《大数据综合应用实践》可选题二：

作者重名识别

# 【任务背景】

在科学文献数据库中，作者名是辨识科研工作者的重要依据，然而，一个名字通常对应多个不同作者。英语国家的人名通常有多种写法，比如省略中间名或者以姓氏加名字首字母加代替全名。在英文文献中，中文名需要用拼音表示，中文名字的拼音写法使得重名问题变得更为突出，因为不同字形但是读音相同或相近的汉字的拼音表示是相同的。作者重名问题提高了根据名字检索科技文献时返回准确和详尽结果的难度，降低了检索的正确率和召回率。有效的作者重名识别算法可以给科研机构为科研项目的立项和评审寻找专家，研究人员寻找某领域的专家学者论文，期刊编辑寻找审阅文章和策划办刊的专家，学术会议组织者寻找主题发言学者的过程带来便利。

# 【任务描述】

给定一堆拥有同名作者的论文，要求返回一组论文聚类，使得一个聚类内部的论文都是一个人的，不同聚类间的论文不属于一个人。最终目的是识别出哪些同名作者的论文属于同一个人。

解决这一问题的常用思路就是通过聚类算法，提取论文特征，定义聚类相似度度量，从而将一堆论文聚成的几类论文，使得聚类内部论文尽可能相似，而类间论文有较大不同，最终可以将每一类论文看成属于同一个人的论文。一个经典的方法是使用原子聚类的思想，大致思路是首先用较强的规则进行聚类，例如：2篇论文如果有2个以上的共同作者，那么这俩篇论文属于同一类，这样可以保证聚类内部的准确率，随后用弱规则将先前的聚类合并，从而提高召回率。有些工作考虑了传统特征的局限性，所以利用了低维语义空间的向量表示方法，通过将论文映射成低维空间的向量表示，从而基于向量使用聚类方法。

本指导书提供一个大型的论文元信息数据集pubs\_raw.json，包含20多万篇论文的元信息，在硬件资源允许的情况下，学生可以直接提取该数据集中的文本信息作为word2vec的训练数据，或者自行从该数据集中筛选用于消歧的作者名和名字对应的论文id，即生成key为作者名，value为出现该作者名的论文id列表的字典，格式与name\_to\_pubs\_train\_500.json，name\_to\_pubs\_test\_100.json相同，也可分别从这两个json文件中提取出用于测试和训练的作者名与论文id数据。测试集中包含的作者名称不应少于20人。上述文件均位于附件文件夹的na-data-kdd18文件夹中。

本实验鼓励学生设计自己的作者重名识别模型，实现对给定数据集中作者重名的识别。学生也可以参考本任务书的附录提供的模型设计思路。学生自行从附件的name\_to\_pubs\_train\_500.json中选取一定数量的数据用于训练模型，并从附件的name\_to\_pubs\_test\_100.json中选取至少20个作者名字，并对这些名字对应的论文进行划分，采用任务书要求的评测指标评价效果。要求实现的模型输出作者名对应论文id的分类结果，输出文件格式参考附件文件夹下的output\_example.json，即key为作者名，value为一个二维不定长列表的字典，value的第一维长度是作者名对应不同作者的数量，第二维长度是每个作者对应的论文数量。

# 【实验目的和要求】

1、通过查阅相关资料和文献，了解作者重名识别（Name disambiguation）的已有方法；

2、学习文本表示的相关方法，如TF-IDF、Word2Vec等方法；

3、设计自己的作者重名识别模型，实现对给定数据集中作者重名的识别；

4、使用Python或Spark等集成开发环境编码实现作者重名辨识模型；

5、若学生选择参考指导书提供的模型设计思路实现模型，需要掌握Spark分布式集群和HDFS分布式文件管理系统的使用；

6、要求学生从name\_to\_pubs\_test\_100.json中任意挑选出至少20个作者名字并对它们对应的论文进行识别，将结果保存为格式与output\_example.json相同的文件，采用指导书要求的评测指标评测识别效果并将评测结果保存到csv文件或其他格式的文本文件中。

# 【输入数据】

下载链接：<https://static.aminer.cn/misc/na-data-kdd18.zip>

原始数据集为json格式，数据集由多篇文章的json数据组成，数据的格式如下：

{  
 **"5b5433eae1cd8e4e150354ca"**: {  
 **"authors"**: [  
 {  
 **"name"**: **"Runzhi Zhang"**,  
 **"org"**: **"Institute of Zoology"**,  
 **"id"**: **"5b5433e6e1cd8e4e15f992c7"** },  
 {  
 **"name"**: **"Hongbin Liang"**,  
 **"org"**: **"Institute of Zoology"**,  
 **"id"**: **"5b5433f3e1cd8e4e1516badf"** },  
 {  
 **"name"**: **"Li Ren"**,  
 **"org"**: **"Institute of Zoology"**,  
 **"id"**: **"5b5433e7e1cd8e4e15fae6ab"** },  
 {  
 **"name"**: **"Guangxue Zhang"**,  
 **"org"**: **"Institute of Zoology"**,  
 **"id"**: **"5b5433f8e1cd8e4e151fdcc5"** }  
 ],  
 **"title"**: **"Induced life cycle transition from holocycly to anholocycly of the Russian wheat aphid (Homoptera: Aphididae)"**,  
 **"keywords"**: [  
 **"Induced transition"**,  
 **"Life cycle pattern"**,  
 **"Russian wheat aphid (diuraphis noxia mordvilko)"** ],  
 **"venue"**: **"Science in China, Series C: Life Sciences"**,  
 **"year"**: 2001,  
 **"abstract"**: **"The Russian wheat aphid (RWA), Diuraphis noxia (Mordvilko), exists with holocyclic life cycle in Tacheng, Xinjiang in Northwest China. It produces males and oviparae to mate and oviposit for overwintering by eggs. Under laboratory conditions with 14 h/d photophase and temperature not lower than 15°C, RWA occurred in parthenogenesis and produced no males. The laboratory populations of Russian wheat aphid, which were kept under natural conditions in fall by 15th, 49th and 81st generation while wild populations produced males and oviparae for mating, produced males and oviparae with their number decreased gradually, but viviparae and nymphs increased sequentially. As a result, it produced a small amount of oviparae and no males emerged in fields by 49 generations' reproduction in laboratory. After development of 81 generations, oviparae happened occasionally and no eggs occurred for overwintering instead of viviparae and nymphs. A hypothesis of RWA disastrous process was proposed. The life cycle of RWA can be changed from holocycly to anholocycly in its long-term spread and evolution. Anholocycly is more dangerous than holocycly to small grains for its strong adaptability and dispersal ability."** }

}

最外层的key值为paperid，最外层的value值为论文信息，包含**authors**（作者信息）、**title**（论文标题）、**keywords**（关键词）、**venue**（期刊/会议名称）、**year**（发表时间）、**abstract**（论文摘要信息）。其中**authors**为一个list列表，包含若干个以json格式存储的作者的信息，包括**name**（作者名）、**org**（所属机构）、**id**（作者真实id/标签值）。

# 【输出数据】

附件中的output\_example.json是指导书提供的示例输出文件。学生需按照输出数据格式对模型的识别结果进行整理，并基于输出结果和输入数据中的真实标签值评测模型在测试集上的性能，将评测结果输出到csv表格或其他格式的文档中。

输出数据为json格式，其中key值为作者名，value值为重名识别结果，形式为不定长的多维数组，示例如下：

{

**"hongbin\_liang"**:

[

[**"5b5433f4e1cd8e4e151980a3",**

**"5b5433f1e1cd8e4e15122f8b",**

**"5b5433f7e1cd8e4e151f58da",**

**"5b5433ebe1cd8e4e1503d368",**

**"5b5433e5e1cd8e4e15f8537e",**

**"5b5433f7e1cd8e4e151e9907",**

**"5b5433eae1cd8e4e1501c8b3",**

**"5b5433f3e1cd8e4e1517230f",**

**"5b5433e9e1cd8e4e15012ea8"**

],

[**"5b5433e6e1cd8e4e15f9c81e",**

**"5b5433f7e1cd8e4e151e8746",**

**"5b5433e8e1cd8e4e15fdce2e"**

],

[**"5b5433e6e1cd8e4e15f91559",**

**"5b5433ede1cd8e4e1509c56c",**

**"5b5433f5e1cd8e4e151a1a9e",**

**"5b5433e4e1cd8e4e15f5a382",**

**"5b5433eae1cd8e4e150399a1",**

**"5b5433e7e1cd8e4e15fca048"**

],

["**5b5433f5e1cd8e4e151b8329",**

**"5b5433f5e1cd8e4e151ae703",**],

[**"5b5433e6e1cd8e4e15facacb"**]

]

}

对存在重名问题的作者名为**"hongbin\_liang"**的识别结果中，key值为**"hongbin\_liang"**，value为不定长的多维数组。其中每个数组中元素由作者集合中包含作者名**"hongbin\_liang"**的paperid组成，预测同一个数组中的paperid对应的论文属于现实世界中名为**"hongbin\_liang"**的一个人。**特别说明**：多维数组的不同数组之间交集为空。value的多维数组维度表示模型预测真实世界中名字为**"hongbin\_liang"**的作者的人数。

# 【评价指标】

使用基于pairwise的precision（准确率）、recall（召回率）以及F-measure（F1值）来评测重名作者的识别效果。正确率和召回率的取值范围是[0,1]，这两个分数的值越接近1，表示正确率或召回率越高。F1值则是正确率和召回率的简单调和平均数。pairwise在本次实验中指辨识结果中同一个数组中的论文对pair，pair的定义如下：

设作者名为*a*的作者对应的论文集合为，中的论文根据模型输出中的多维数组生成作者名*a*有关的论文对pair集合：

本文的F1值根据以下定义和公式计算得出：

1. ，正确识别的作者对数量
2. ，错误识别的作者对数量
3. ，未识别出的作者对数量

举例说明：科技文献数据库中的6篇文献A、B、C、D、E、F的**authors**中包含有相同名字的作者名称，其中A、B、C论文中的重名作者的真实标签（**id**）相同，均为label1，D、E的真实标签（**id**）相同，均为label2，F的真实标签（**id**）为label3。则正确的论文对有 (A,B) (A,C) (B,C) (D,E)

假设在预测模型的结果中，A、B、C、D的预测标签相同，E、F预测标签相同。模型预测该作者在现实世界中为2个人。模型输出的预测结果中论文对有(A,B) (A,C) (A,D) (B,C) (B,D) (C,D) (E,F)

模型正确识别的论文对有 (A,B) (A,C) (B,C)，则TP=3

模型错误识别的论文对有 (A,D) (B,D) (C,D) (E,F)，则FP=4

模型未识别出的论文对有 (D,E) ，则FN=1

则precision=3/(3+4)=3/7 recall=3/(3+1)=3/4

实验中自行划分训练集和测试集，假使测试集是100个作者名对应的重名作者数据集，macro-P、macro-R表示100个作者名字对应的precision、recall值的平均值。Macro-F1值表示macro-P和macro-R的调和平均值。

# 【附录】

## 【扩展阅读】

作者重名问题是数据挖掘和信息检索研究方向下的热点问题。有效的作者重名识别算法可以给科研机构为科研项目的立项和评审寻找专家，研究人员寻找某领域的专家学者论文，期刊编辑寻找审阅文章和策划办刊的专家，学术会议组织者寻找主题发言学者的过程带来便利。引发作者重名问题的因素主要有2个。第1个因素，名字存在缩写形式，英语国家的人名通常有多种写法，比如省略中间名或者以姓氏加名字首字母代替全名。数字书目索引与图书馆项目（英语：Digital Bibliography & Library Project，简称DBLP）提供计算机领域科学文献的搜索服务，它只储存这些文献的相关元数据，如标题，作者，发表日期等。如图 1所示，在DBLP中检索D. Johnson，返回结果同时包含Daniel Johnson与David Johnson，这说明署名为D. Johnson的作者的论文可能属于Daniel Johnson和David Johnson。



图 1 DBLP中的名字缩写示例

第二个因素，不同的人名字相同。在英文文献中，中文名需要用拼音表示，不同字形但是读音相同或相近的汉字的拼音表示是相同的，这使得涉及中文名的作者重名问题变得更为复杂和突出。如图 2所示，若在DBLP中搜索Wei Wang，返回结果是包含了212个Wei Wang的预识别结果，并且这一数字还在增加。

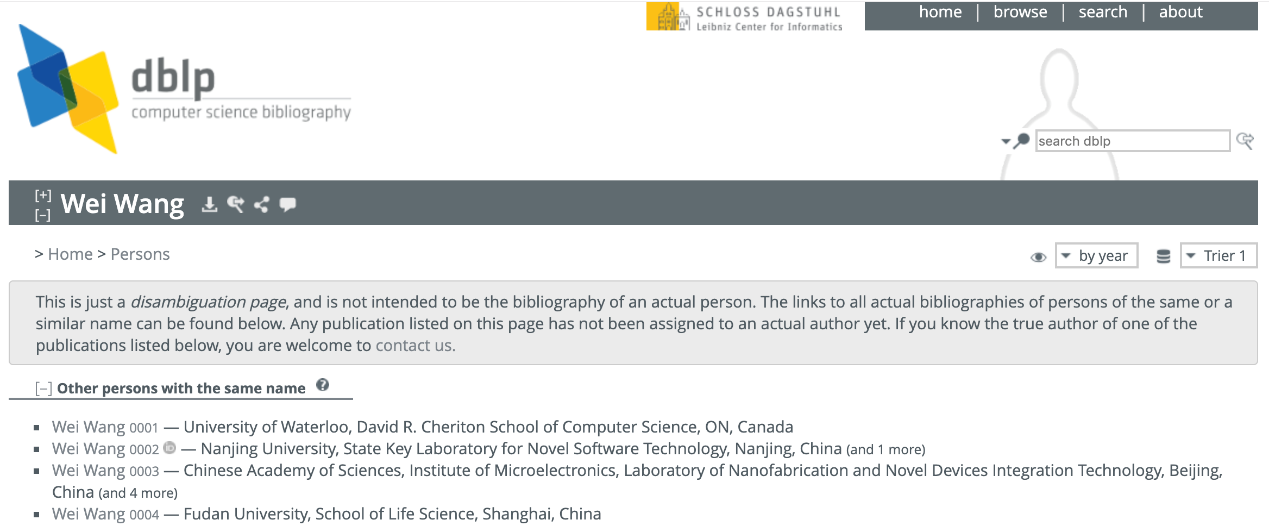


图 2 DBLP的作者重名示例

如图 3所示，若在中国知网数据库中搜索“王伟”，可以得到一个确定的作者列表，知网采用用户认领的方式处理重名问题，但是该方式必须要求用户主动参与，代价高且效率低。因此通过大数据算法智能地对作者重名识别可以节省成本，提高效率。

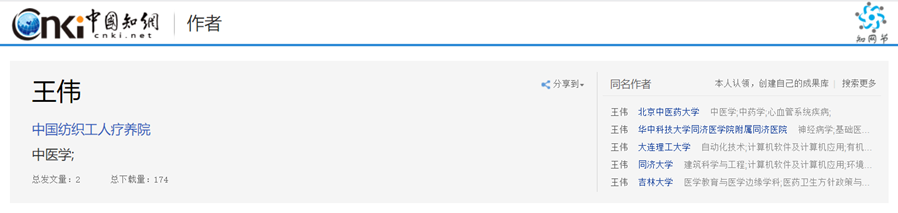


图 3 知网数据库中的作者重名

近十年来，作者重名问题一直受到研究者的广泛关注，在许多应用中，重名识别一直被视为一个具有挑战性的问题，如科学文献管理、人物搜索、社交网络分析等，同时，随着科学文献的大量增长，使得该问题的解决变得愈加困难与紧迫。尽管重名识别已经在学术界和工业界被大量研究，但由于数据的杂乱以及重名情景十分复杂，导致该问题仍未能很好解决。

作者重名问题中存在2个主要挑战。第一个挑战是如何处理日益增长的数据集，每年有成千上万的文献被加入到数字文献数据库中，使得重名问题日趋复杂，计算量不断增大。收录各种论文的线上学术搜索系统(例Google Scholar, DBLP和AMiner等)已经成为目前全球学术界重要且最受欢迎的学术交流以及论文搜索平台。然而由于论文分配算法的局限性，现有的学术系统内部存在着大量的论文分配错误；此外，每天都会有大量新论文进入系统。如何准确快速的将论文分配到系统中已有作者档案以及维护作者档案的一致性，是现有的线上学术系统亟待解决的难题。 由于学术系统内部的数据十分巨大（AMiner大约有130，000，000作者档案，以及超过200，000，000篇论文），导致作者重名情景十分复杂，要快速且准确的解决重名识别问题还是有很大的障碍。

第二个挑战是如何提高识别的精确度。重名识别问题中的冷启动问题属于无监督学习，适合用聚类算法来处理，但是聚类算法的目标聚类数量K是一个对结果影响很大的参数，如何设置K的值对识别的精确度有很大影响。

## 【参考模型】

本指导书提出的解决作者重名问题的模型框架如图4所示。本实验中需要设计实现学习模型，结合使用分布式图计算框架Spark GraphX和分布式文件管理系统HDFS实现作者重名识别。



图4 作者重名辨识方法的框架

该模型的实现基于Spark，以一个作者名*a*为例，设*a*对应的论文集合为，首先通过Spark GraphX构建一个全连接的论文网络，网络中的每个节点都代表一篇论文，通过Spark GraphX提供的API对网络中的任意2个节点与计算不同特征的相似度，并通过一个二分类器判断与是否属于同一个作者，属于则将的标签设置为1，否则置为0；最后将网络中所有标签为1的边构成的连通块输出即为作者名*a*对应的论文划分结果。**本指导书提供的模型设计思路仅供参考，学生可以对模型的细节进行修改和完善。**模型的具体步骤和组成部分如下。

### 【Step 1 特征工程】

特征工程的目的是将论文的文本特征转换成易于计算的形式，比如通过Word2Vec或其他词嵌入模型将数据集中的文本特征转换为向量表示。首先需要对数据集中的文本数据进行清洗，删除或替换文本中的特殊字符，例如将作者姓名中可能存在的“-”和S“.”替换成“\_”，将文本统一转换成小写等。

数据集经过特征工程转换的结果可以有两种形式。第一种是每篇论文都单独对应一个特征向量，在计算相似度时，直接对不同论文的特征向量计算余弦相似度或欧氏距离。第二种是每篇论文的不同属性对应不同特征，例如论文的标题与论文中作者的机构可以分别对应2个词嵌入向量，论文的合作者名字对应一个字符串集合，在计算相似度时，对不同类型的特征采用不同的计算方式，例如向量可以采用余弦相似度计算，字符串集合可以采用jaccard系数计算。本指导书实现的是第二种转换方式，每篇论文都有text，org，coauthor，year共4种特征，其中text是标题与摘要的词嵌入向量，org是作者机构的词嵌入向量，coauthor是合作者名字集合，year是论文发表年份。

### 【Step 2 构建网络】

以一个作者名*a*为例，设*a*对应的论文集合为，首先通过Spark GraphX构建一个全连接的论文网络，即网络中任意2个节点都存在一条边连接。网络中的每个节点都代表一篇论文。节点的特征对应特征工程得到的论文特征。

### 【Step 3 分类判别】

通过Spark GraphX提供的API对网络中的任意2个节点与计算不同特征的相似度，将，，， 4个相似分数组成的向量作为一个二分类器（本文使用逻辑回归模型）的输入，二分类器输出0或1，1代表与属于同一个作者，0代表与不属于同一个作者，将的标签设置为二分类器的输出结果，以边的标签为1为过滤条件从中生成子图，并通过Spark GraphX提供的API输出子图中的连通块，每一个连通块包含的论文都视为同一个作者所写，即得到了作者名*a*对应论文集合的划分结果。

## 【相关知识】

### 【Word2Vec】

Word2Vec能够以无监督的方式从大量语料中学习得到词向量，Word2Vec已经成为NLP领域最常用的词嵌入算法。Word2Vec与TF-IDF向量空间模型相比，大幅提高了文本分类任务的准确度，并在潜在语义分析等领域具有优势。虽然字符串形式的文本可以通过计算编辑距离得到文本在字符上的相似度，但无法得出文本在语义上的相似性。转换为分布式表示（高维空间中的向量）的文本，可以通过计算向量的欧氏距离或余弦相似度得到不同文本的语义相似度，这在文本分类任务中是一种常见的方法。

Word2Vec是一种预测单词分布式表示的算法。Word2Vec由浅层的双层神经网络组成，经过训练可以生成能够还原语言上下文环境的词向量。语义相近的单词转换为分布式表示后，在向量空间中很接近，这使得Word2Vec模型对新词的推广更容易，鲁棒性也更好。许多自然语言处理工作都用到了分布式表示，例如命名实体识别、消歧、解析、标记和机器翻译。



图5 Word2vec模型

Word2Vec可以使用skip-grams或CBOW（连续词袋）模型来生成词的分布式表示。CBOW通过词窗中的周围词来预测中心词，如图5所示；相反地，skip-grams是通过词窗中的中心词来预测周围词，如图5所示。词窗可以理解为一个固定大小的移动窗格，在语料中顺序移动，每次读取固定数量的词语。skip-grams的预测次数要多于CBOW，因而在数据量较少时，skip-grams训练出的词向量质量要高于CBOW，但skip-grams的训练时长也更长。

Spark MLlib实现了基于skip-grams的Word2Vec模型，当输入数据是一段文本时，模型先将文本中的每个单词转化成词向量，再通过求和取均值的方式获得这段文本的向量表示。

使用Spark实现的Word2Vec，将包含论文元信息的输入数据集作为训练集，训练生成输入数据中文本信息（**title**（论文标题）、**keywords**（关键词）、**venue**（期刊/会议名称）、**abstract**（论文摘要信息）、**org**（所属机构））的向量表示。

### 【逻辑回归】

逻辑回归（Logistic Regression）是预测二分类或多分类的常用方法。它是广义线性模型中预测结果的概率的一个特例。它通过最小化公式8所示的损失函数来训练模型：

对于二分类问题，算法将输出一个二元逻辑回归模型。给定一个用***x***表示的数据点，模型通过公式9所示的逻辑函数（sigmoid函数）进行预测：

其中，*x*为输入变量，*w*为逻辑回归要求取的参数。默认情况下，如果，则预测结果为正（1分类），否则为负（0分类），不同于线性支持向量机，逻辑回归模型的输出在概率上有很好的解释性。

在统计学中，逻辑模型是一种广泛使用的统计模型，其基本形式是使用逻辑函数（sigmoid函数）来模拟二值因变量，逻辑函数的图像如图6所示。

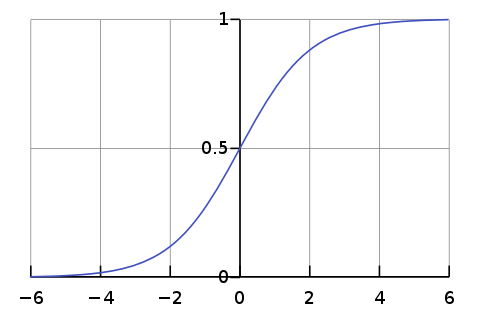


图 6逻辑函数图像

二元逻辑模型的输出是一个具有两个可能值的因变量，例如“成功/失败”，“赢/输”等，这些变量可以用“0”和“1”表示。在逻辑模型中，输出值标记为“1”的对数几率是一个或多个自变量的线性组合；自变量可以是二元变量（两个类，由指示变量编码，指示变量是用来代表类别的数值）或连续变量（任何实际值）；逻辑模型输出值等于“1”的相应概率可以在[0,1]之间变化。

第一步，根据Word2Vec生成的论文特征向量表示，计算重名作者之间的多个相似分数，包括论文的标题与摘要的相似分数、发表时间相似分数、作者所属机构的相似分数、合作者关系相似分数等。相似分数计算公式由学生自行设计；第二步，使用多种相似分数，以及训练集的真实标签值，训练逻辑回归二分类模型，生成重名作者对的预测值，1表示重名作者对是同一个人，0表示不是。第三步，根据预测结果，制定合适的策略，融合重名作者对的预测值为最终的作者重名识别结果，按照格式要求，输出作者重名识别结果。

## 【开发环境】

推荐在ubuntu 1604/ ubuntu 1804下搭建开发环境，Windows10环境下需要对Hadoop进行额外配置。开发环境包含的软件如下：Hadoop 2.7、Spark 2.4.0+、Scala 2.11.8+。

其中Spark需要配置成Standalone模式（单机伪分布式）。注意hdfs与local模式的区别，需要先配置和初始化hdfs空间。

## 【参考文献】

参考文献见附件

## 【提交材料】

**1、实验报告**

所交实验报告至少包括以下内容（可以增加）：

（1）数据介绍；

（2）方法介绍；

（3）实验结果与分析；

（4）关键代码说明；

（6）实验总结。

2、代码。

3、PPT。