FIND:基于特征工程的增量同名消歧方法

赛道二 第一名 小孩子才做选择题

颜杨 尹子长 丁丽媛 郭聪颖

北京邮电大学网络与交换技术国家重点实验室

## 摘要

OAG-WhoIsWho系列竞赛赛道二挑战的基础任务目标是论文消歧，根据论文的详细信息以及作者与论文之间的联系，去区分属于不同作者的同名论文，然后准确快速的将论文分配到系统中已有作者档案，获得良好的论文消歧结果。本文介绍了基于特征工程的增量的同名消歧模型。此外，本文研究出了合理的数据划分方法，以进一步增强系统性能。实验结果表明，本文提出的模型可以使消歧准确率指标有效提高，最终测试数据集上的消歧Weighted F1指标为0.964，比其它方法的准确率提高了0.03。

**关键词 同名消歧 集成学习 增量学习 数据划分**

## 1 介绍

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，现有的收录各种论文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。故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），导致作者同名情景十分复杂，要快速且准确的解决同名消歧问题还是有很大的障碍。但是良好的消歧结果是确保学术系统中，专家知识搜索有效性、数字图书馆的高质量内容管理以及个性化学术服务的重要前提，也可影响到其他相关领域。

本文旨在开发一个基于特征工程的增量的同名消歧算法，准确快速的将论文分配到系统中已有作者档案，获得良好的论文消歧结果。该任务为OAG-WhoIsWho系列竞赛赛道二的挑战。在这项挑战中提供的数据集记录了205,498篇论文。所有论文的元信息和作者和论文的对应关系，其中包含一个待分配论文的列表。

尽管同名消歧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同名情景十分复杂，导致该问题仍未能很好解决。现有工作关于同名消歧的研究方法主要分为基于特征的方法和基于链接的方法。基于特征的方法通过监督学习的方式，[2,3.4,5]等研究工作采用基于特征的方法，利用文档的特征向量来学习两篇论文之间的距离函数。基于链接的方法，也叫做基于图的方法。[8,9,10]的研究工作均采用了基于链接的方法来解决同名消歧任务。这些方法能够利用图的拓扑结构，聚集其邻居节点的信息。

为了解决该任务，本文提出了合理的数据预处理和数据划分方法以及高效且鲁棒性强的分类模型。本文设计的分类模型采用boosting算法，由六个具有两层CatBoostClassifier，两层XGBClassifier，和两层LGBMClassifier，1个Random Forest以及最后的单层CatBoostClassifier结构的cell基分类器的加权联合得到分类决策结果。本文将已有的作者档案与新增论文进行比较，提取合作者，单位机构或者会议期刊之间相似度等众多特征，随后利用分类模型对同名作者进行分类，从而得到与目标最为匹配的作者。根据以前的工作可以知道，CatBoost的优势是对类别特征的处理，同时输出结果的鲁棒性也非常高，不需要费力去调参也能获得非常不错的结果。XGBoost是大规模并行boosted tree的工具，能较好的处理大规模的模型输入，防止过拟合，降低模型的复杂度。LightGBM采用了基于梯度的单边采样（GOSS）方法来过滤数据样例寻找分类数据的分割值，高效而且准确度很高。此外本文在进行数据划分时，对每个作者的论文集中按照发表时间排序，进行顺序抽取，抽取排序最靠后的论文作为验证集训练，贴合论文“增量”的特性，这样的划分方法也使得最终分类准确度有非常明显地提升。

因此，在采用时间增长型数据划分方法之后，本文提出的分类模型可以更好的工作，分类精准度得到明显提升。

本文的其余部分安排如下：第二章主要介绍同名消歧的相关工作，在第三章中将详细描述本文所提出的方法，实验结果在第四章中给出。

## 2 相关工作

目前，关于同名消歧的研究方法主要分为基于特征的方法和基于链接的方法[1]。

### 2.1 基于特征的方法

通过监督学习的方式，[2,3.4,5]等研究工作采用基于特征的方法，利用文档的特征向量来学习两篇论文之间的距离函数。Huang[3]等人最早利用blocking[4]的方法将相似的同名论文构成候选集。然后利用Online SVM学习算法计算两文献之间相似度，并最终采用DBSCAN的方法进行文档聚类。该类方法虽在进行文献著者消歧有较高的准确度，但需人工构建训练集，面对海量数据集进行人工标注几乎是不可能完成的，从而限制了该方法在文献著者消歧中的应用。Yoshida[5]等人提出了一种两阶段聚类的方法，通过第一步聚类，该方法能够更好的学到文档的特征向量表示。Han[2]等人也提出了基于支持向量机（SVM）和朴素贝叶斯的消歧方法, 采用“ K—Way Spectral”聚类算法来对DBLP中的作者进行同名消歧。首先，作者将每篇文章表述成题目、作者和会议称名的特征向量，然后，计算所有文章之间的相似度并组成一个相似矩阵；最后，利用QR分解和特征值分析对文章进行聚类，从而达到同名消歧的目的。该方法存在的不足是仅利用了每篇文章的三类特征，而且Han[2]等人也指出，如果能利用其它特征，比如每篇文章作者的电子邮件等，可通过在一定程度上提高实验结果的精度。后来，Louppe[6]等人利用分类器学习论文之间的相似度，并结合半监督的分层聚类(HAC)方法[7]生成最终聚类结果。

### 2.2 基于链接的方法

基于链接的方法也叫做基于图的方法。[8,9,10]的研究工作均采用了基于链接的方法来解决同名消歧任务。这些方法能够利用图的拓扑结，聚集其邻居节点的信息GHOST[11] 仅通过共同作者信息，为每个歧义作者名称构建了文档图。它使用经过精心设计的相似性函数，并使用Affinity Propagation(AP)算法[12]来生成聚类结果。Tan[13]等提出了一种基于搜索引擎的方法来对同名作者的论文进行消歧。其算法的基本思想为：将每篇文章的题目放到搜索引擎中，利用回传的网格页面作为特征，然后采用各种层次聚类算法来对同名作者的所有文章进行聚类，从而达达到消歧的目的。唐等人[14]采用隐马尔可夫随机场方法，利用一个统一概率框架来模拟节点特征和边缘特征。张等人[15]通过基于文档相似性和共同作者关系的三张图，来学习图的向量表示，从而解决同名消歧问题。

在这篇文章中抽取了60多个有效的相关特征，充分提取了论文和作者之间的潜在信息，并在增量的同名消歧比赛中达到了较好的效果。

## 3 方法

### 3.1 数据预处理

为了把数据输入到本文提出的模型中，首先对应用到的数据进行了清洗，去掉数据中的上下划线，特殊字符等。然后对数据格式进行了整理，将所有的数据按照作者id划分为数据集合，一个作者id的所有论文信息集合结构如下：{aid: {year: {orgs,venues,keywords,coauthor}}，其中aid为作者的id，year为论文发表年份，orgs为作者所属机构，venues为发表会议，keywords为关键词，coauthor为共同作者。将这些信息包括abstract，keywords等进行embedding处理，然后进行存储。

### 3.2 正负样本对的构造

为了提高模型的分类准确度，本文在训练样本中采取了正负样本组成三元组的形式进行模型训练。对于每个作者构造正负样本集合，然后在数据集中随机取样构造正负样本对，将正负样本对输入模型中进行训练，这种对抗式训练对于模型预测的精确度的提高有很大的帮助。

### 3.3 特征工程

根据已有的特征数据，包括共同作者coauthors，论文发表年份year，作者所署机构org，论文发表会议venue，论文标题title，论文摘要abstract，论文关键词keywords，构造特征向量。根据对这些特征对论文分类精确度影响的分析，本文构造了一些新的组合特征，总结模型的所有输入特征以及意义介绍如表一。

|  |  |
| --- | --- |
| **特征名称** | **意义** |
| **coauthors\_count** | 所有共同作者的数量 |
| **coauthors\_count\_by\_all\_count** | 共同作者数量占所有作者的比例 |
| **coauthors\_count\_by\_this\_coauthor\_count** | 共同作者数占当前论文作者数的比例 |
| **this\_paper\_coauthor\_count** | 一篇文章共同作者的数量 |
| **this\_paper\_coathor\_count\_by\_all\_coauthor** | 一篇文章共同作者占所有共同作者的比例 |
| **this\_paper\_coauthor\_count\_by\_this\_paper\_coauthor\_count** | 归一化 |
| **min\_diff** | 发表年份与数据集中年分最小值之间的差 |
| **max\_diff** | 发表年份与数据集中年分最大值之间的差 |
| **mean\_diff** | 数据集中年分的平均值 |
| **meadian\_diff** | 数据集中年份的中值 |
| **min\_max\_avg\_diff** | 发表年份与数据集中年份的最大值，最小值的均值的差 |
| **is\_in\_range** | 发表年份是否在规定的范围内 |
| **this\_year\_count** | 发表年份出现的次数 |
| **this\_year\_count\_by\_all\_year** | 发表年份的数目占所有年份的比例 |
| **is\_in\_cate\_range** | 发表年份是否出现过 |
| **before\_one** | 发表年份前，距离当前发表年份最近的年份差值 |
| **before\_two** | 发表年份前，距离当前发表年份第二近的年份差值 |
| **later\_one** | 发表年份后，距离当前发表年份最近的年份差值 |
| **later\_two** | 发表年份后，距离当前发表年份第二近的年份差值 |
| **venue\_max\_score** | 发表会议名称与作者以前发表过会议的最大Jaro–Winkler distance |
| **venue\_mean\_score** | 发表会议名称与作者以前发表过会议的平均Jaro–Winkler distance |
| **venue\_max\_add\_score** | 发表会议名称与作者以前发表过会议的最大字符集合得分 |
| **venue\_mean\_add\_score** | 发表会议名称与作者以前发表过会议的平均字符集合得分 |
| **venue\_is\_match** | 作者以前是否在当前会议发表过论文 |
| **venue\_score\_add\_score** | venue\_max\_score + venue\_max\_add\_score |
| **org\_max\_score** | 作者机构名称与作者以前的机构名称的最大Jaro–Winkler distance |
| **org\_mean\_score** | 作者机构名称与作者以前的机构名称的平均Jaro–Winkler distance |
| **org\_max\_add\_score** | 作者机构名称与作者以前的机构名称的最大字符集合得分 |
| **org\_mean\_add\_score** | 作者机构名称与作者以前的机构名称的平均字符集合得分 |
| **org\_is\_match** | 作者以前是否以当前机构的名义发表过论文 |
| **org\_score\_add\_score** | org\_max\_score + org\_max\_add\_score |
| **org\_year\_abs\_diff** | org\_max\_score对应的年份与当前年份的绝对值差 |
| **keywords\_max\_score** | 关键字与作者以前发表论文关键字的最大Jaro–Winkler distance |
| **keywords\_mean\_score** | 关键字与作者以前发表论文关键字的平均Jaro–Winkler distance |
| **rela\_year\_diff** | 与当前发表年份最近的发表年份的差值 |
| **rela\_coauthor\_count** | 与当前发表年份最近的发表年份论文的共同作者数 |
| **rela\_coauthor\_count\_by1** | 与当前发表年份最近的发表年份论文的共同作者数占当前论文作者数的比例 |
| **rela\_coauthor\_count\_by2** | 与当前发表年份最近的发表年份论文的共同作者数占以前那篇论文作者数的比例 |
| **rela\_org\_mean\_score** | 与当前发表年份最近的发表年份论文的机构名称的平均Jaro–Winkler distance |
| **rela\_org\_max\_add\_score** | 与当前发表年份最近的发表年份论文的机构名称的最大字符集合得分 |
| **rela\_org\_mean\_add\_score** | 与当前发表年份最近的发表年份论文的机构名称的平均字符集合得分 |
| **rela\_org\_score\_add\_score** | rela\_org\_max\_score + rela\_org\_max\_add\_score |
| **rela\_venue\_max\_score** | 与当前发表年份最近的发表年份论文的会议的最大Jaro–Winkler distance |
| **rela\_venue\_mean\_score** | 与当前发表年份最近的发表年份论文的会议的平均Jaro–Winkler distance |
| **rela\_venue\_max\_add\_score** | 与当前发表年份最近的发表年份论文的会议的最大字符集合得分 |
| **rela\_venue\_mean\_add\_score** | 与当前发表年份最近的发表年份论文的会议的平均字符集合得分 |
| **rela\_venue\_score\_add\_score** | rela\_venue\_max\_score + rela\_venue\_max\_add\_score |
| **rela\_keyword\_max\_score** | 与当前发表年份最近的发表年份论文的关键字的最大Jaro–Winkler distance |
| **rela\_org\_max\_score** | 与当前发表年份最近的发表年份论文的机构名称的最大Jaro–Winkler distance |
| **rela\_keyword\_mean\_score** | 与当前发表年份最近的发表年份论文的关键字的平均Jaro–Winkler distance |
| **org\_set\_count** | 当前论文作者机构名称单词集合与作者以前论文机构单词集合的交集 |
| **org\_set\_count\_by\_all\_count** | 当前论文作者机构名称单词集合与作者以前论文机构单词集合的交集占以前论文机构单词集合的比例 |
| **org\_set\_count\_by\_this\_count** | 当前论文作者机构名称单词集合与作者以前论文机构单词集合的交集占当前论文机构单词集合的比例 |
| **venue\_word\_count** | 当前论文会议名称单词集合与作者以前论文会议单词集合的交集 |
| **venue\_word\_count\_by\_all\_count** | 当前论文会议名称单词集合与作者以前论文会议单词集合的交集占以前论文会议单词集合的比例 |
| **venue\_word\_count\_by\_this\_venue\_count** | 当前论文会议名称单词集合与作者以前论文会议单词集合的交集占当前论文会议单词集合的比例 |
| **keyword\_count** | 当前论文关键词单词集合与作者以前论文关键词单词集合的交集 |
| **keyword\_count\_by\_all\_count** | 当前论文关键词单词集合与作者以前论文关键词单词集合的交集占以前论文关键词单词集合的比例 |
| **keyword\_count\_by\_this\_keyword\_count** | 当前论文关键词单词集合与作者以前论文关键词单词集合的交集占当前论文关键词单词集合的比例 |
| **title** | 当前论文标题的隐向量与作者标题隐向量的距离 |

**表一 特征说明**

### 3.4 模型训练

在这部分，模型的一个cell由两层CatBoostClassifier、两层XGBClassifier，两层LGBMClassifier，一个Random Forest以及一个最上层的单层CatBoostClassifier组成，具体参数见表二。模型最终结果由六个cell得结果做均值得到，模型结构图见图一。

特征数据同时分别输入到四个子模型中，最后在最上层的CatBoost中汇总得到输出结果。CatBoost模型可直接对类别特征进行处理，在很多公开数据集上的表现都相当优异。从它的名字也可以看出来（CatBoost = Category and Boosting），它的优势是对类别特征的处理，同时输出结果的鲁棒性也非常高，不需要费力去调参也能获得非常不错的结果。这里首先叠加了两层CatBoost分类器去处理数据，直接抓取训练数据中的类别特征。

XGBClassifier在这里使用的弱学习类型是默认的gbtree ，这种类型不需要另外调参，有利于节约实验时间。XGBoost是大规模并行boosted tree的工具，被广泛用于数据挖掘比赛。它通过代价函数里的正则项很好的控制了模型的复杂度，能较好的处理大规模的模型输入，防止过拟合，降低模型的复杂度。LightGBM 分类器是一个梯度 boosting 框架，使用基于学习算法的决策树。其中，该方法采用了基于梯度的单边采样（GOSS）方法来过滤数据样例寻找分类数据的分割值，高效而且准确度很高。经过前面分类器的处理，训练集中的训练样例梯度已经非常小了，而GOSS 在小梯度数据样例上引入一个常数因子，在减少数据样例数量与保持已学习决策树的准确度之间取得了很好的平衡。最后，将前面CatBoost、XGB、LGBM的输出输入到一个meta-model中，这里meta-model选用一个浅层CatBoost，其输出结果即认为是最终分类结果。

分别将数据输入到六个cell中，然后对所有的结果进行均值计算得到最终的分类结果。

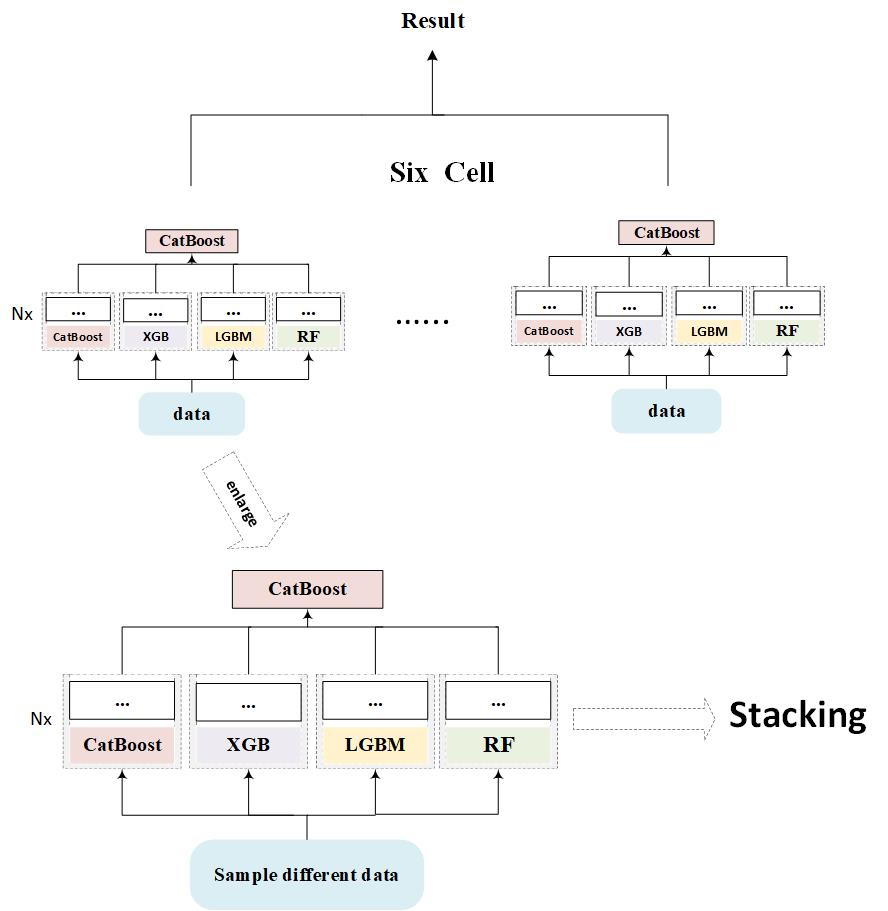
可以看到，本文在模型结构的设计，以及参数的调节上，都是尽量简化。其一是因为所处理的数据规模非常大，涉及到的特征也非常多，手动调参在加大工作量的同时可能也并不能得到比较好的结果，所以本文在调参上尽量简化。其二是在模型的结构方面，设计复杂而又效果好的模型耗时耗力，可能会超出比赛的时间限制， 而且容易产生过拟合的情况，且结果鲁棒性不高。因此在结构设计上，本文直接参考在各大比赛中表现相当不错的boosting 算法，把比赛的重心放在了数据特征工程的构建上。

### 3.5 “增量”的理解

对于赛道二题目的另一个关键考虑是，对于“增量”的理解。本赛道的题目是，同名论文的增量消歧，因此主要是对于“增量”，即新增的论文进行消歧，因此在对已有数据集进行训练的时候，本文就在考虑如何体现已有数据中的“增量”。最自然的想法就是按照时间顺序来体现增量问题，对于每个作者的所有论文，本文通过发表时间进行了排序，那么最近发表的可以认为是该作者的“增量论文”。因此在输入数据进行训练的时候并不是随机抽取，而是在每个作者的论文集中按照时间顺序抽取，抽取排序最靠后的论文作为验证集训练，以此体现增量的特性。

|  |  |
| --- | --- |
| 分类器 | 参数设置 |
| CatBoostClassifier | iterations=180, learning*rate=0.1, depth=7, loss*function='Logloss',eval*metric='Logloss', task*type='GPU', random*seed=RANDOM*SEED |
| CatBoostClassifier | iterations=500, learning*rate=0.1, depth=4, loss*function='Logloss',eval*metric='Logloss', task*type='GPU', random*seed=RANDOM*SEED |
| XGBClassifier | max*depth=7, learning*rate=0.05, n*estimators=180, subsample=0.8,n*jobs=-1, min*child*weight=6, random*state=RANDOM*SEED |
| XGBClassifier | max*depth=4, learning*rate=0.05, n*estimators=350, subsample=0.8,n*jobs=-1,min*child*weight=6, random*state=RANDOM*SEED |
| LGBMClassifier | max*depth=7, learning*rate=0.01, n*estimators=800, objective='binary',subsample=0.8, n*jobs=23, num*leaves=82, random*state=RANDOM\_SEED |
| LGBMClassifier | max*depth=4, learning*rate=0.01, n*estimators=2000, objective='binary',subsample=0.8, n*jobs=23, num*leaves=12, random*state=RANDOM\_SEED |
| CatBoostClassifier | iterations=150, learning*rate=0.1, depth=2, loss*function='Logloss',eval*metric='Logloss', task*type='GPU', random*seed=RANDOM*SEED |

**表二** **分类器结构**



**图一 模型结构图**

## 4 实验

本章节介绍了比赛中提供的数据集和测评方法，并介绍了本模型在数据集上的实验结果。

### 4.1 数据集

比赛提供了一个基于AMiner的数据集用于训练和测试本文提出的模型。数据集已经划分了训练集、验证集和测试集，用于训练的数据集包括221个名字的22839个作者，共计205498篇论文。论文的属性在表三中已经给出。

|  |  |  |  |
| --- | --- | --- | --- |
| **域** | **类型** | **含义** | **举例** |
| id | string | 论文ID | 53e9ab9eb7602d970354a97e |
| title | string | 题目 | Data mining: concepts and techniques |
| authors.name | string | 作者姓名 | Jiawei Han |
| author.org | string | 作者单位 | department of computer science university of illinois at urbana champaign |
| venue | string | 会议/期刊 | Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial |
| year | int | 发表年份 | 2000 |
| keywords | list of strings | 关键词 | ["data mining", "structured data", "world wide web", "social network", "relational data"] |
| abstract | string | 摘要 | Our ability to generate... |

**表三 论文属性**

在比赛给定的验证集和测试集上对本文提出的模型进行测评。

### 4.2评价方法

实验使用Weighted F1作为模型评估度量，其计算方法如下。

对于每一个参与测评的作者，Precision、Recall和Weight的计算方法如下：

其中，*#CorrectlyPredictedToAutho*r为被正确预测到作者的论文数目，*#TotalPredictedToTheAuthor*为全部被预测到作者的论文数目，#*TotalPaperBelongToTheAuthor*为作者拥有的论文数，#*UnassignedPaperOfTheAuthor*为实际应该分配到该作者的论文数，#*TotalUnassignedPaper*为全部待分配论文数。

对于所有作者，WeighedF1计算方法如下.

其中，*M*是作者总数，*Precisioni*为第*i*个作者的*Precision*，*weighti*为第*i*个作者的*Weight*，*Recalli*为第*i*个作者的*Recall*。

### 4.3 实验结果

实验在验证集和测试集上进行了测评，实验结果如表四所示。

|  |  |
| --- | --- |
|  | Weighted F1 |
| 验证集 | 0.86299 |
| 测试集 | 0.96405 |

**表四 实验结果**

在验证集和测试集上本文提出的模型均表现出较好的效果，获得了高于其他模型的性能。

## 5 结论

同名消歧任务确实是一个具有挑战性的问题。在科学文献管理、人物搜索、社交网络分析等应用中，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。在本文中，本文提出了合理的数据预处理和数据划分方法以及高效且鲁棒性强的分类模型。本文的分类模型采用boosting算法，由六个具有两层CatBoostClassifier，两层XGBClassifier，和两层LGBMClassifier，一个Random Forest以及最后的单层CatBoostClassifier结构的cell基分类器的加权联合得到决策结果。由于涉及的数据规模较大，特征较多，本文在模型结构的设计，以及参数的调节上，都采用尽量简化的处理。特别的是，本文在进行数据划分时，对每个作者的论文集中按照发表时间排序，进行顺序抽取，抽取排序最靠后的论文作为验证集训练，贴合“增量”的特性，这样的划分方法也使得最终分类准确度有非常明显地提升。将来将尝试一些新方法来进行更有效的特征的筛选和模型的优化。

## 参考文献

[1] Zhang Y , Zhang F , Yao P , et al. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop[C]// the 24th ACM SIGKDD International Conference. ACM, 2018.

[2] Han H , Giles L , Zha H , et al. Two supervised learning approaches for name disambiguation in author citations[C]// Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on. ACM, 2004.

[3] Jian Huang C L G . Efficient name disambiguation for large-scale databases[C]// European Conference on Principle & Practice of Knowledge Discovery in Databases. Springer-Verlag, 2006.

[4] Steorts R C , Ventura S L , Sadinle M , et al. A Comparison of Blocking Methods for Record Linkage[C]// International Conference on Privacy in Statistical Databases. Springer International Publishing, 2014.

[5] Yoshida M , Ikeda M , Ono S , et al. Person name disambiguation by bootstrapping.[C]// International Acm Sigir Conference on Research & Development in Information Retrieval. ACM, 2010.

[6] Louppe G , Al-Natsheh H , Susik M , et al. Ethnicity sensitive author disambiguation using semi-supervised learning[J]. 2015.

[7] Pedersen T , Purandare A , Kulkarni A . Name Discrimination by Clustering Similar Contexts[C]// Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings. Springer-Verlag, 2005.

[8] Bekkerman R , Mccallum A . [ACM Press the 14th international conference - Chiba, Japan (2005.05.10-2005.05.14)] Proceedings of the 14th international conference on World Wide Web, - WWW "05 - Disambiguating Web appearances of people in a social network[C]// International Conference on World Wide Web. ACM, 2005:463.

[9] Hermansson L , Kerola T , Johansson F , et al. Entity disambiguation in anonymized graphs using graph kernels[C]// Acm International Conference on Conference on Information & Knowledge Management. ACM, 2013.

[10] Pallika H. Kanani, Andrew McCallum, Chris Pal: Improving Author Coreference by Resource-Bounded Information Gathering from the Web. IJCAI 2007: 429-434

[11] Fan X , Wang J , Pu X , et al. On Graph-Based Name Disambiguation[J]. Journal of Data & Information Quality, 2011, 2(2):1-23.

[12] Xiao Y , Yu J . Semi-Supervised Clustering Based on Affinity Propagation Algorithm[J]. Journal of Software, 2008, 19(11).

[13] Tan Y F , Kan M Y , Lee D . Search engine driven author disambiguation[C]// Jcdl. ACM, 2006.

[14] Tang J , Fong A C M , Wang B , et al. A Unified Probabilistic Framework for Name Disambiguation in Digital Library[J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(6):975-987.

[15] Zhang B , Hasan M A . Name Disambiguation in Anonymized Graphs using Network Embedding[J]. 2017.