哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

自然语言处理

实验一：汉语分词系统

School of Computer Science and Technology

Harbin Institute of Technology

1 实验目标

本次实验目的是对汉语自动分词技术有一个全面的了解，包括从词典的建立、分词算法的实现、性能评价和优化等环节。本次实验所要用到的知识如下：

- 基本编程能力（文件处理、数据统计等）
- 相关的查找算法及数据结构实现能力
- 语料库相关知识
- 正反向最大匹配分词算法
- N 元语言模型相关知识
- 分词性能评价常用指标

2 实验环境

编程语言：C/C++、python、或者 Java(任选)

其他无特殊要求

3 实验内容及要求

训练集：199801_seg&pos.txt（1998 年 1 月《人民日报》的分词语料库）

2021 年加发 1 倍的训练数据；

人名地名资源

最终测试集：格式参见 199801_sent.txt，数据来自多来源文本。

注意：因数据版权约定，同学不可将所有接收到的数据透露给任何其他同学和单位，局部可用于任何非学习目的。原则上应在完成实验后，删除所接收到的全部训练数据。

3.1 词典的构建

输入文件：199801_seg&pos.txt

输出：dic.txt（自己形成的分词词典）

提交要求：1）dic.txt；

2）实验报告：须说明分词单位的标准、以及词典文件格式说明；

须对自己所构建的词典进行分析；

{提示：所提取词典没有要求一定写代码完成☺；对词典的分析要从实用的角度进行分析，分析手段得当，结果有价值}

3.2 正反向最大匹配分词实现

输入文件：199801_sent.txt（1998 年 1 月《人民日报》语料，未分词）

dic.txt(自己形成的分词词典)

输出：seg_FMM.txt 和 seg_BMM.txt(正反向最大匹配分词结果，格式参照分词语料 “词/_词/_.....”)*这里的_代表空格

编程要求：

- 自己定义词典的数据结构，并书写词典查找算法。不允许使用类似 list， dict (python 特例允许使用 list)等编程语言内置的数据结构
- 鼓励最少代码量的系统实现

提交要求：1）seg_FMM.txt 和 seg_BMM.txt；

2）程序源代码；

3）实验报告：须说明程序实现过程中的收获；

{提示：写最少的代码☺}

3.3 正反向最大匹配分词效果分析

输入文件：199801_seg&pos.txt（1998 年 1 月《人民日报》的分词语料库）

seg_FMM.txt、seg_BMM.txt

输出：score.txt(包括准确率（precision)、召回率（recall)，F 值的结果文件)

编程要求：

- 自己编写评价代码
- 保证评价结果的正确性

提交要求：1）score.txt；

2）评价结果的误差,将影响本次实验最终成绩(例如,在精确率指标上,自己计算结果为0.96,最终核查结果为0.97,|0.96-0.97|*100=1,则本次实验成绩最终得分将被扣除 1 分。这里的误差包括“精确率误差+召回率误差”，不再考虑 F 值的误差)；

3）实验报告：须分析正反向对大匹配在分词精度上的差异,分析角度独特有加分（最终实验成绩上最多加 3 分)；

{提示：看似简单，但是很多同学修改这段代码的时间比初次完成的时间要长---如果自己写，不用内置的函数。祝早日通过☺}

{针对实验课中提出的问题，补充说明：1}注意实验输入输出的要求；2）本小节不检查代

码，不用提交评价工具；3)特别提示：如果采用本小节输出的 score.txt 来讨论分词性能，是很不严谨的，将被扣除本小节 50%的得分}

3.4 基于机械匹配的分词系统的速度优化

输入文件：199801_sent.txt（1998 年 1 月《人民日报》语料，未分词）

输出：TimeCost.txt（分词所用时间）

编程要求：

- 任选前后向最大匹配分词算法其中之一，尽可能对分词系统速度优化，最低要求实现二分查找；
- 禁止使用开发环境内置的数据结构，查找算法和数据结构都要求独立实现；
- 程序初始化时间不考虑在内，仅计算从分词过程开始到分词结果输出完成的耗时

提交要求：1）TimeCost.txt（应包含优化前后的分词耗时）；

2）程序源代码；

3）实验报告：须详细描述所实现的优化方案，分析优化技术的效果，尝试揭示分词速度进一步优化的关键；

{提示：挑战索引结构，比如哈希什么的（找到恰当的哈希函数不太容易）；有同学直接手写了双 Trie 树结构，很惊艳；另外，这里速度相对提升有底线要求☺}

3.5 基于统计语言模型的分词系统实现

输入文件：test.txt（未分词的最终测试集，多种来源）

训练数据：199801_seg&pos.txt（注：2021 年会加发 1 倍的数据）

输出：seg_LM.txt（利用统计语言模型分词结果，格式参照分词语料）

编程要求：

- 根据 199801_seg&pos.txt 建立随后需要使用的统计语言模型；
- 使用动态规划，实现全切分有向图的搜索；
- 至少使用一元语言模型（最大词频分词）
- 鼓励实现基于二元语言模型的分词系统；
- 鼓励实现未登录词识别；

提交要求：1）能够读入指定的测试文件，输出文件为 seg_LM.txt；

2）分词程序涉及的全部源代码（及第三方工具，如有*）；

3）实验报告：须对程序中的重点实现代码进行说明（可用流程图对算法进行辅助说明）；对比分析各种不同分词方法的性能；

*对于第三方工具的的要求，依据实验课上的解释和要求;备忘提示；仅仅调用类似结巴分词工具的做法，是违规的；

{提示：一元文法挺有效。二元文法难在参数平滑，程序实现也更复杂；最大的福利：所有编程的限制取消，编程语言的内置函数、库，放开使用☺}

3.6 其实可以没有这一节，只不过刷榜的同学多了，再给出一些说明：

在上述工作全部完成，并入数提交的基础上，可以放开限制，使用任何您能驾驭的方法模型，只要是能证明是自己的方法（关键的环节是自己的实现），而不是纯使用第三方工具（调API、封装别人的接口）。

当然，训练数据不允许超出已经给定的数据，第三方词典什么的，基本不符合这一原则。

针对人命地名识别，会有一些针对性的数据集提供。

4 实验报告

不要流水账；

按照 ACL 论文的内容安排撰写，凝练自己工作的核心（发现、贡献），巧妙的讲上述实验结果，自己的设计、心得，写出来。

按照 ACL 会议排版要求，网上有模板。

正文部分不允许出现源代码，在说明问题时可使用伪代码（如需附代码，请使用附录）

请确保实验报告格式清晰、一致，内容的条理性和完整性

5 提交方式

中期验收：11.27,23:59’59”(完成并提交到 3.5 节之前的实验工作)

最终截止日期：12.5,23:59’59”

提交方式：提供所要求的代码和报告,可以通过 QQ 发送,或者网盘云文件的形式。

由助教确认收取。

6 评分方式

1）该实验成绩=编程实现成绩+报告成绩

2）编程实现成绩:12 分

6 分：3.3 完成，个人独立完成；

7 分：3.4 完成，个人独立完成；

8 分及以上：3.5 完成，小组成员不超过 3 人，根据完成度和贡献度确定分数；

完成度评分：正确完成动态规划，以 1 元语言模型输出结果，评分 8；

在上述基础上，以 2 元语言模型数据结果，评分 9；

在上述基础上，正确进行了未登录词识别，并采取了进一步的性能优化手段，以最高性能记为 12 分，其余根据性能差异，按比例取得；

贡献度评分：小组内每人预分配 3 分，根据组内贡献度，最终决定每人得分；

要求每人贡献度得分不能相同，分数总和等于 3*n（n 为小组人数）；

{特别提示：本实验不是课外竞赛，要求提交的代码中，必须包含 uni-gram 或者 bi-gram 分词结果，否则评分不超过 8 分；在最终的性能冲刺中，单独调用某个的第三方分词 api、某个第三方分词工具，是不被认可的；}

3）报告成绩:5 分

内容完整

格式规范

包含所使用的参考文献[重要]

参考 ACL 会议论文模板，内容安排和呈现方式越贴近 ACL 要求，得分越高

没有一定的页数限制，建议排版后正文在 4-8 页之间。

**最后的备注：独立完成，意味着所有应提交的代码和报告都应分别有物理存在，并按时提交（从未想过要在实验要求中写下这句话，但是出现了个别多人小组提交了一份代码，认为符合独立完成的要求……）