解题思路和具体试错

基于论文结构信息与文本信息的作者名消歧

IJCAI 2021 - WhoIsWho Task2- 第二名 Complex808 组
谢文锦[1]，刘思源[1]
1 西南大学计算机与信息科学学院

1.整体思路

对于同名消歧的冷启动问题，可将其视为对待消歧名字所对应的论文进行聚类的问题。可根据论文之间的结构相似性，结合文本相似性，量化两两论文之间的相似程度，得到论文的相似性矩阵。从而，可使用聚类算法将论文划分成不同的簇，每一簇对应一个特定作者的论文集合。

另外，对于聚类过程中产生的离群论文，可通过基于相似度阈值的匹配方法，将其分配到现有的簇或者一个新簇中。整体思路示意图如图 1。

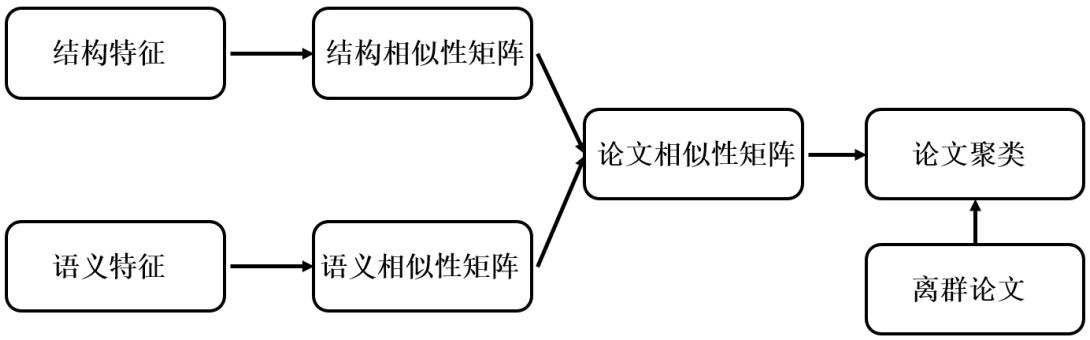

图 1 算法整体思路

2.基于结构信息的论文表征学习

当两篇待消歧的论文之间具有某些共现（co-occurrence）特征时，我们认为论文之间存在连边关系，由此可以得到论文的结构特征。

2.1 异质网络构建

数据集中一篇论文的特征包含 title, abstract, author, venue, organization, year 和 keyword 等。

从经验上分析，首先，当两篇待消歧的论文在除了待消歧的作者名字之外还有其他的共同作者时，两篇论文则很可能是由同一人发表的。其次，如果两篇论文的作者来自同一机构的话，则这也很可能是同一人。另外，两篇论文是研究的同一个研究领域的话，也很可能是来自于同一个人所发表的。

因此，我们将论文特征中的 author、organization、venue、title、keyword 提取出来（后三者视为论文的研究领域），构建异质网络。该网络具有四种类型的节点：论文、论

文的作者、论文的机构（这里的机构采用的是待消歧名字的机构信息）、论文的主题（包含发表刊物、题目和关键词）。

网络如图 2 所示。

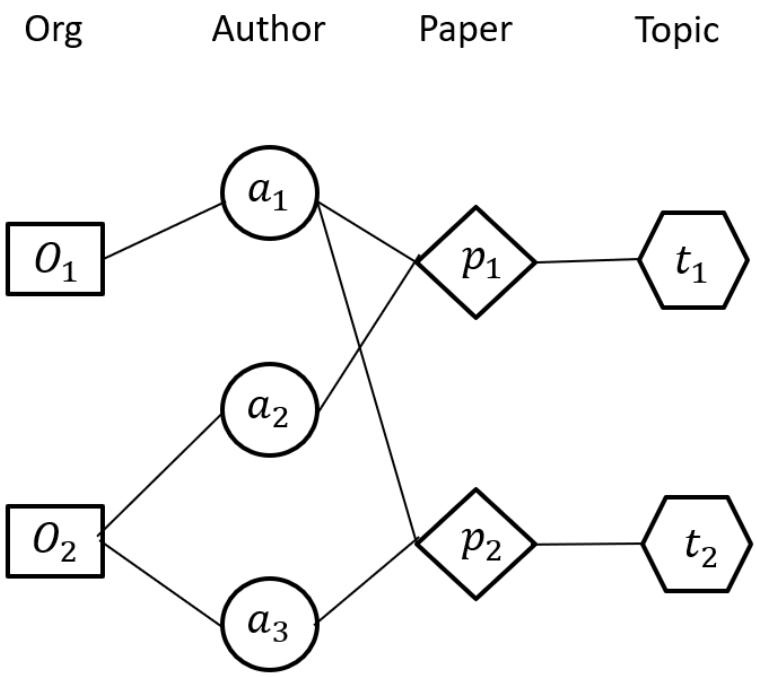

图 2 论文异质网络

2.2 基于元路径的随机游走

Metapath2Vec 是针对异质网络的表示学习而设计的模型，该模型通过基于元路径的随机游走生成由网络中的异质节点所组成的路径，并把路径作为 word2vec 的输入，得到每个异质结点所对应的表示向量[1]。

元路径指的是预先设定的游走规则，在异质网络上进行游走采样时，必须按照该规则进行游走。这里元路径的设置往往具有其语义含义，如设置 APA（Author-Paper-Author）的元路径，则在图 2 中所采样到的路径，如“$a_1 - p_1 - a_2$”，表示论文p_1有a_1和a_2两个作者，a_1和a_2在这里具有共现关系。

根据 3.1.1 的分析，我们采用了 PAPOPTP 的元路径进行随机游走。值得注意的是，由于我们这里只需要得到论文的表示，因此，在随机游走采样时，我们只记录路径上的 Paper 节点即可。因此得到的实际游走路径可抽象成图 3。这里我们将 title 和 keyword 中的词合在一起作为论文的“word”特征。

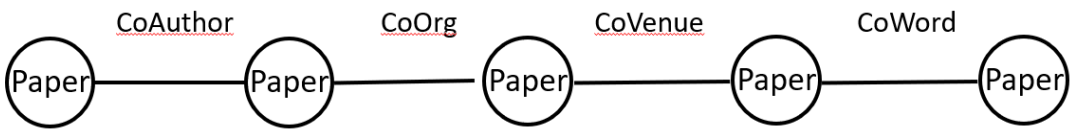

图 3 基于共现关系的论文元路径

同时，在某个节点在元路径指导下朝着某类型的边随机选择下一个节点游走的过程中，会考虑到边的权重，权重越大，那么节点沿着这条关系游走的概率就越大。在实现过程中，我们把 CoOrg、 CoVenue 和 CoWord 处理为单词的共现，把论文的机构、刊物、题目、关键词字段读取成单词（去掉停用词），将对应字段中相同单词的数量作为边的权重。比如，机构“Beijing Normal University”和“East China Normal University”，在去掉停用词“University”后，共有的词是“Normal”，则 CoOrg 边权重记为 1。

在某些情况下，有些关系对于一些论文来说是缺失的，例如某个论文中所有的作者并没有出现在其他任意一个论文的作者列表中，那么对它来说 CoAuthor 这个关系是缺少的，当出现这种情况时，就采用更灵活的策略，即根据元路径中当前缺失关系的下一个关系游走，对于上诉情况，就转而根据它的 CoOrg 关系进行游走。

具体的实验代码详见“utils”cell 模块中的 MetaPathGenerator 类。

2.3 使用 word2vec 得到论文结构关系向量

我们使用 gensim 库中的 word2vec 的 skip-gram 模型训练上节的路径集，将路径集看作语料库，语料库中的论文 id 看作词汇，最后得到每个论文 id 对应的表征向量。Word2vec 训练参数如下：

```
model = word2vec.Word2Vec(sentences, size=100,negative =25, min_count=1, window=10)
```

在采样的路径中，我们并不保存起始点，这样对于图中的孤立节点（即没有边与其他节点相连的节点），这个节点对应的论文 id 不存在于路径集中，也无法得到其节点关系表征向量。我们把这些节点的关系表征向量设为全 0 向量，并保存到离群论文集中，后续再做处理。

在训练 word2vec 的时候，我们采用了 Bagging 的思想，采用上节中的策略采用 k 个路径集，使用 k 个路径集训练得到 k 个 word2vec 的模型，生成 k 组论文向量，每组论文向量求余弦相似性矩阵，再对这 k 个相似性矩阵求均值，得到最终的论文关系相似性矩阵。这部分代码在“name disambiguation (test)”cell 中。

2.4 试错过程

在这里我们使用了 author、organization、venue、title、keyword 这五个特征，实验发现若只是单独使用其中某一个特征的话预测效果都会变差。

除了待消歧作者名的 organization，每篇论文的其他作者也包含 organization，如果考虑所有作者的 organization 来计算 CoOrg 连边的话，实验结果也不好，反而会造成干扰。

同时，在实验过程中，我们发现在游走的元路径中完全加入 CoVenue 和 CoWord 连边时，效果反而会变差，对此我们认为是“根据相同的研究主题就认为两篇文章为同一人所作”这一猜想比另外两者更弱，是一种弱连接关系。同时，根据字段中词的共现也会干扰这种关系。也有论文表明一个研究者职业生涯的研究方向可能会大幅改变[2]。因此，我们在游走时设置了概率 p 参数，每次按元路径游走时，只会以概率 p 走到 CoVenue 和 CoWord 的连边，否则在走过 CoOrg 边后，会直接跳转到 CoAuthor 边。实验结果发现，这样的游走规则，比完全加入 CoVenue 和 CoWord 和完全不加入都要好。

3.基于文本信息的论文表征学习

3.1 论文语义表征学习

这部分的处理比较简单，是将训练集、验证集和测试集中“pub”的所有文本提取到一个语料集中，包括论文的 title、organization、abstract、year。我们把这些语料以空格间隔，合成同一段文本，对这个文本进行预处理，首先把字母小写化，去除各种非字母的符号，接着去掉多余的空格，以空格分词，去掉停用词和长度小于 3 的词。然后使用 gensim 的 word2vec 模型训练词向量。

对每个待消歧的名字，得到其所有论文的语义表征向量，当有的论文的所有词都不存在于 word2vec 模型中时，将其语义表征向量置为全 0，并保存到离群论文集中，后续再做处理。最后求得论文两两间的余弦相似度，得到论文语义相似性矩阵。

此部分的代码在“train word2vec”cell 模块中，Word2vec 训练参数如下：

model = word2vec.Word2Vec(sentences, size=200,negative =5, min_count=2, window=5)

3.2 试错过程

在实验中，我们也尝试了在语料集中删去某些信息，比如 organization、abstract、venue 和 year 等，实验发现效果都不如加入所有信息。我们认为是因为 organization 具有比较强的地域特征，对论文的表示具有很好的帮助。abstract 和 venue 能够提供对论文研究主题的支撑，对表示论文的研究方向是个很好的语料。Year 则反应了论文发表的连贯性，相近年份的词向量具有更高的相似性，比如与 2012 最相似的年份词是 2011 和 2013。在文献[2]中也提到科研工作者的研究兴趣在一个短的时间区间内是不会变化或是在小范围内徘徊的，因此将年份作为语料也能带来更好的效果。

4.论文聚类和离群论文分配

将上诉得到的基于论文结构特征的相似性矩阵和基于文本特征的相似性矩阵加权相加后，得到总的论文相似性矩阵，据此对论文进行聚类，聚类得到的每一个簇视为为同一作者所写。同时，对离群论文进行单独处理。

4.1 基于 DBSCAN 对论文聚类

经过实验测试，我们选择将基于论文结构特征的相似性矩阵和基于文本特征的相似性矩阵按 1:1 的比例相加，这也侧面反应两种特征对论文聚类所做出的贡献是差不多的。

将总的论文相似性矩阵输入 DBSCAN 中，得到聚类结果。DBSCAN 的参数设置为：eps=0.2，min_samples=4，这意味着一个簇中最少的论文个数为 4，这会产生一部分已经划分好的论文簇和许多 label 为-1 的离群点，这些离群点不属于任何簇。我们把这些 label 为-1 的论文加入离群论文集中。除了这些在离群论文集中的论文，我们把其他论文的聚类结果作为这些论文最终的聚类结果。这个结果称为预聚类结果。

在实验中我们对比了其他聚类方法，比如 K-means 和层次聚类，但是这两类聚类方法需要预先设定 K，即聚类簇的数量，我们采取了人为设置的方法，发现效果并不理想。

4.2 离群论文的分配

这一步对分配过程中产生的离群论文进行进一步的划分，将这些离群论文集里的论文用阈值匹配的方法重新分配给已经聚类好的簇或者新的簇中。具体操作如下。

首先对于离群论文集中的每一篇论文，比较它与每个预聚类论文集中的论文，得到跟它匹配相似度最高的一个论文，如果他们两个的相似度不小于阈值 α，则把前者分配到后者所在的簇中；否则把它单独归为新的一个簇。

做完上步，对于离群论文集中的每一篇论文，再比较它与离群论文集中其他每个论文匹

配相似度，如果两者的相似度不小于阈值 α，则把后者分配到前者所在的簇中；否则不变。

其中两篇论文 pi 和 pj 的匹配相似度 s(pi,pj)的定义如下：

1. 初始 s(pi,pj)为 0
2. s(pi,pj) = s(pi,pj) +（pi 和 pj 的共同作者数）×1.75
3. s(pi,pj) = s(pi,pj) + tanimoto（pi 的 venue，pj 的 venue）
4. s(pi,pj) = s(pi,pj) + tanimoto（pi 中待消歧名的 organization，pj 中待消歧名的 organization）
5. s(pi,pj) = s(pi,pj) + （pi 和 pj 中 title 的共词数）/2.0
6. 输出 s(pi,pj)

其中 tanimoto(p,q)指两个序列的 tanimoto 相似度，定义为 p 和 q 两个序列的交集除以并集。

阈值 α 我们取为 1.5。

生成和匹配离群论文的原因是这部分论文往往特征不够明显，造成其与其他论文之间的相似度比较小，或者这些论文本身就是属于一个论文数较少的作者，这些形式的论文用我们上述所说的表征向量学习方法的效果不如使用直接使用特征进行匹配的方法的效果好。

经过上述操作，将离群论文集中的论文也分配到了各自的簇中，将离散论文集和预聚类论文集整合，这样就得到了所有论文的聚类结果，这个聚类结果就是我们最终对于某待消歧名字的消歧结果。

其他

我们对所有上文未提到的文本预处理的操作都是一样的，首先把字母小写化，去除各种非字母的符号，接着去掉多余的空格，若文本需要分词，则在分词后去掉停用词，和长度小于 2 的词。我们使用的停用词表和符号表如下：

```
r = '[!“”"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~—～’]+'
stopword=['at','based','in','of','for','on','and','to','an','using','with','the','by','we','be','is','are','can']
stopword1= ['university','univ','china','department','dept','laboratory','lab','school',
            'al','et', 'institute','inst','college','chinese','beijing','journal','science','international']
```

同时，观察测试集发现，测试集中待消歧名字中有“li-min_zhu”，“hai-bin_song”等名字，其中“-”为我们预设的停用词，与语料库中名字储存的格式有出入，将其改写为“limin_zhu”，“haibin_song”等，再进行论文的分配。

本文的主要解题思路和主要参数都借鉴了《基于网络嵌入和语义表征的作者名消歧》报告，感谢 OAG- WhoIsWho 竞赛中赛道一第一名乔子越王寒雪组的分享[3]。

参考文献

【1】Dong Y, Chawla N V, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks[C]//Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017: 135-144.

【2】Jia T, Wang D, Szymanski B K. Quantifying patterns of research-interest evolution[J]. Nature Human Behaviour, 2017, 1(4): 1-7.

【3】乔子越，王寒雪，《基于网络嵌入和语义表征的作者名消歧》，OAG- WhoIsWho 赛道一第一名分享