# 项目2：使用Stylometry要验证作者

**投稿截止日期：**

第1阶段： **下午5:00，周五二零一九年十月十八日** 为伪

第2阶段： **下午5:00，星期五2019年10月25日** 的代码。

价值： **20%** CITS1401的。

*需单独完成。*

您应该构建一个包含以下问题解决方案的Python 3程序，并在LMS上以电子方式提交程序。不允许使用其他提交方式。

您应已阅读并理解大学的学术行为准则。根据这项政策，您可以与其他学生讨论理解该项目所需的一般原则，但是您提交的工作必须是您自己努力的结果。因此，将使用窃检测和其他系统来检测潜在的渎职行为。此外，如果您提交的不是您自己的工作，那么您会学到的东西很少，因此很可能会通过期末考试。

您必须在上面列出的提交截止日期之前提交项目。根据西澳大学的政策，将在提交作业的最后期限之后的每一天（或不足一天）扣除10%的滞纳金。但是，为了便于及时标记作业，在截止日期后的7天后将不允许提交任何作业。

**概述**

西澳大学和全国每所大学（可能在银河系周围）一样，都非常担心以鬼笔迹提交作业。这也称为合约作弊。无论您叫什么名字，鬼笔迹都是关于让其他人来完成您的工作，但要像提交只是您的工作一样提交它。在这种情况下，我们关注论文。据信发病率很低，但这显然不是一件好事。

从不同的角度来看，关于不同的作者的作品是否实际上是这些作者的争论在不同时期引起了激烈的争论。例如，威廉·莎士比亚将所有作品归功于威廉·莎士比亚吗？检查这两个问题的一种方法是使用笔测法。也就是说，不是像寻找可疑窃时那样直接查看文本的内容，而是造型师寻找风格上的相似之处。换句话说，假设作者将对相似的内容，小说，非小说等使用相似的样式，则特定作者使用语言的方式相似，而不是页面上实际单词的相似性。

**CITS1401 Python项目2的计算思维2019年第2学期**

该项目的工作是编写一个程序，读取一个或两个包含要分析的作品的文本文件，并为每个文件建立一个概要文件。然后列出配置文件，或者如果有两个文本文件，则比较两个配置文件，返回一个分数，该分数反映了两个作品之间的风格差异；低分（低至0）表示同一作者可能对这两部作品负责，而高分则意味着不同的作者。

**规范：您的程序需要执行的操作**

**输入：**

您的程序必须定义具有以下签名的main函数：

def main（textfile1，textfile2，feature）

第一个和第二个参数是带有要分析工作的文本文件的名称。 。第三个参数是将用于比较文档配置文件的功能类型。允许的特征名称为：“标点符号”，“字母组合”，“连词”和“复合”。

**输出：**

需要该函数按以下提供的顺序返回以下输出：

•来自成对比较的分数四舍五入到小数点后四位；

•包含第一个文件的配置文件（textfile1），以及

的字典•包含配置文件的字典第二个文件（textfile1）

**的详细说明**

•为了本项目的目的，句子是一个单词序列，后跟句号，问号或感叹号，而反过来又必须带引号（因此句子是引号或口头表达的结尾）或空白（空格，制表符或换行符）。因此：

*这是一些文本。这是更多的文本，*

包含一个句子，后跟另一个句子的开头。

•您需要使用字典来创建输入文件的配置文件。每个文档的配置文件将包含某些单词（不区分大小写）的出现次数和标点符号。

•所计数的单词或标点符号取决于输入功能，这些输入特征可以是：“标点符号”，“单字组合”，“连词”和“复合”。

•对于 **连词**：您的程序需要计算以下单词的出现次数：

“还”，“虽然”，“和”，“作为”，“因为”，“之前”，“但是”，“用于”，“ if”，“ nor”，“ of”，“ or”，“ since”，“ that”，“ though”，“ until”，“ when”，“ whenever”，“ whereas”，“ where”，“，而”，

‘尚’



**CITS1401计算与Python项目2第2学期**

2019•思考 **对**unigram：你的程序需要以统计文件中的每个单词出现的次数。请考虑文件中包含的以下三行文本：

这是一个文档。这只是一个文档测试不
会引起问题

字数将为：“ a”：3，“ document”：2，“ this”：2，“ is”：2，“ only”：1，“ should”：1，“ not”“ 1，”cause“：1，”problem“：1

•对于 **标点符号**：您的程序应计算某些标点符号：逗号和分号。此外，您的程序还应计算单引号和连字符，但只能在特定情况下使用。具体来说，您的程序应该对单引号进行计数，但只能在单引号显示为带字母的单引号时，即表示诸如“不应”或“不会”之类的缩写。（撇号是（您的程序应该计算破折号（减号），但仅当它们被字母包围时，才表示复合词，例如“ compound-word”。其他标点符号或字母，例如“。”当不在句子的末尾时，应将其视为空格，因此可以作为单词的结尾。为此，数字串也是传达信息的单词，因此，在不太可能出现浮点数的情况下，如3.142所示，它被视为两个单词

*注意：我们将使用的某些文本包含连字符（即“-”）。这应视为空格字符字符*

•对于 **复合**：您的程序应包含数字标点符号（如上所述）和连词的出现。此外，您的程序还应该向配置文件添加与文本相关的两个其他参数：每个句子的平均单词数和每个段落的平均句子数（其中一个段落是任何数目的语句，接着是一个空行或通过的文本的结尾的

。•每个的字和标点符号应放置，与它们各自的计数一起，在字典中，这就是所谓的 *简档*。•

第一输出由麦n函数是应使用标准距离公式计算的相应配置文件之间的距离：

$$\text{ddddddddd} = \sum_{ii} (\text{pppppppppddppppppp1}_{ii} - \text{pppppppppddpppp2}_{ii})^2$$

•主函数返回的第二个和第三个输出是与第一个输出的 *配置文件* 相对应和第二个文本文件。的形式返回的 *配置文件* 以字典，其中每个单词都是键，值是键的出现次数，例如 {“ also”：10，“ got”：6}，其中 “ also” 和 “ got” 是键，分别发生了10次和6次。

**使用Python Project 2进行CITS1401计算思维2019年第2学期**

**示例：**

从LMS上的Project 2文件夹下载project2data.zip文件。提供了一个示例交互作为sampleanswers.txt，您可以在sampleresult.txt中找到它。结果基于三个文件：sample1.txt和sample2.txt，均摘自Mark Twain的“密西西比岛上的生活”。

**一些要检查的**

文本文件zip文件中还包含一些文本文件，供您试用。除“袋鼠”外，所有文本均来自古腾堡计划（www.gutenberg.org）。所有文件的末尾都有一个长文本，其中包含Gutenberg项目的许可和使用条款。我在文件licenseNterms.txt中提供了Gutenberg条款和许可，而不是在文本中保留了它们，因为这可能会影响配置文件。

**作者标题小说/非小说类**

亨利·劳森布什小说的孩子
DH劳伦斯的无意识幻想小说
马克·吐温在密西西比州的生活小说
DH劳伦斯·海和撒丁岛的非小说
DH劳伦斯·袋鼠小说
马克·吐温哈克贝利·芬恩小说的冒险
安德鲁·巴顿'Banjo'Paterson
 三象力量小说

一点警告。如果决定从Project Gutenberg下载自己的文本，请注意，许多文本都包含伪造的Unicode字符。不幸的是，我们在CITS1401中使用的文件输入输出功能（我每天都使用）仅适用于标准ASCII字符集，因此，如果文本中包含Unicode字符，则会导致异常。尽管Python能够很好地处理Unicode，但仍需要特殊的输入输出功能，这超出了本单元的范围。我所做的是使用Unix命令：cat-vet *filename* 使Unicode字符在ASCII字符集中可见，然后使用文本编辑器将其删除。（单调乏味的）。

**重要提示：**

您将注意到没有要求您编写特定的功能。那已经留给你了。但是，与Project 1一样，， **程序必须定义的顶级函数这一点很重要 main（）** 如上所述。 main（） 然后应调用其他函数。 （当然，这些可能会调用其他函数。）

**CITS1401 Python项目2的计算思维第二学期2 2019**

年之所以重要，是因为当我测试您的程序时，我的测试程序将调用 main（） 函数。因此，如果您未能定义 main（）或使用其他签名定义它，则我的程序将无法测试您的程序。

**要避免的事情：**

程序需要避免一些事情。

• **不允许** 导入 **任何** 除外，Python模块 math 或 os。尽管使用其他模块是完全明智的做法（以及我经常可能这样做的方式），但它消除了项目不同方面的许多要点，这是关于练习创建代码以准确

提取字符串的各个部分的要点。您需要的内容，以及使用基本的Python结构（在本例中为字典）。

•请 **不要** 假设输入文件名以.txt结束。在非Microsoft Windows的系统中，诸如.csv和.txt之类的文件名后缀不是必需的。

•请确保您的程序 **不会** 调用 输入（） 或者 print（） 函数。这将导致您的程序挂起，等待我的自动化测试系统无法提供的输入。实际上，将发生的是标记程序检测到调用，并且根本不会测试您的代码。

**提交：**

**第1阶段：**

提交一个PDF文件，其中包含根据L2软件开发过程和项目1阶段1提交中讨论的准则解决问题的方法和/或伪代码。在提交之前，您需要与实验室演示者讨论文档。必须之前提交此文件， **201910月18日下午5:00** 在LMS年以避免在Project 2评分中扣除10%。这将是您在课程中开发的解决问题能力的形成性反馈。如果您不提交文件，将从您获得的第二阶段提交成绩中扣除项目总分数的10%。

**第2阶段：**

之前通过LMS提交包含所有功能的单个Python（.py）文件。 **于201910月25日下午5:00** 在LMS上年

*如果您有特殊考虑或延迟提交，则需要联系单位协调员。*

**标记专栏：**

您的程序将被标记为30分（后来缩放为最终标记的20%）。

根据程序完成多项测试的程度（反映程序的正常使用情况）以及程序如何处理各种错误状态（例如不存在输入文件），将在30分中获得22分。除了



之外，您还需要创造性地考虑程序可能面临的输入问题。

30个标记中的8个将是 *样式* （4/8）“代码清晰易读”， *效率为* （4/8）“您的程序结构良好且运行高效”。对于样式，请考虑使用注释，明智的变量名，在程序顶部的名称等。（请查看您的讲义，在此处进行讨论。）

**样式专栏：**

0乱码，难以理解

1-2风格真的很差

3风格很好或很好，只有很少的失误

4出色的风格，真的很容易阅读和遵循

您的程序将遍历各种大小的文本文件（可能包括大型语料库），因此请尽量减少您的次数程序查看相同的数据项。您可能希望使用字典（或集合，如果您准备阅读文档），而不要使用列表。

**效率规程：**

0代码太不完整，无法判断效率，或者解决了错误的问题

1非常差的效率，附加的循环，不适当使用 readline（）

2可接受的效率，一个或多个失误

3效率高，失误少极高

4效率，应该没有大文件上的问题

正在使用自动测试，以便对所有提交的程序进行相同的测试。有时，程序中发生一个错误，这意味着没有测试通过。如果标记能够发现原因并立即解决问题，那么他们将被允许这样做，并且您现在（已解决）的程序将对测试中得到的分数进行打分，减去2分，因为其他学生将不会从中受益标记干预。不过，这比获得零更好。另一方面，如果该错误很难修复，则标记需要继续进行其他提交。