人工智能技术实验报告

实验 1、拼音输入法

姓 名: 孟 念
学 号: 1800092850
班 号: 40240902-0

人工智能技术
(春季, 2021)

清华大学
计算机学院
马少平老师

2021 年 4 月 2 日

目 录

1 基本要求

使用基于字的二元模型，实现一个拼音到汉字的转换程序。
完成实验报告，主要内容如下:

- 介绍算法的基本思路和实现过程
- 展示实验效果，选取效果好和差的例子进行分析
- 对比参数选择，进行性能分析
- 总结收获，提出改进方案

2 实验代码

2.1 马尔可夫模型训练

- **目标:** 获取以汉字开头的序列的可能性: **start probabilities**.
  获取从拼音 (观察) 到汉字 (隐含状态) 的可能性: **emission probabilities**.
  获取一个汉子被另一个汉子跟踪的可能性: **transition probabilities**.
- **实现:** 首先，我从文档中提取了汉字。因为没有提供拼音，我需要使用外部工具提取拼音。这是因为在解码过程中，我们需要知道一个拼音能代表哪些汉字。我决定使用叫 pypinyin 的一个 PythonPackage。然后，我简单地统计了每个汉字的出现次数。最后，我将计数更改为概率。

2.2 Viterbi 解码

- **目标:** 根据一个拼音序列，用计算效率高的方式找到最可能的汉字序列。
- **实现:** 对于输入序列中的每个拼音，我首先检索出各自的概率。对于每个可能的汉字，我计算其概率。在动态字典中，我一直保存当前和上一个状态的分数。去下一个状态时，程序就会删除状态两个字段之前的分数。最后，得出得分最高的隐含状态序列。

2.3 经验教训

请在这一节详细说明需要分析的内容

- **标准化:** 最初，我试验了直接使用计数，效果还不错，但并不完美。在切换概率时，将 emission 与 transition 概率用同样的标准化很重要。否则，当它们相互相乘时，其中一个可能会有更大的权重。
- **计算复杂性:** 最初，解码的时候我保存了所有可能的隐含状态序列，其复杂性由 O(k^{n}) 增长。k 是一个观察的可能隐藏状态（就是给定拼音的可能汉字）。我计算出每个拼音的平均可能汉字量为 17.65。**n** 是观察序列的长度（几个拼音）。比如 qing hua da xue ji suan ji xi 这样的一个序列，复杂性将为 O(17.65^8)。这时我才意识到 Viterbi 的重要性。通过只在内存中保存两个状态，我们将其降低到 O(nk^2)。序列的复杂性将为 O($8 * 17.65^2$)。Viterbi 只保存最可能的序列，我们只需要在内存中保存两个观察的可能状态概率。

3 实验结果

3.1 结果分析

下面请看结果和分析。

表 1: 实验结果

方向	数据	描述
输入	qing hua da xue ji suan ji xi ren gong zhi neng ji qi xue xi shu ju wa jue	为下面的实验输入。
结果	顷骅达靴鸡酸鸡系 钰龚酯能 鸡齐靴🄴 鼠飓袜决	标准化的错误。
结果	清华大学计算机系 人工智能 机器学习 数据挖掘	标准化做好了。这个结果只用最小的文档来训练 (20MB，2016-11.txt)
输入	si dong fei dong hai shi shen lou bu guan san qi er shi yi	为下面的实验输入。难度比较高。
结果	司董飞动 还是渗漏 不管三期而是一	汉字都不对。这个结果是用所有的文档来训练 (1000MB，*.txt)。可是，实际上说我也不会知道那些拼音的汉子。

3.2 改进思想

- **增长数据：** 用的数据只是 1000MB，所以像《似懂非懂》,《海市蜃楼》这样的词可能一次也没有出现在数据中。为了解决这个问题需要更多数据!
- **马尔可夫性质：** 马尔可夫性就是下个隐含状态仅依赖于当前的状态。这样我们的计算更容易，但是质量也变得更差。《海市蜃楼》的《楼》显然不仅要看《蜃》，还要看前面的字。解决这个问题有两种方案：第

一个就是看更多的过去的隐含状态。可以用一个二阶还是三阶的马尔可夫模型。第二种选择是改变模式。通过深度学习方法我们可以看每个状态模型。当然深度学习也会需要更多的计算量，但一般来说，它比马尔科夫模型更有效率。