Kaldi 学习基础篇（三）-- AWK 基础

原创：静默 Kaldi 学习 昨天

Kaldi 是一个夹杂了多种语言的综合工具，其中一种语言就是上一章中介绍的 Shell 语言，其实 AWK 也是Shell 语言中的一个工具，但是之所以单独拿出来进行介绍的主要原因在于，AWK 虽然能过跟 Shell 融合处理文本，但是它有其独有的语法。

大家可能会有这样的疑问，既然 Shell 可以处理，为什么还要AWK这样的陌生工具夹杂在一起呢?Awk 的优势在于能够快速便捷的处理文本等一系列看似繁琐的任务，用机器学习的专业术语来说，就是特征预处理比较方便。

本节不会面面俱到的讲解AWK，因为这个工具本质上已经可以是一门独立的语言。因此，这里主要通过作者总结出一些 AWK 最为常用的方式和方法。如果读者想要全面的学习 AWK， 那么希望读者自己寻找相应的书籍。

本文文主要通过一个具体的例子来说明 AWK 的基础方法。

AWK 之格式细化

AWK 是一个按行处理的工具，可能读者会问什么叫按行处理的工具?这里举个简单的例子进行说明。

```
1 我,你,它
2 狗,猫,兔
```

假如上面两行存放在文件 Readme 中，其中大家可以看到每一个字都是使用逗号隔开。如果我希望获得这里有多少个字，大家应该怎么处理呢?可能小伙伴们会毫不犹豫的说6个字，那如果 Readme 中有不确定行数的数据，不确定列数的数据时，你还能这么轻松的说出该文件中有多少个字么?答案当然是不能。而AWK 能够让你快速给出结果。

```
awk -F ',' 'BEGIN {sum=0;}                          \
            {for(i=1; i<=NF; i++){sum=sum+1;}} \
            END {print sum }' Readme
```

使用上述代码即可顺利获取该文件中有多少个字。大家现在不需要因为看不懂代码而担忧，这里给出代码的意义在于让读者有一个清楚的认知。另外，这段代码看似简单，但内容却是涵盖了 AWK 的大部分基础。接下来的内容，主要通过这一段代码来讲解 AWK。

正如这段代码中所看到的，开头一个AWK ，这里明确声明使用工具awk 来进行行处理理业务。上文已经提到过，awk 是一个按行处理文本的工具。那么问题就来了，如果是按照行处理，文本的存储是千奇百怪的，我们该怎么区分呢?就如 Readme 中的内容一样，我们现在是按照逗号区分每一列，如果我们使用空格区分，那么这段代码还能正常运行么?如果不能运行，我们该如何修改以至于使它能过正常运行呢?

正如代码中看到的，大写字母 -F ',' 表示一行中按照逗号进行列分隔开来，也就是说，一行中每一列的分隔符是逗号。

接下来大家通过代码可以看到，一个单引号引起了除去 Readme 之外的所有内容。而正是这些内容，最终计算出 Readme中有多少个汉字。

我们接下来介绍 BEGIN { sum = 0;} 和 END {print sum},正如中文对应的意思，BEGIN 译为 开始，END 译为 结束。

```
BEGIN { sum = 0;}
```

这段代码的意思为:在读取 Readme 内容之前，首先执行 BEGIN 内的语句，这里的意思是对sum初始化并附值为0.

```
END {print sum}
```

相反，对于END 这段代码的意思是:当 Readme 中所有的文本处理完之后在执行END内的代码。

注意: 正如上文所说，awk 是一个按行处理文本的工具。理论上来说，每一行都会执行给出的代码，但是，这里的BEGIN 后面跟的第一个{}内的语句和END 后面的所有语句句在整个执行过程中有且仅执行一次。

紧接着，我们介绍中间行代码:

```
{for(i=1; i<=NF; i++){sum=sum+1;}}
```

首先，读者需要明确一点，这一行代码是否是Readme 中每一行内容都会执行这一行代码，换句话说，是否这一行代码会处理 Readme 中所有行的内容呢?答案是肯定的。这里其实就是一个简单的求和，首先我们需要确定一个符号:NF

什么是*NF*呢?

前文提到过，Readme 中是使用逗号进行每一列分隔。那么此处NF的意思就是说明一一行中有多少列。

*NF*的作用用是什什么呢?

NF在这段代码中的作用在于界定该行中有多少列，从而通过for 循环求的该行中有多少个汉字。

注意: NF 是从 1开始记数的。

读者看到这里可能对这段代码有了更加明确的理解。但是读者会问，为什么NF是从1开始记数而不是0开始呢?

答案其实很简单，在awk 中，如果想要输出一行中第N列的数据，仅仅需要使用如下代码:

```
{print $N }
```

但是在 awk 中，一个明确的规定，如果使用 $0 则表示该行的所有内容。故而，NF 从1 开始记数而并非是0.这也是为什么代码中for是从1开始，而并非是0的原因。

注意: 在awk 中，如果需要输出某列数据，仅需要使用 $ 这个符号在加上列数即可。

符号意义

```
NR 表示行号
```

如需要输出 Readme 中每一行的行号则使用如下代码:

```
awk '{print NR}' Readme
```

注意: 从结果中同样可以看出NR默认是从1开始记数的，但是如果因为某些特殊的原因，需要从0开始记数，那么，读者可以使用BEGIN 并在打括号中进行NR默认值的修改。

总结

本章主要介绍了了跟kaldi相关的awk 工工具的使用用方方法，而而这些方方法也都代kaldi 脚本中有所体现。

如有需要欢迎添加微信公众号以及微信群