EnglishHelper 设计文档

组员介绍

设计：计42**胡润邦**

前端开发(UI)：计42**方燠**

后端开发(功能函数)：**杨天龙**

细节设计

1. 将UI代码和功能函数代码分离，UI只需调用函数接口
2. 功能函数func.cpp，围绕EnlglishHelper支持的4个功能，提供以下几个主要函数

单词查询部分Query

QueryPair\* query(QString a);

//查单词，传入单词，返回result类，有两个成员

//result->frist 为QString单词

//result->second 为QString\*亲缘单词数组，以"###"表示结束

记背、测试部分Test

TestPair\* startTest(int tot);//测试准备

//在Test开始时调用

//传入Test的单词个数，返回长度为测试个数的result数组，result类有三个成员

//result[i]->first 为第i个单词的User编号

//result[i]->second[j]分别为4个相似单词

//result[i]->third为正确单词的编号 (即result->second[ result[i]->third ]为正确单词)

void answerForTest(int id,Answer answer);

//Test过程中，答完每一道题时均需调用，用来记录答题信息

//传入单词的编号和这个单词是否答对

void endTest();

//Test结束需调用，处理记忆规律和更新用户文件

分析文章部分Article

QString analysisArticle(QString article,int tot);

//分析文章，传入QString，返回生单词(包含回车)

1. 外部文件有单词库dictionary.txt和用户记忆库user.txt两部分，它们分别对应数据库Word类和User类。

Word类内容如下

Class Word{

string word;//单词

string info;//释义(包含所有要求内容，用回车符隔开)

};

User类内容如下：

class User{

public:

QString word;

Type type;//0生词 1灰词 2熟词

int time;//最近访问时间

int totVisit;//总访问次数

int totWroing//总错数

int continueCorrect;//连续正确

int id;//在单词库中的编号

};

1. 记忆遗忘规律——影响记忆的因素

对一个单词的记忆程度：

1.和时间有关

2.和见到这个单词的总次数有关

3.和总错误次数有关

4.和连续答对次数有关

1. 熟词、灰词、生词的分类

1.词库分为用户记忆库和单词库两部分

2.一开始记忆库为空，单词库里的单词增减与记忆库无关

3.每次测试时，若需要抽取tot个单词，先从灰词里抽取min（60%\*tot，总生词数），生词里抽取min（总生词数，30%tot），熟词里抽取min（10%\*tot，熟词数）

若抽取数不够tot，按灰词，生词的顺序从记忆库中抽取单词补足

若还是不够，就从大单词库中随机选择单词加入记忆库中的生词部分，并加入测试，直到达到tot个。

4.开始测试

比对当前时间和最后访问时间

熟词：时间差大于5d且连续答对<=1次划入灰词

时间差大于10d且连续答对<=3划入灰词

灰词：时间差大于5d且连续答对<=1划入灰词

时间差大于10d划入生词

5.测试中：分答对和答错两种情况

每开始测一个单词

修改见到这个单词的总次数，修改最后访问时间

答对时

修改连续答对次数

答错时

修改总答错次数，连续答对次数改为0

测试结束后对刚测试的单词进行重新分类

熟词：答对不管，答错划入灰词

灰词：连续答对次数达到3次划入熟词，连续答对少于3次且 总答错次数/总次数 > 70%划入生词

生词：答对且 总答错次数/总次数 <=50% 划入灰词，答错不管

6.测试结束，保存用户信息

Func.cpp里主要功能函数的实现

每一个主要函数都分成很多个不同子功能函数，这些子函数分别放在不同的namespace里以避免冲突。

子函数的详细内容和作用限于篇幅不一一细说，详情可见func.cpp。其中主要包括

1. 寻找亲缘单词(词根相同词性不同的单词)
2. 寻找相似单词(长得很像但是释义上完全没有关系的单词)
3. 载入数据库，测试过程中的自动储存
4. 建立数据库索引
5. 获取系统时间并计算
6. 抽取释义中的中文部分并分离出例句只留下单词意思
7. 测试单词的提取，生词熟词灰词的划分
8. 将文章切分为一个个字符串，过滤空格和数字，处理首字母大小写

。。。。。。

等等功能