on frequency and show them to user
        if user does not choose any:
            if cur_end > 0: //结果太长
                right = cur_end - 1
            else:
                show no results available
                break
        else:
            record current result
            left = cur_end + 1
            right = pinyin.size() - 1
```

程序与算法实现

- qp_translate() 伪代码如下：

```
//left 起点 right 终点 cur 当前递归位置 end 成功返回的递归终点
bool qp_translate(pinyin[...], &res, int left, int right, int cur, int& end):
    if cur < right AND current path contains directory named $the next shengmu$:
        //go to that path
        if qp_translate(pinyin, res, left, right, cur+1, end) is true:
            return true
    if current path contains file "pinyin[left~cur].txt":
        get res from that file
        end = cur
        return true
    return false
```


请输入拼音，回车结束：
qingdajiajianchilajifenlei
1. 请 2. 轻 3. 清 4. 轻 5. 清 6. 氢 7. 轻 8. 清 9. 氢 10. 轻 11.
7. 清 18. 氢 19. 轻 20. 清 21. 氢 22. 轻 23. 清 24. 氢 25. 轻 26.
2. 清 33. 氢 34. 情 35. 轻 36. 清 37. 氢 38. 情 39. 青 40. 轻 41.
7. 氢 48. 情 49. 青 50. 轻 51. 清 52. 氢 53. 情 54. 青 55. 轻 56.
2. 轻 63. 调情 64. 清 65. 氢 66. 晴 67. 情 68. 蜻 69. 青 70. 轻 7
蜻 77. 青 78. 轻 79. 调情 80. 氰 81. 氢 82. 晴 83. 情 84. 庆 85.
91. 蜻 92. 苘 93. 罄 94. 紫 95. 筭 96. 磬 97. 氰 98. 槩 99. 晴 10
. 倾
请输入对应标号：
(输入0表示没有符合的选项)
1
1. 大家 2. 打架 3. 打假
请输入对应标号：
(输入0表示没有符合的选项)
1
1. 坚持 2. 尖齿 3. 减持
请输入对应标号：
(输入0表示没有符合的选项)
1
1. 垃圾 2. 拉吉
请输入对应标号：
(输入0表示没有符合的选项)
1
1. 分类 2. 酚类
请输入对应标号：
(输入0表示没有符合的选项)
1
请大家坚持垃圾分类

请输入拼音，回车结束：
jqr
1. 机器人 2. 几千人
请输入对应标号：
(输入0表示没有符合的选项)
1
机器人

THE END...