Lab4实验报告

> 实验题目：最短路径中文文本分词
>
> 姓名：侯新铭
>
> 学号：2021201651

1. 需求分析

中文文本没有自然分界符，对文本语义的解析需要进行（相较于英文等语言）额外的分词`segmentation`操作。本次lab目标基于词库，实现一个输入中文文本，返回成功分词后文本的基础软件。

(1) 输入

在项目`segmentation.exe`程序的输入框中输入待分词中文文本。

> 可支持文本长度：经文本长度测试，单次切词最多可支持约1万字文本
>
> 运行时间：几乎可忽略不计

(2) 输出

在交互界面的输出框显示分词后的字符串，格式为以"/"间隔的纯汉字文本。

(3) 功能实现

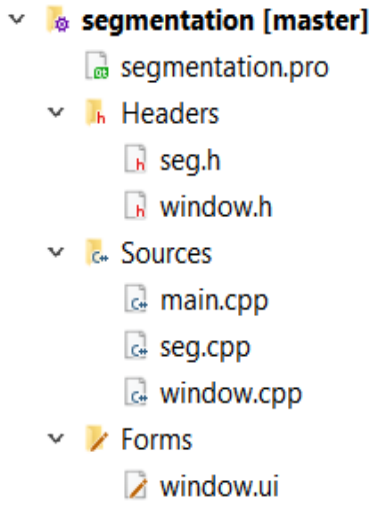
我开发了一个基础的软件程序，在其可视化界面中，用户可交互式输入中文文本，点击分词按钮后即可立即看到分词后结果；后续用户直接更改其输入的文本内容，实现实时分词。

(4) 使用样例

2. 概要设计

项目架构

①使用Qt Creator开发项目主体，项目文件包括：



②头文件调用关系为:

```
|-- main.cpp
        |-- window.h
            |-- seg.h

|-- window.cpp
        |-- window.h

|-- seg.cpp
        |-- seg.h
```

③头文件的主体分别为：

seg.h

```
#ifndef SEG_H
#define SEG_H
#include <queue>
#include <string>
#include <array>
#include <set>
using namespace std;
#define LEN 640
#define INF 0X7FFFFFFF
class Seg
{
public:
    // 载入dict.txt到set类型的dict变量
    bool loadDict(const string &location);
    string cut(string &s);
```

```
private:
    set<string> dict;
};

#endif
```

window.h

```
#ifndef WINDOW_H
#define WINDOW_H

#include <QMainWindow>
#include "seg.h"


QT_BEGIN_NAMESPACE
namespace Ui { class window; }
QT_END_NAMESPACE

class window : public QMainWindow
{
    Q_OBJECT

public:
    window(QWidget *parent = nullptr);
    ~window();

private slots:
    void on_pushButton_clicked();

private:
    Ui::window *ui;
    Seg seg;
};
#endif
```

3. 详细设计

(1) 分词功能实现

①词典加载

```
#include "seg.h"
#include <fstream>
#include <codecvt>
#include <locale>

// 载入dict.txt到set类型的dict变量
```

```
bool Seg::loadDict(const string &location)
{
    ifstream fin(location); // 通过ifstream流读取文件
    if (!fin.is_open())
    {
        return false;
    }

    string line;        // 将文件逐行读取到字符串line中，截取第一部分
    while (!fin.eof()) // 读取到文件末尾的EOF前一直执行while循环
    {
        getline(fin, line);
        int end = line.find_first_of(' '); // end对应第一列后的空格的索引下标,也即单词字符串的长度
        if (end != -1)                     // find_first_of()函数返回值不为-1，即找到了所给' '字符
        {
            dict.insert(line.substr(0, end)); // 将该词语子串插入到类内的private变量dict中
        }
     }
    return true;
}
```

②分词函数

采用最短路径匹配分词算法实现，主体循环遍历文本全部字符，其内包含如下3部分：

- 借助分隔符切分出由中文字符构成的单句
- 基于词典构建单句对应的有向无环图
- 执行dijkstra算法进行分词
- 将该单句切分出的词逆序添加到路径向量中

遍历完全部单句，主体循环结束，得到了由全部分词构成的路径向量，添加"/"作为分隔符，转化为分词后字符串，即为所求。

```
// 使用dijkstra算法获得最短路径
string Seg::cut(string &s)
{
    int startPos = 0;
    int fullLen = s.size();
    vector<string> path; // 使用vector记录最终所求的最短路径，方便进行插入、倒置等操作
    int count = 0;       // 用来记录上一句插词结束后path中的词数，作为下一句向path中插词的位置标记

    while (startPos < fullLen)
    {
        // 判断中文字符方式：基于中文字符由3字节构成，转化为unsigned int必然大于0x7f
        while (!((unsigned int)s[startPos] > 0x7f)) // 找到首中文字符的位置作为
```

```
sentence的起始
        {
            startPos++;
            continue;
        }
        if(startPos>=fullLen) break;
        int endPos = startPos;
        do
        {
            endPos += 3;                                                    // 注意到
一个中文字符在utf-8中占3个字节，故以3为步长
        } while (endPos < fullLen and ((unsigned int)s[endPos] > 0x7f)); // 找到连
续中文字符末位置作为sentence的末尾的下一位
        string sentence = s.substr(startPos, endPos - startPos);

        int num = sentence.length() / 3; // 恰好为当前sentence包含的词数
        array<int, LEN> g;
        g.fill(-1);
        array<array<int, LEN>, LEN> graph;
        graph.fill(g); // 定义2dim array graph,索引到的数值记录sentence各位置之间的可
达性。每个位置均初始化为-1，表示不可达

        for (int i = 0; i < num; i++)
        {
            graph[i][i + 1] = 2; // 每个字符和下一个字符显然可连通，距离为2，即闭区间
跨越2个词可达到
        }

        for (int i = 0; i <= num - 2; i++)
        {
            for (int j = 2; j <= num - i and j <= 12; j++) // dict最长词长度为12字
符，作为查找上限
            {
                string checkStr = sentence.substr(i * 3, j * 3);
                if (dict.count(checkStr))
                {
                    graph[i][i + j] = 1; // 更新gragh，表示i节点和i+j节点可处于同一个
词内，距离为1
                }
            }
        }
        // 下述部分执行dijkstra算法过程，对当前的sentence进行切词
        // 初始化
        array<int, LEN> d; // 记录dijkstra算法执行到当前时刻，各节点到初始节点的距离
        d.fill(INF);
        d[0] = 0;
        array<int, LEN> preNum; // 记录各节点当前所得的最小路径的前驱节点序号
        preNum.fill(-1);
        array<int, LEN> used; // 记录是否已作为最短路径节点使用过
        used.fill(0);
        for (int i = 1; i <= num; i++) // 初始化与初始节点直接相连的各节点
        {
            if (graph[0][i] > 0)
            {
```

```
                d[i] = graph[0][i];
                preNum[i] = 0;
            }
        }
        // dijkstra算法主体
        for (int i = 1; i <= num; i++) // 遍历寻找到未被使用过的节点中距初始节点的最短距离，该节点序号记为k
        {
            int min = INF;
            int k = 0;
            if (!used[i] and d[i] < min)
            {
                min = d[i];
                k = i;
            }
            used[k] = 1; // k节点被使用，更新used

            for (int j = 1; j <= num; j++)
            {
                if (graph[k][j] > 0 and graph[k][j] + min < d[j]) // 借助k节点可以构造出到节点j到初始节点的更短距离，更新d和preNum
                {
                    d[j] = graph[k][j] + d[k];
                    preNum[j] = k;
                }
            }
        }

        while (num != 0) // 从当前centence最后字符处往前不断找前驱节点，过程中把词edges构建为路径vector path
        {
            string word = sentence.substr(preNum[num] * 3, (num - preNum[num]) * 3); // 截取以当前位置为末节点的词edge
            wstring_convert<codecvt_utf8<wchar_t>> cov;
// 欲借助c++的iswpunct()和iswspace()来判断是否为中文标点，需先将string转化为wchar_t
            wchar_t ch = cov.from_bytes(word)[0];
            if (!iswpunct(ch) && !iswspace(ch)) // 判断为无（非）中文标点的word，插入path中固定位置count处，自然地实现了逆序插入
            {
                path.insert(path.begin() + count, word);
            }
            num = preNum[num];
        }
        count = path.size(); // 完成当前centence的分词后，更新位置标记count
        startPos = endPos;   // 更新startPos，向后推进
    }
    string output;
    for (int i = 0; i < count; i++) // 返回path中各词用"/"连接后的字符串
    {
        output.append(path[i] + "/");
    }
    return output;
}
```

(2) GUI界面开发

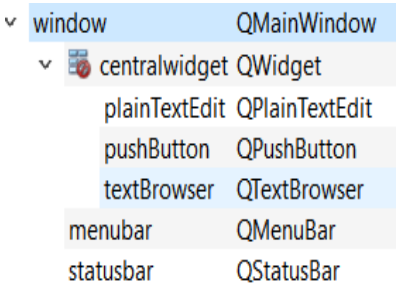
①使用Qt Creator创建`Qt Widgets Application`类型项目，生成主函数**main.cpp**为：

```
#include "window.h"

#include <QApplication>

// 使用Qt Creator创建project后自生成main代码，执行后，运行应用程序
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    window w;
    w.show();
    return a.exec();
}
```

②在项目中新建**Q MainWindow**类型文件组，命名为`window`，其中包括：

```
|-- window.h
|-- window.cpp
|-- window.ui
```

③进而使用Qt Designer设计GUI界面，主要包括输入框、输出框、push按钮三个组件，如下图所示：



④在**window.h**中的`class window`中实例化已经写好的`class Seg`以调用`loadDict`和`cut`函数，最后实现qt中的接口槽函数`on_pushButton_clicked()`：

```
void window::on_pushButton_clicked()
{
    string s = ui->plainTextEdit->toPlainText().toStdString();
    string output = seg.cut(s);
    ui->textBrowser->setText(QString::fromStdString(output));
}
```

4. 调试分析

时间复杂度：

- dict通过哈希表形式的set定义, 调用check函数的时间复杂度为O(1)；
- 对于有向图G=(V, E) 记n 为节点数，dijkstra算法在分词最短路径问题中，搜索`d[]`中最小距离需O(n)，内层循环构建新edge，更新参数的循环也需O(n)，算法总体时间复杂度为是O(n^2)。

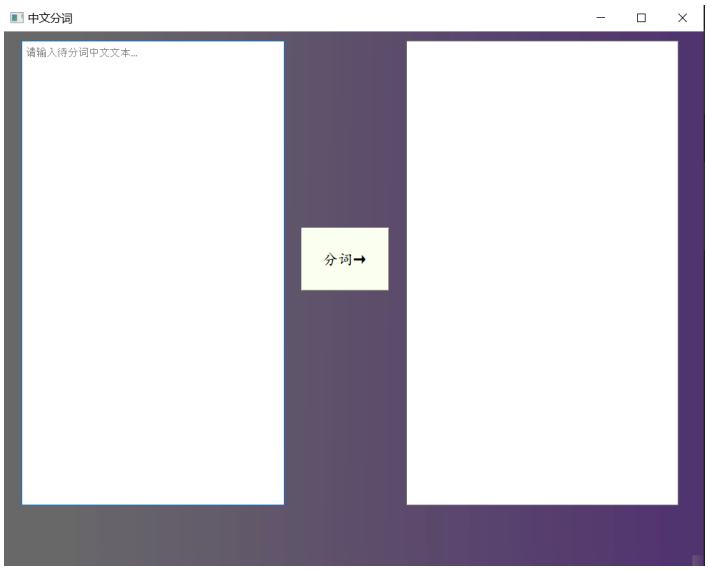
其他通过调试解决了的问题已在代码中详细注释。

5. 用户手册

- 项目在`Window`系统上开发，执行在Qt Creator由项目文件夹构建出的`segmentation.exe`程序即可。

> 请注意确保**window.cpp**中调用`loadDict`函数时所传路径为您的电脑中`dict.txt`文件的绝对路径。

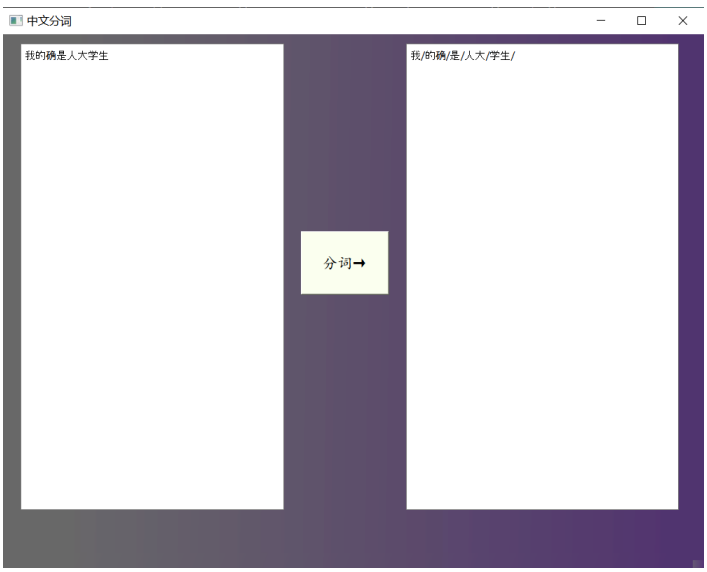
- 程序运行后，您将看到如下交互界面：



- 界面左侧设为输入框，右侧设为输出框，在输入框输入您待分词的由中文字符构成的文本（长度上限约一万字），单击界面中间的分词按钮，即可在右侧的输出框看到分词结果。

6. 测试结果

(1) 单句分词测试



(2) 长文本分词测试


中文分词
文继荣:勇闯无人区
尊敬的潘云鹤院士、怀进鹏书记、杜占元局长、李萌副部长、高润生副司长、黄铁军院长、刘伟校长、张磊校友、
各位嘉宾、老师们、同学们:
春水初生,春林日盛,春晖方暖。今天我们相聚中国人民大学,共同见证高瓴人工智能学院的成立,感谢学校和社会各界在学院筹备过程中的大力支持!
习近平总书记在中央政治局第九次集体学习中指出,要"勇闯人工智能科技前沿的无人区,努力在人工智能发展方向和理论、方法、工具、系统等方面取得变革性、颠覆性突破"。今年 2 月份春节后我们向潘云鹤院士汇报高瓴人工智能学院筹备工作的时候,他也向我们提出了学院建设要"勇闯无人区"的期望。
我们的愿景是打造一个能够影响和塑造未来人工智能时代的世界一流学院,为中国和全球思考并创造"智能而有温度"的未来。在人民大学这样一所以人文社会科学见长的著名学府中建设一个世界一流的人工智能学院是一个全新的挑战,是我们需要勇闯的无人区。
首先,我们要在学院体制机制建设上开拓创新。在学院筹备之初,学校领导和张磊校友就提出了"高起点、高水平、国际性、创新性"的发展方针,希望我们大胆探索学院学科建设的一流创新体制机制。高瓴人工智
分词→
文/继/荣/勇闯/无人区/尊敬/的/潘/云/鹤/院士/怀/进/鹏/书记/杜/占/元/局长/李/萌/副部长/高/润/生/副司长/黄/铁军/院长/刘/伟/校长/张/磊/校友/各位/嘉宾/老师/们/同学们/春/水/初生/春/林/日盛/春晖/方/暖/今天/我们/相聚/中国人民大学/共同/见证/高/瓴/人工智能/学院/的/成立/感谢/学校/和/社会各界/在/学院/筹备/过程/中/的/大力支持/习近平/总书记/在/中央政治局/第九次/集体/学习/中/指出/要/勇闯/人工智能/科技前沿/的/无人区/努力/在/人工智能/发展/方向/和/理论/方法/工具/系统/等/方面/取得/变革性/颠覆性/突破/今年/月份/春/节后/我们/向/潘/云/鹤/院士/汇/报高/瓴/人工智能/学院/筹备/工作/的/时候/他/也/向/我们/提出/了/学院/建设/要/勇闯/无人区/的/期望/我们/的/愿/景/是/打造/一/个/能够/影响/和/塑造/未来/人工智能/时代/的/世界/一流/学院/为/中国/和/全球/思考/并/创造/智能/而有/温度/的/未来/在/人民大学/这样/一/所以/人文/社会科学/见长/的/著名/学府/中/建设/一个/世界/一流/的/人工智能/学院/是/一/个/全新/的/挑战/是/我们/需要/勇闯/的/无人区/首先/我们/要/在/学院/体制/机制/建设/上/开拓创新/在/学院/筹备/之/初/学校/领导/和/张/磊/校友/就/提出/了/高起点/高水平/国际性/创新性/的/发展/方针/希望/我们/大胆/探索/学院/学科建设/的/一流/创新/体制/机制/高/瓴/人工智能/学院/得/积极探索/新/的/科研/和/人才/评价/制度/不拘一格/选拔/和/培养/青年/人才/学院/得/借鉴/国际/名校/规范/创新/人才/管理机制/教师/序列/得/全面/实行/制/学院/得/为/教师/打造/国际/一流/的/工作/生活/环境/提供/与/国际/知名/大学/和/研究/机构/接轨/的/学术/平台/薪酬/水平/住房条件/子女教
中文分词
窦志成:从微软到人大
从企业转到高校,就像过去人生中的许多重要转折点一样,窦志成并不是刻意追求,而是顺其自然。但他对凡事均竭尽全力,总能在看似被动的经历中取得佳绩:在过去的六年里,他先是担任人大信息学院特别研究员,随后被破格晋升为教授,如今又被任命为人大高瓴人工智能学院副院长。
事实上,作为一路拔尖的学霸,从小学到初中、高中,甚至是大学,窦志成的学习成绩一直名列前茅,稳扎稳打。但高考后,窦志成没有选择报考清北,而是去了南开读计算机。
当时,学校里负责为学生提供报考意见的老师跟他说,前一年的承德市状元去了南开大学读计算机,体验非常好。
那时候,计算机刚刚起步,周边人一致认为这个专业很有前途。作为理科生,窦志成的内心对前沿技术自然十分向往,于是便报了南开大学信息技术科学学院计算机专业。
结果,窦志成去了南开后才发现:"那个师兄报的不是计算机,而是国贸专业。"真是误打误撞。
更让窦志成没有想到的是,因为一个"信息不对称",他将在南开度过最宝贵的九年时光:本科、保研、转博,成为一名中国"土生土长"的计算机博士,重点研究领域还是信息检索。
分词→
窦/志成/从/微软/到/人大/从/企业/转到/高校/就/像/过去/人生/中/的/许多/重要/转折点/一样/窦/志成/并/不是/刻意追求/而是/顺其自然/但/他/对/凡事/均/竭尽全力/总能/在/看似/被动/的/经历/中/取得/佳绩/在/过去/的/六年/里/他/先是/担任/人大/信息/学院/特别/研究员/随后/被/破格/晋升为/教授/如今/又/被/任命/为/人大/高/瓴/人工智能/学院/副/院长/事实上/作为/一路/拔尖/的/学/霸/从小/学到/初中/高中/甚至/是/大学/窦/志成/的/学习成绩/一直/名列前茅/稳扎稳打/但/高考/后/窦/志成/没有/选择/报考/清/北/而是/去/了/南开/读/计算机/当时/学校/里/负责/为/学生/提供/报考/意见/的/老师/跟/他/说/前/一年/的/承德市/状元/去/了/南开大学/读/计算机/体验/非常/好/那/时候/计算机/刚刚/起步/周边/人/一致/认为/这/个/专业/很/有/前途/作为/理科生/窦/志成/的/内心/对/前沿技术/自然/十分/向往/于/是/便/报/了/南开大学/信息技术/科学/学院/计算机专业/结果/窦/志成/去/了/南开/后/才/发现/那/个/师兄/报/的/不是/计算机/而是/国贸/专业/真是/误打误撞/更/让/窦/志成/没有/想到/的/是/因为/一/个/信息/不/对称/他/将/在/南开/度过/最/宝贵/的/九年/时光/本科/保/研/转/博/成为/一/名/中国/土生土长/的/计算机/博士/重点/研究/领域/还是/信息检索/
中文分词
张磊:选择做时间的朋友
希望大家都能坚持自己内心的选择,不骄不躁,好故事都是来自于有挑战的生活;持之以恒,时间终将会成为你的朋友。在这里与同学们共勉。
除了选择的几个问题。作为人大的校友,我觉得还有一点很重要,那就是我们不仅要掌握科学思辨的能力,还要心中长存人文精神的火种。当今时代,伴随基因技术、机器人和人工智能技术的发展,科技爆炸、奇点临近,人类将进入新纪元,我们的生活也会迎来巨大的挑战。而大家在人大的学习生活,恰恰赋予了我们广博的视野和人文情怀,这将帮助我们处乱不惊,面对纷繁复杂的世界,不断去追问问题的本质。我本科是学国际金融的,没有编过程也没有技术背景,但是我后来照样投资了一批科技企业,它们现在在各自领域内引领世界潮流。我感谢咱们人大的人文教育,相信同学们也会从中获益无穷。
作为投资人,我常说起我的三个投资哲学:"守正用奇"、"弱水三千,但取一瓢"、"栽李不言、下自成蹊",这三句话分别源自《道德经》、《论语》、《史记》。虽然现代金融投资的工具和方法大多源于西方,但如何使用好这些工具,我还是更推崇我们优秀的中国哲学思想和传统民族文化。我们要有充分的文化自信,要珍惜人大给我们的人文土壤,好好汲取营养。过去未去,未来已来。我希望我们人大学子,以后不管进入哪个行业从事什么工作,都能保持乐观和激情,用人文的情怀去雕琢自己,美化身边。赠人玫瑰,手有余香。
分词→
张/磊/选择/做/时间/的/朋友/希望/大家/都/能/坚持/自己/内心/的/选择/不骄不躁/好/故事/都/是/来自/于/有/挑战/的/生活/持之以恒/时间/终将/会/成为/你/的/朋友/在/这里/与/同学们/共勉/除了/选择/的/几个/问题/作为/人大/的/校友/我/觉得/还有/一点/很/重要/那/就是/我们/不仅/要/掌握/科学/思辨/的/能力/还要/心/中/长存/人文精神/的/火种/当今/时代/伴随/基因/技术/机器人/和/人工智能/技术/的/发展/科技/爆炸/奇点/临近/人类/将/进入/新纪元/我们/的/生活/也/会/迎来/巨大/的/挑战/而/大家/在/人大/的/学习/生活/恰恰/赋予/了/我们/广博/的/视野/和/人文/情怀/这/将/帮助/我们/处乱不惊/面对/纷繁复杂/的/世界/不/断/去/追问/问题/的/本质/我/本科/是/学/国际金融/的/没有/编/过程/也/没有/技术/背景/但/是/我/后来/照样/投资/了/一批/科技/企业/它们/现在/在/各自/领域/内/引领/世界潮流/我/感谢/咱们/人大/的/人文/教育/相信/同学们/也/会/从/中/获益/无穷/作为/投资人/我/常/说起/我/的/三个/投资/哲学/守/正用/奇/弱水三千/但/取/一瓢/栽李不言/下自成蹊/这/三句话/分别/源自/道德经/论语/史记/虽然/现代/金融/投资/的/工具/和/方法/大多/源于/西方/但/如何/使用/好/这些/工具/我/还是/更/推崇/我们/优秀/的/中国/哲学思想/和/传统/民族/文化/我们/要/有/充分/的/文化/自信/要/珍惜/人大/给/我们/的/人文/土壤/好好/汲取/营养/过去/未去/未来/已/来/我/希望/我们/人大/学子/以后/不管/进入/哪个/行业/从事/什么/工作/都/能/保持/乐观/和/激情/用/人文/的/情怀/去/雕琢/自己/美化/身边/赠/人/玫瑰/手有余香/