

《数据结构与算法（1）》课程实验

基于文本内容的电影检索与推荐 (第二部分)

教师：张力老师

助教：李璇 罗剑桥

2019 年 11 月 26 日

一、 实验目的

本次实验通过实现基于文本内容的电影检索与推荐系统,可以对电影网页进行信息提取并分词,以此为基础构建倒排文档索引,实现电影查询及简单推荐的功能。

实验中需要实现的算法和功能有:分析网页结构并提取信息、中文分词、索引机制的实现、倒排文档及查询系统的构建、推荐系统的简单实现。希望通过常用的数据结构与算法进行训练,锻炼同学们的实际编程能力。

二、 实验环境

开发环境 (建议)

- 操作系统: Windows 10
- IDE: Visual Studio
- 编程语言: C++

测试环境 (检查标准)

- Windows 10 专业版 64 位
- CPU：Intel® Core(TM) i7-7Y75 CPU @ 1.30GHz 1.60GHz
- 内存：16.00GB
- IDE：Visual Studio 2015

三、 评分方案

首先，遵守学术职业素养的基本原则：如果在提交的实验结果中发现相互抄袭现象，被抄袭和抄袭者的本次实验分数均为 0 分。如果发现使用第三方代码的情况，若未直接注明出处，则视为抄袭，抄袭者的本次实验分数为 0 分，若注明了，则根据使用情况酌情考虑扣分。

实验评分将依照两部分进行：系统运行结果、系统实现内容。

系统运行结果是指助教正确运行提交的可执行文件，并根据运行结果进行评分。评分标准包括系统的是否可执行，输出结果是否正确，系统的效率等；

系统实现内容是指代码是否实现了要求的数据结构与算法，助教将会检查实验报告及源码实现进行综合给分。

具体的实验评分项将在实验内容中说明。

实验中鼓励创新，在完成基础任务的情况下，任何与实验相关的、有意义的创新都将有机会获得额外加分。加分项上不封顶，但与基础得分的总分不超过 110 分（基础满分 100 分）。

四、 实验提交

最终实验要求提交 3 部分内容，请参考以下说明按照文件夹进行组织。

在实验材料中，包含一个提交样例目录[提交样例]。提交作业时请根据样例目录中文件夹[2015311966_李璇_实验 2]的组织 and 命名格式，在其子目录下放置对应内容。

1. 源代码 放置 VS 项目工程, 务必删除.sdf 等大文件和编译产生结果文件。
2. 可执行文件：放置可以直接运行的可执行文件，该目录下应该同时包含 readme 说明文件及相关配置和输入文件。
3. 实验报告：pdf 格式，**不超过 4 页**，正文使用宋体小四号字，单倍行距；
实验报告中要求提供包括但不限于以下信息：实验目标、实验环境、抽象数据结构说明、算法说明、实验流程、操作说明、实验结果、功能亮点、实验体会；言简意赅阐述清楚即可，不要复制代码或截图代码。鼓励图文并茂辅助说明，但注意引用图片的版权。

注：实验一部分同学未按照要求格式提交的作业，请务必按照要求格式提交作业，否则将按照规定酌情扣分！

五、 实验内容

本学期实验由两部分组成。

实验 1 电影信息提取与分词

实现基础数据结构，通过对网页的解析，实现电影信息的提取与文本分词。

实验 2 电影检索与推荐

在实验 1 的基础上进行，通过实验 1 预留的接口，以上千网页作为数据库，实现根据输入关键字在数据库中检索电影的功能，并能够针对特定电影根据不同规则进行推荐。

5.1 实验目标

本次实验是整个课程实验的第二部分，目标是在豆瓣电影数据库中，利用倒排文档对电影数据库进行组织，并对倒排文档结构使用索引(使用平衡二叉树)，从而实现针对给定输入关键词的电影检索与推荐。

具体来讲，首先可以根据给定的 1083 个网页文件，使用实验 1 中的接口完成电影信息提取和分词操作，并使用电影信息和分词结果构建倒排文档，使用平衡二叉树完成倒排文档的组织 and 索引；然后，在给定关键词的情况下，使用倒排文档完成电影的检索功能；最后，指定特定的电影网页后，可根据该电影网页的信息，返回长度为 5 的推荐电影列表。

最终程序能够完成查询关键字返回电影网页名称、根据特定电影网页推荐相关电影的功能。鼓励开发用户界面，实现用户友好的操作方式。

6.2 数据结构及算法要求

实验中设计到的数据结构与算法有：

- 数据结构：平衡二叉树（必做）、哈希表（可选）、文档链表；
- 算法：查询算法、推荐算法；

本次实验中，要求同学们实现两种数据结构：平衡二叉树、文档链表，另外可自行实现哈希表。每项数据结构需要实现的基本操作如下：

数据结构	函数名称	函数功能
平衡二叉树	Insert	添加节点
	Search	查找节点
	Adjust	调整二叉树使其平衡
	Remove	移除某节点（可选）
	Edit	编辑某节点（可选）
文档链表	Add	添加文档
	Search	搜索文档
	Edit	修改文档
	Remove	删除某文档

实验评分过程中，将严格参照上述数据结构的功能进行评分。建议每种数据结构单独创建一个类，在类中实现上述函数，并添加相关注释。

在之后的功能实现中，除特殊声明的步骤外，所使用的数据结构必须是以上自行实现的数据结构类型。

注：在执行文件读写等最基本的操作时，可以使用 C++ 自带的字符串类型进行读写，但不能使用与其相关的系统函数对其进行其它变化操作。例如，读取完成后，应立刻使用自定义的字符串数据结构对文件内容进行保存，此后的操作都在自定义字符串数据结构上完成。

完成基础数据结构的实现后，可进行本次实验任务的开发。本次实验要求实现的功能如下：

1. 词典索引机制：要求使用平衡二叉树（必做）或哈希表（选做）对词典进行索引，即通过单词能快速获取单词在词典中对应的节点。
2. 文档链表更新：单词对应的文档链表在遍历数据库的过程中不断更新，根据单词出现次数保持排序。
3. 倒排文档查询：综合上述两点，以词典和文档链表为基础构建倒排文档。
4. 关键词查询：输入多个关键词，能够根据倒排文档快速返回搜索结果。
5. 电影推荐：输入电影名称，自定义规则返回推荐电影列表。
6. 图形化界面，用户友好的操作模式。

如果有时间，可以尝试使用哈希表替换平衡二叉树对词典进行组织，并比较两种索引机制在建立索引、查询效率等方面的差异，可适当加分。

注：除特殊声明的相关实验步骤外，以上数据结构和算法需要自行实现。

倒排文档的相关格式参考可参见附录 1。

6.3 测试方案

实验完成时，助教需要能够使用提交的可执行文件直接获取实验结果。

1. **批量搜索：能够根据查询文件，得到结果文件。**

查询文件的格式为：每一行是一个查询，关键词之间使用空格分开。比如查询文件如下：

```
1 客轮 冰山
2 戏剧 爱恨情仇
3 |
```

查询结果文件保存格式为：每一行为对应的查询结果，使用 (urlID, occurTimes), (英文空格)，并用空格隔开多个查询结果；其中 urlID 表示对应电影序号，occurTimes 表示多个关键字出现的总次数。例如（仅示例，由于页面文件编号未定，最终结果不一定与下图相同）：

```
1 (21,6) (36,3) (57,2)
2 (33,7) (78,4) (132,2) (111,1)
3 |
```

注意：

- 1) 查询文件与 exe 目录同级，命名为“query1.txt”；
- 2) 结果文件保存在与 exe 同级目录下，命名为“result1.txt”；
- 3) 在执行查询之前，首先需要对输入的关键词进行分词操作；
- 4) 查询的逻辑上，需要返回出现任意关键词的文档。即多个查询词之间的关键是逻辑“或”的关系，但同时出现多个关键词文档的排序应该靠前。（此处将根据检索效果评分）

2. 批量搜索：能够根据推荐查询文件，得到推荐结果文件。

其中查询文件的格式为：每一行为一个查询，为电影名。比如推荐查询文件如下：

```
1 海角七号
2 辛德勒的名单
3 |
```

推荐结果文件保存格式为：每一行为对应的推荐结果，使用 (docID,

newsName) 表示, 并用逗号隔开多个查询结果 ;其中 docID 表示对应的文件号, newsName 表示电影标题。 如果输入电影名称不在数据库中, 输出相关信息。

例如 (仅示例, 最终结果与下图不同) :

```
1 (20, 罗马假日) (59, 志明与春娇) (390, 两小无猜), (239, 甜蜜蜜), (10, 不能说的秘密)
2 (641, 拯救大兵杰瑞) (39, 窃听风暴), (87, 飞越疯人院), (90, 鬼子来了) (59, 美丽人生)
3 该电影不在数据库中, 无法推荐
4 |
```

注意 :

- 1) 推荐查询文件与 exe 目录同级, 命名为“query2.txt” ;
- 2) 结果文件保存在与 exe 同级目录下, 命名为“result2.txt” ;

3. GUI 交互界面

在界面中, 要能够输入关键词 (1 或多个) 进行查询, 返回查询结果及相应的推荐电影。

输入和输出方式由程序自定义, 请务必说明操作方法。

注 : 如果没有说明程序的使用方法, 导致助教不知如何使用程序进行功能测试 ; 或者助教在使用程序的过程中始终发生崩溃无法继续, 将酌情扣分。

注 : 建议至少完成基本的界面操作, 方便选择不同功能 ; 同时, 搜索返回结果能够通过点击等方式获取详细电影信息, 并在显示电影信息的同时给出搜索关键词所在的位置, 以及针对该电影的推荐结果。

测试方法 :

为确保助教能够顺利对实验进行测试, 请务必注意以下几点 :

1. 助教只负责在可执行程序的同级目录下, 放置 query1.txt、query2.txt 文件。

(请同学们将输入数据库 input 文件夹自行放置在相关位置。)

2.在可执行程序的同级目录下，自定义放置所需要的配置文件、词库等其它所需输入数据。放置路径可自定义，但确保使用的是可执行程序的相对路径，以保证可移植性。（即，项目移植到另外一台电脑上，也可以顺利执行。）

3.在可执行程序目录下，有 query.exe 和 gui.exe 两份可执行程序，能够分别实现上述两个功能测试的内容，执行 query.exe 后要求在同级目录输出 result1.txt 表示批量查询结果，result2.txt 表示批量推荐结果。

测试时，助教将使用脚本执行 query.exe，并读取 result1.txt、result2.txt 的结果，根据查询结果及运行效率进行打分。另外，助教将手动执行 gui.exe,根据程序的实现功能及用户体验进行评分。

因此，请大家务必按照上述要求组织提交内容。

注：如果助教无法通过上述配置获取实验结果，最终所得分数将扣除 30%。

6.4 评分细则

助教将根据以下模块对实验进行评分。

模块	内容	分数
数据结构	词典（平衡二叉树）	20%
	倒排文档及建立	20%
功能	批量检索功能	15%
	批量推荐功能	5%
	用户交互搜索与推荐	12%

	效率	8%
	用户友好性	5%
文档与代码风格	相关文档	10%
	代码风格与注释	5%
*亮点与加分项	相关特色功能点	10%

注：与实验 1 相同，**如果程序无法执行，将酌情扣分**。另外，需要提醒一下，由于数据量较大，包括数据库文本数量、词表内的词的数量等。所以同学们需要仔细研究，找出不需要的存储和计算开销，进行相应的优化。

亮点与加分项需要在文档中说明，加分将会根据实现的亮点进行评判。

六、 其它事项

实验报告：

除了代码工程之外，**实验报告是体现你工作量的重要工具，也是助教进行实验评分的主要依据之一**。请同学们合理分配写代码和实验报告的时间，实验报告以简洁清晰为主。

代码注释：

在实际工程开发中，代码注释非常重要。在此不给同学们规定哪里一定要写注释，但希望同学们在关键的变量、方法、算法步骤处使用注释进行简单说明，帮助他人（很可能是几年以后的你自己）理解代码的功能。

作业迟交：

作业若未能按时在网络学堂上提交，可通过邮件或其他方式提交给助教。迟交的时间点按照助教确认为准。若出现迟交作业，需要在作业评分的基础上扣除相应分数，按照迟交的天数，扣分依次为 5%、15%、30%、50%、70%、100%。迟交天数按照向上取整计算。

其它未尽事宜，将在网络学堂上补充通知，谢谢。

附录 1：倒排文档格式说明

倒排文档源于实际应用中需要根据属性的值来查找记录。具体到课程实验内容中，即根据某个“单词”查找出现过该单词的所有“文章”。

在实现倒排文档的过程中，需要解决的问题主要有两个，首先是如何在海量数据库中定位到当前单词所在的位置，其次是找到该单词后如何获取搜索结果

这就将我们需要实现功能划分为两部分，倒排索引和文档链表。

倒排索引部分要求使用平衡二叉树实现，即给定一个单词 A，能够通过索引结构快速查找到单词 A 对应的节点。节点包含了这个单词的相关信息。相关信息可以参考如下：

Term (String)	TermID(int)	DF(int)	Occur(int)
单词	单词 ID	单词出现在多少篇文章中	单词总的出现次数

另外，节点还应该包含其关联的文档链表的头节点。在文档链表中，至少需要的存储信息如下：

TermID	DocID(int)	Times(int)	DocID(int)	Times(int)
单词 ID	出现该单词的文档 ID	单词的出现次数	出现该单词的文档 ID	单词的出现次数

其中，词典中每个单词都有一个文档链表，文档链表的节点中包含出现该单词的文档 ID 和词语在相应文档中的出现次数。

上述文档链表的结构可自行定义，只要包含相关信息并实现对应功能即可。