拼音输入法更新文档

Z输入法, V 2.0

周昊一,张绍文,郑健,张昊,周佳雨

注:本文档是多人合作完成的,因此语言风格上会有一些差异,可能还有一些重复,其中一至五是一部分,六是一部分,七是一部分.组员学号,具体分工等在八中有说明.

# 更新内容

此次更新主要是涉及到语言模型方面.基于HMM模型训练并建立了一个语言模型,这个语言模型主要的用途是更好地实现简拼功能.同时,在语言模型的训练过程中,也获取了新的带有权重数据的字典与词典,进一步提升了输入过程中的准确率.

此外,还对原有的一些方面进行了一些优化.例如在音节切分方面,再加入了反向最大匹配,用于处理类似于”xian->xi’an”这样的情况.另外还对运行速度进行了一些优化,由于HMM模型的加入,原本性能会有显著下降,通过这些优化尽可能的将性能维持在原有水平.

# 基于语料库的新词库生成

## 语料库

本次所使用的语料库是2007~2011年5年的人民日报的数据.总计196051篇文章,892M的文本.

语料库是本次整个工作的基础,新的字库,词库,以及语言模型都是在它的基础上产生的.因此语料库的语言偏向性即代表了最后整个输入法的偏向性.

由于人民日报中的文章用词都比较正式,鲜有口语并且多涉及人名地名,因此最后可以看到生成的整个语言模型对于口语化的语句输入不能够很好的处理,但是对于比较正式的语句能够较为顺畅的输入.

## 新的字库与词库的挖掘与生成

在上一个版本中,所使用的字库与词库都是人工搜集或生成的.具体的讲,字库是我们在半手工地生成的完整字库,并对其中一些字的权重作了一定的人工标注.而词库则是从网上收集的高频词词库,由于词数太多,做人工标注不现实,并且也无法保证标注结果的有效性,因此没有做标注.

因此,在这一版本中,我们希望能通过语料库来得到一个更为完善,有数据支撑的词库.

我们的做法是,先对文本进行分词,再对分好词的文本进行统计,记录每个出现了的词的词频,同时也记录每一个词的词频.

接下来进行一次筛选,将词频过低的词去除,因为这些词频过低的词显然不是常见的词语.根据观察,通常低词频的词都是人名与地名,这都是我们所不希望出现在我们的核心词库中的.

随后我们根据词(单字也算作词)的词频f算出它的权重w.虽然可以直接令w=f,但是实际上高词频的词与低词频的词之间词频差距太大,在计算概率时会因为浮点数精度问题导致得到为0的概率,因此需要某种手段来缓和高频词和低频词之间的差距.我们所选用的方法是取对数,w=ln(f+1),之所以+1是为了保证w不为0..

最后,将得到的词库输出为我们所需要的格式.

# 基于HMM的简拼查找

# 第一版中的简拼

第一版中的简拼查询是基于简拼还原为全拼的思路来做的.将简拼还原为所有可能的全拼,再对这些全拼进行查找.

这种方法的主要问题是随着简写部分的增多,搜索空间是呈指数级增长的.原因是它将简拼还原为所有可能的全拼,但是这些全拼在词典中很有可能是不存在的.

对于全首字母简拼则是将这些全首字母全拼也全部加入到词典中,当作词语进行直接查询.

第一版中的简拼还有一个很大的问题就是无法对最终结果进行有效的排序,因为没有相关的信息可以参考,只能当作等概率处理.

# HMM模型的训练

首先我们描述一下输入法中所使用的HMM模型:

用户的输入表示一系列用户输入状态,例如(xin),(xing)就是两个不同的状态.每个用户输入状态(即拼音)对应一个中文字串.如假设在某次输入时(xin)对应的中文字串是(新),但在其他时候(xin)还可能对于(心),(信)等.

在这里表示的是可观测的外部状态,而表示的是不可观测的内部状态.不同的内部状态有可能对应相同的外部状态,不同的外部状态也可能对应相同的内部状态(比如简拼的情况).而我们的目标就是对一个给定的,要能得到一个内部状态的序列,其中内部状态是按照可能性的大小降序排列的.

对于全拼输入而言,这个内部状态集合是很容易得到的,直接查字典即可.

关键是对于简拼而言,这个内部状态集合是隐含的.因为在字典中无法得到简拼的相关信息.

但是我们可以根据字典中信息得到从简拼到全拼的转移概率.如对于简拼(mingt),对于全拼(mingtian)有一个转移概率p,对于(mingti)也有一个转移概率p’.而(mingtian)对于(明天)的概率若是q,则(mingt)对应到(明天)的概率就是pq.

根据这个计算方法,我们可以计算出一个简拼到所有可能的全拼的概率,并进一步推算出到内部状态的概率.

在实际的算法中,并不是从简拼到全拼进行推算,而是从所有已知的全拼推算出对应的所有简拼情况的转移概率.

由于我们的输入法并不是要算最可能的情况,而是要算出所有情况,并进行排序,因此只能用朴素算法算出完整的转移矩阵.

# HMM模型的使用

经过训练,我们得到的是一个完整的由简拼到全拼的转移矩阵.在使用它来对简拼进行查找时,只需要做一步的推测,由简拼到所有可能的全拼的转移概率,以及这些全拼对应的中文串及其概率算出这个简拼对应的所有中文串及其概率.

# 其他改进与优化

## 简拼的完整功能描述

## 目前的实现

# 测试

不同于英文输入法中文拼音输入法在输入识别时面临着切音，分词，选词等诸多问题，而基于规律或是基于统计的算法都不能穷尽汉语的所有可能，尤其是在本项目中基于词法分析的算法，在面对规则性不强的新词、网络词汇很有可能导致出现在输入之后机器识别出的结果与人的预期结果有着不一致的地方，基于统计规律的由于训练集的有限性也存在着不尽如人意的地方。这使得黑箱测试尤为重要。

在本次试验测试中，采用单元检测的方法，依次检测切音，分词，选词等各个环节。

在切音环节我们随机的从训练集（词库共计有8万个词汇）中抽取1%，对其进行无差别切音检测，比较切音结果与预期是否相同。在比较结果中我们发现形如“xian（西安）”即第二个音无声母，会被识别成“xian（先）”而“先”这个词在日常生活中又是切实存在的，那么我们可以强制要求再输入“西安”时加入切音符“xi’an”，在接下来的工作中则会结合词频，上下文Ngrammar概率在“xi’an”识别的基础上加强对于“xian”基于统计规律的多重识别切音，智能识别成“xian（先）”或“xi’an（西安）”；形如“ganga（尴尬）”即的二个音的声母或起始音节被误识别为上一个音的尾音，即会被识别成“gang’a”这是我们所不能接受的，需要结合统计规律来避免这种情况。

在分词环节如键入“dajiazaoshanghao（大家早上好）”我们期望被识别为“dajia|zaoshang|hao”三个词，而由于训练集的有限性，汉语的多义性（歧义性）很有可能被识别成期望外的结果。比如“shuiguoranhoushu（水果然后熟）”我们希望分成“shuiguo|ranhou|shu”而不是“shui|guoran|hou|shu（水|果然|后熟）”，而这些问题只有通过大量的黑箱测试，找出问题句段，并用这些句段补充训练集才能更好的分词。

选词部分目前是基于词频的统计排序尚未结合上写文，目前检测主要是检测词频的排序以及词频改变是否能及时更新。

# 小组情况

本小组共有5个人.

Leader为周昊一(MF1233055),负责核心代码,以及用户界面的编写.

其余4个人为:

张绍文(MF123304),负责字典,词典的搜集,完善,以及后续工作中语言模型的训练.

郑健(MF1233054),同上.

张昊(MG1233046),同上.

周佳雨(MG1233091),负责测试用例的编写,并对程序进行测试.