

信息检索·第四次作业

专业：软件工程 姓名：胥倩雯

学号：1511504 完成时间：2017.1.3

一、实验项目

垃圾邮件识别系统

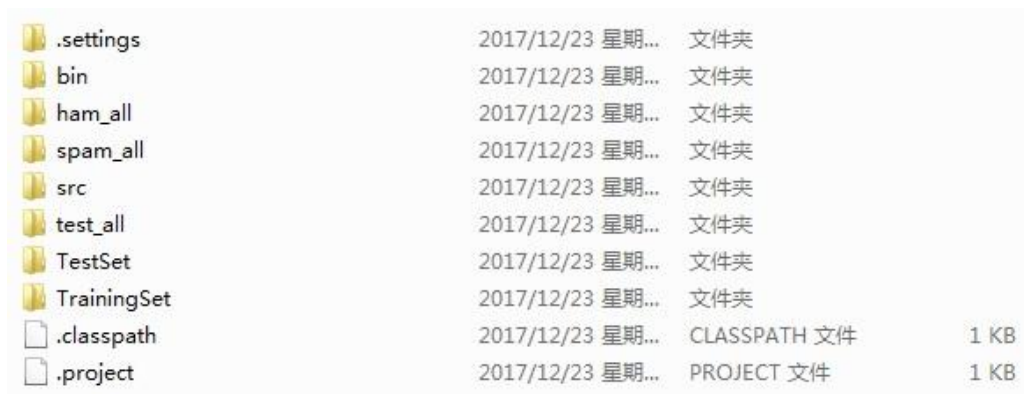
二、实验环境

Java (version 7) + eclipse

三、实现程度

1. 能够对中文和英文的垃圾邮件进行识别
2. 基于朴素贝叶斯
3. 就测试样例来讲，无论中文和英文，召回率均能够达到 83%以上，准确率能够在测试样例中高达 1
4. 若要提高召回率或准确率，可以增加训练样例，为了运行时间考虑，我的训练样例基数较小

四、实验效果



.settings	2017/12/23 星期...	文件夹	
bin	2017/12/23 星期...	文件夹	
ham_all	2017/12/23 星期...	文件夹	
spam_all	2017/12/23 星期...	文件夹	
src	2017/12/23 星期...	文件夹	
test_all	2017/12/23 星期...	文件夹	
TestSet	2017/12/23 星期...	文件夹	
TrainingSet	2017/12/23 星期...	文件夹	
.classpath	2017/12/23 星期...	CLASSPATH 文件	1 KB
.project	2017/12/23 星期...	PROJECT 文件	1 KB

源码\mailgabase 文件夹如上图所示。

其中，src 文件夹里存放了源码。

\TraningSet\SMSSpamCollection 存放英文垃圾邮件和非垃圾邮件的训练样例。具体内容如下图所示：

其中每一行都是一个训练样例。

每一行的第一个单词要么是 ham，要么是 spam。“ham”代表不是垃圾邮件，“spam”代表是垃圾邮件。

```

1 ham Go until jurong point, crazy.. Available only in bugis n great world la e buffet...
  Cine there got amore wat...
2 ham Ok lar... Joking wif u oni...
3 spam Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to
  87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
4 ham U dun say so early hor... U c already then say...
5 ham Nah I don't think he goes to usf, he lives around here though
6 spam FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some
  fun you up for it still? Tb ok! XxX std chgs to send, £1.50 to rcv
7 ham Even my brother is not like to speak with me. They treat me like aids patent.
8 ham As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set
  as your callertune for all Callers. Press *9 to copy your friends Callertune
9 spam WINNER!! As a valued network customer you have been selected to receive a £900
  prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
10 spam Had your mobile 11 months or more? U R entitled to Update to the latest colour
  mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
11 ham I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k?
  I've cried enough today.
12 spam SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575.
  Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
13 spam URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot! Txt
  the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18

```

\TestSet\TestFile.txt 文件里存放着 1574 个测试样例，其中有 213 个垃圾邮件。

```

1 ham K...k...when will you give treat?
2 spam This is the 2nd time we have tried to contact u. U have won the £400 prize. 2 claim
  is easy, just call 087104711148 NOW! Only 10p per minute. BT-national-rate
3 ham He's just gonna worry for nothing. And he won't give you money its no use.
4 ham Did you get any gift? This year i didnt get anything. So bad
5 ham somewhere out there beneath the pale moon light someone think in of u some where out
  there where dreams come true... goodnite & sweet dreams

```

与训练样例格式相同，我们会根据每一行除第一个单词以外的单词判断是否是垃圾邮件，并将判断结果与第一个单词进行校验。

```

该封邮件是垃圾邮件的概率为:0.9823845850780613,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9996785724785603,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999990899392326,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999999999998783,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999990899392326,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.999999977899705,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9976046142795282,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999999830036137,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.999954451143782,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999992279608075,实际是否为垃圾邮件:true
垃圾邮件总数213,正确识别了180封垃圾邮件,召回率0.8450704225352113,准确率:1.0

```

运行了我的程序，得到如上结果，召回率为 0.84 以上，准确率 1.0。

对于中文的垃圾邮件识别，用到了 IKAnalyzer 分词器。其中 ham_all 文件夹中存放非垃圾邮件的语料。

spam_all 文件夹中存放垃圾邮件语料。

test_all 中存放测试样例，其中 ham 文件夹中存放 600 个非垃圾邮件样例。spam 文件夹中存放 120 个垃圾邮件样例。

语料库我是下载的网上的，中文和英文的样例格式不相同。故对它们进行了部分不同的处理。英文垃圾邮件识别运行代码 myspamrecog.java，中文版运行代码 chinesesпам.java。

其实中文的语料库有 4.5 万个左右的训练样例，但是这样训练花费的时间太长啦。所以我仅保留了部分训练样例，里面有垃圾邮件，也有非垃圾邮件。由于训练的基数很小，所以可能会对精度造成一定影响。

我保留了 4372 个非垃圾邮件，3085 个垃圾邮件作为训练样例。

测试结果如下：

```
该封邮件是垃圾邮件的概率为:0.9999999999999983Pup: 1.3865136165875271E-80Pdown:2.243292214657772E-95,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 8.029472808412471E-51Pdown:2.5594442505242893E-73,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 3.1663842564795754E-81Pdown:1.3456586830450645E-112,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999999975335864Pup: 6.3165841329021616E-55Pdown:1.5579309120815837E-63,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 1.4195471262423096E-137Pdown:1.2316810766964127E-164,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 7.735649110319393E-47Pdown:1.3319538307862223E-71,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 3.038879474111013E-58Pdown:1.0896820767476295E-95,实际是否为垃圾邮件:true
垃圾邮件总数120,正确识别了115封垃圾邮件,召回率0.9583333333333334,准确率: 0.9829059829059829
```

召回率为 0.958，准确率为 0.983。

然后我修改了一个参数，对是垃圾邮件的判定更为严格。修改过后，测试结果：

```
该封邮件是垃圾邮件的概率为:1.0Pup: 8.029472808412471E-51Pdown:2.5594442505242893E-73,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 3.1663842564795754E-81Pdown:1.3456586830450645E-112,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:0.9999999975335864Pup: 6.3165841329021616E-55Pdown:1.5579309120815837E-63,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 1.4195471262423096E-137Pdown:1.2316810766964127E-164,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 7.735649110319393E-47Pdown:1.3319538307862223E-71,实际是否为垃圾邮件:true
该封邮件是垃圾邮件的概率为:1.0Pup: 3.038879474111013E-58Pdown:1.0896820767476295E-95,实际是否为垃圾邮件:true
垃圾邮件总数120,正确识别了112封垃圾邮件,召回率0.9333333333333333,准确率: 1.0
```

召回率为 0.933，准确率为 1。

可以看到，召回率下降了，准确率提升了。

我们发现用朴素贝叶斯来进行垃圾邮件识别还是较有效的，如果还想要提高召回率和准确率，可以增加训练样例。

五、实验分析

1. 首先，定义一个类，用于存储每个在测试文本中出现的单词的在几个垃圾邮件中出现，在几个非垃圾邮件中出现，以及一个邮件存在这个单词那么为垃圾邮件的可能性为多少。

```
class keyword
{
    public String keyword;
    public int inspam;
    public int inlegit;
    public double isspamprob;

    public keyword(String keyword, int inspam, int inlegit)
    {
        this.keyword = keyword;
        this.inspam = inspam;
        this.inlegit = inlegit;
    }
}
```

2. 接着，在 myspamrecog 主函数里，我们读取训练文件的每一个样例里的每一个单词，将每个单词及其对应的信息存储在 HashMap<String, keyword>类型的 keyMap 中，并把每个训练样例存储在 List<String>类型的 allmail 中。
3. 然后对每个测试样例是否是垃圾邮件进行判断，是的话，里面所有单词对应的 keyword 的 inspam 加一，否则 inlegit 加一。
4. spamnum 和 hamnum 分别代表训练文件中，垃圾邮件总数和非垃圾邮件总数。
5. 接着，我们通过下面的公式计算邮件中存在这个单词对应是垃圾邮件的条件概率（A 代表垃圾邮件，B 代表单词存在）

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})}$$

```
double Spam = 1.0 * kw.inspam / spamnum;
double Spamsun = 1.0 * spamnum / (spamnum + legitnum);
double Legit = 1.0 * kw.inlegit / legitnum;
double Legitsun = 1.0 * legitnum / (spamnum + legitnum);
kw.isspamprob=(Spam * Spamsun) / (Spam * Spamsun + Legit * Legitsun);
```

6. 以上，准备工作做好啦。接下来开始进行测试。
7. 对于每个测试样例，提取出里面在训练样例中存在的单词，对这些单词进行计算。计算公式如下：

$$P(A|T_1, \dots, T_n) = \frac{P(T_1, \dots, T_n|A)P(A)}{P(T_1, \dots, T_n)} = \frac{P(T_1|A)P(T_2|A) \dots P(T_{n-1}|A)P(T_n|A)P(A)}{P(T_1)P(T_2) \dots P(T_{n-1})P(T_n)}$$

```
for (String kw : oneMailKeyword)
{
    Pup = Pup * keyMap.get(kw).inspam / spamnum;
    Pdown = Pdown * (keyMap.get(kw).inspam + keyMap.get(kw).inlegit) / (spamnum + legitnum);
}
double Pmail = Pup / (Pup + Pdown);
```

对于每个单词，Pup 乘 $P(T_i|A)$ ，Pdown 乘 $P(T_i)$ 。

8. 因为 $P(T_i|A) / P(T_i)$ 是非常有可能大于 1 的，所以我们将分子加一份到分母，借以让最后的结果小于 1，易于判断。
9. 如果 Pmail 值大于 0.999，那么我们认为它是垃圾邮件，再与真实的是否是垃圾邮件比对。如果是，那么识别成功，如果不是垃圾邮件却被我们误认为是垃圾邮件，那么识别错误。

```

// 成功识别
if (Pmail > 0.999 && testmail.get(i).startsWith("spam"))
{
    rightnum++;
}
// 识别错误
if (Pmail > 0.999 && testmail.get(i).startsWith("ham"))
{
    errornum++;
}

```

中文垃圾邮件识别 chinesespam.java:

中文的垃圾邮件识别与较为相似，但语料库格式和英文不同，所以代码有些微的不同处理。

除此以外，中文文本读取时要经过一些处理，否则会变成乱码。

```
BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(filename), "utf-8"));
```

此外，我们还排除了一些无关的词项对判断是否是垃圾文档的影响，例如数字，以及发件日期等。

六、相关配置

直接在 eclipse 中引入 mailgabase 文件夹即可，我在 Java 文件中用得是相对路径，英文和中文的训练样例和测试样例都在 mailgabase 文件夹里。

七、总结

这次实验，我知道了其实我们所学的知识能够帮我们实现许多原以为很复杂的应用。若是想要提高准确率和召回率，我们可以增大训练集数据量，但是数据量越大，计算花费的时间就越长。在此次实验中，对于每一个样例，我只关心了关键词是否出现，并未考虑关键词出现的频率和样例的长度，如果把这些元素都添加进来的话，实现的效果可能会更好。其次，我对于每个关键词的计算都是认为它们相互独立，其实真实情况下，单词与单词之间是有相关性的（近义词、反义词等、词性等），如果先将关键词聚类，再用朴素贝叶斯计算，可能结果会更好。