数据挖掘报告

李晨昊　2017011466
李逸凡　2017012774
李岳霖　2017011452

2020 年 6 月 21 日

目录

1 项目介绍

在许多应用中，同名消歧 (Name Disambiguation —- aiming at disambiguating WhoIsWho) 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。

收录各种论文的线上学术搜索系统 (例如 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。故如何准确快速的将论文分配到系统中已有作者档案，是现有的线上学术系统亟待解决的难题。

本项目为冷启动论文同名消歧问题提供了两个解决方案：基于 LightGBM 分类置信度与层次聚类的解决方案，以及基于 GSDMM 算法和聚类合并的解决方案。其中前者的效果更好，也是我们最终提交的版本。这两个解决方案的具体思路和实现将在下文详述。

1.1 任务描述

给定一组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。

1.2 数据集描述

数据集分为训练集和验证集，分别都由两个文件组成。

第一个文件是作者信息，组织成一个字典 (dictionary, 记为 dict1)。 dict1 的键 (key) 是作者姓名。dict1 的值 (value) 是表示同名作者集合的字典 (记为 dict2)。 dict2 的键 (key) 是作者 ID, dict2 的值 (value) 是该作者的论文 ID 列表。在验证集中只给出了同名作者的论文 id, 需要我们自己去完成分类工作。

第二个文件是论文元信息，也是一个字典，key 为论文 id, value 为一个对象，含有题目、作者、作者机构、期刊名、发表年份、关键字、摘要。

1.3 评价指标

我们使用了 pairwise F1 作为评价模型的指标。具体定义如下：

$$precision = \frac{\#PairsCorrectlyPredictedToSameAuthor}{TotalPairsPredictedToSameAuthor}$$

$$recall = \frac{\#PairsCorrectlyPredictedToSameAuthor}{TotalPairsToSameAuthor}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

2 基于 LightGBM 分类置信度与层次聚类的解决方案

我们参考了 2019 年比赛第二名的分享 [1]，尝试通过基于二分类的置信度生成论文与论文之间的相似度矩阵，并且通过层次聚类的方式生成最终的聚类结果。

这种方案是我们在最优提交中使用的方案。

2.1 数据预处理

赛方提供的数据质量相对较低，包含很多脏数据，需要进行数据清洗和预处理。

首先，数据中论文的标题、摘要等信息采用了不同的语言，包括中文、英文、日文等等，在不同的语种之间计算文本相似度过于困难，因此我们选择将文本统一翻译为同一种语言。我们选择翻译为英文，因为针对英文 NLP 的技术、预训练模型等都更为成熟。翻译完成后，我们通过 Python 的 `nltk` 包对其进行文本预处理，包括去除停用词、词性标注和 stemming 等等。同时，我们也将原论文使用的语言提取为一个额外的信息字段，以减少论文信息的丢失。

对于缺失的文本字段，我们使用空字符串填充，并且提取各字段缺失个数作为额外信息字段，目的同样是减少信息损失。

2.2 特征工程

完成了数据预处理后，数据集中表意的英文文本字段包括标题、期刊、关键词和摘要。我们通过词嵌入模型将文本转化为词向量，便于后期进行相似度以及二分类模型的计算。我们通过 `gensim`[1] 包，尝试了以下几种词嵌入模型：

- conceptnet-numberbatch-17-06-300
- glove-wiki-gigaword-300

[1] https://www.biendata.xyz/models/category/3643/L_notebook/

- word2vec-google-news-300

上述模型均将一个单词或词根转化为一个 300 维的向量。

另一方面,“作者姓名”字段是一个极其重要的特征。如果两篇文章的姓名字段没有相符合数据，那它们显然不会是出于同一位作者之手。但处理这个字段的困难在于，publication 数据集中对作者姓名字段的表达方式是多种多样的。例如 `Fenghe Qiu` 和 `F. Qiu` 可能是同一位作者。因此比赛的主题除了“同名消歧”之外，事实上还包含了这些姓名字段内容不同而实则是同一位作者的考察。我们对作者姓名进行了手动正则匹配的处理，尽可能减少使用标点符号使用方式、全名或首字母缩写、大小写使用方式等差异导致的，同一作者姓名表示形式不同的问题。

上述问题在“作者所属机构”字段中也存在。我们也进行了类似处理，这能够使对作者姓名和所属机构字段的相似性判定，比直接使用文本相似度的有关公式进行计算得出的结果更加准确。

2.3 模型架构

我们的模型分为两个部分：

- 通过两篇文章之间的相似度判断文章是否为同一作者的二分类器，输出“是”的置信度
- 将上述置信度作为文档间相似度的度量，输出聚类结果的层次聚类器

2.3.1 二分类器

我们选择使用 LightGBM 模型作为二分类器，因为它在传统的机器学习模型中表现最好，远超传统的 SVM 等模型，同时其训练速度也远远快于思路类似的 XGBoost 等，在我们所面对的较大型数据集上有很好的性能表现。

在训练的过程中，我们基于所参考论文给出的超参数，进一步采用 Grid Search 的方式对超参数进行了调优，部分结果如下所示：

num_leaves	max_depth	feature_fraction	performance
200	21	0.7	0.892
200	21	0.8	0.895
200	25	0.7	0.796
200	25	0.8	0.852
300	21	0.7	0.846
300	21	0.8	0.853
300	25	0.7	0.894
300	25	0.8	0.836

LightGBM 模型分类的目标是交叉熵损失函数：

$$H(A,B) = -\sum_i P_A(x_i) \log\left[P_B(x_i)\right]$$

交叉熵最小的实质本质上就是一个极大似然估计，本身就具有概率意义。因此模型在对测试集数据进行预测时通过使用 `predict_proba` API, 输出的不仅是“是”或者“否”的标签，还有判定为“是”的置信度。这个置信度便可以作为文档与文档之间的相似度。

2.3.2 聚类器

在数据挖掘课堂上介绍的几种聚类器度量中，我们选择了层次型聚类，因为其他模型在这里均不适用。

- 基于质心进行分割的聚类方法：无法定义质心个数
- 基于密度的聚类方法：文章两两之间无法定义距离（分类器模型输出的是相似度，不能直接作为距离使用）

在层次型聚类中，另一个要定义的 metric 是两篇文章聚类后的簇与其他文章/簇的相似度，我们探索了使用 MIN, MAX 和 Group Average 的方法，但最终发现在我们参考的文章中提出的以下定义效果最好：

$$d_{AB} = \frac{1}{|C_A||C_B|} \sum_i \sum_j \log(p_{ij})$$

其中 p_{ij} 是相似度矩阵中 A 和 B 的相似度。

3 基于 GSDMM 算法和聚类合并的解决方案

我们还参考了 2019 年比赛第五名的分享 [2]，尝试进行基于规则的聚类，也就是说不考虑训练集 (或者仅用其来帮助调整少量的超参数)，直接在测试集上进行聚类，并在聚类结果的基础上进行基于规则的进一步的合并。以改进聚类结果。

这是我们项目前期尝试的方案，之后因为其效果不佳，所以并不是最优提交中使用的方案。

与基于 LightGBM 分类置信度与层次聚类的解决方案中功能相似的部分将不再介绍，例如数据的预处理等，这些在本方案中都有类似的过程，虽然具体做法不完全一样 (因为后续对本方案没有继续维护了，也没有把新方案中的可能更优秀的预处理方法引入进来)，但是思路是大致一致的。

3.1 基于 GSDMM 的聚类

GSDMM[2] 是一种基于狄利克雷多项式混合模型的收缩型吉布斯采样算法，它具有以下特征，对我们的应用都是非常有益的：

1. 可以自动推断簇的数量，不需要预先指定。
2. 通过调节参数，可以明确地控制平衡聚类结果的正确性和完整性。
3. 收敛迅速。
4. 与基于向量空间模型（VSM）的算法不同，GSDMM 可以解决短文本的稀疏和高维问题。

GSDMM 的输入是一组文本，每篇文本用分词后的词列表来表示。算法的基本步骤如下：

1. 将文本随机分配到指定个数的聚类，这个个数并不是最终的簇数，而是它的一个上界。
2. 维护以下数组：每个簇中的文本数，每个簇中的词数，每个簇中每个词出现的次数。
3. 进行一定轮数的迭代，每轮迭代中，对于每篇文本，计算它归属于每个簇的概率，依据此概率为把它重新分配到一个簇中。
4. 到达轮数时，或者一轮迭代中没有发生任何重新分配时，结束聚类。这时有的簇可能是空的，这时最初指定的簇数上界大于实际簇数。

算法的核心，即计算一篇文本属于一个簇的概率的公式为 (摘自原论文) ：

[2] https://www.biendata.xyz/models/category/3679/L_notebook/

$$p(z_d = z|\vec{z}_{\neg d}, \vec{d}) \propto \frac{m_{z,\neg d} + \alpha}{D - 1 + K\alpha} \frac{\prod_{w \in d}(n^w_{z,\neg d} + \beta)}{\prod_{i=1}^{N_d}(n_{z,\neg d} + V\beta + i - 1)}$$

其中 m_z 表示簇 z 中的文本数，n_z 表示簇 z 中的词数，n^w_z 表示簇 z 中的 w 的出现次数，V 表示总词汇表大小，D 表示文本数，K 表示簇数，α 和 β 是两个超参数。$\neg d$ 表示簇 z 中去掉文本 d 后再统计对应的数据，如果不去掉的话，会更加倾向于将文本分配到它本来所在的簇中。

需要注意的是实际计算累乘的过程中需要使用对数来避免溢出。

原始的算法仅适用于对一组文本进行聚类，实际上我们需要对论文进行聚类，每篇论文有有多篇文本作为其特征。我们对原算法进行了一点修改，对于一篇论文，计算多组文本的分配概率，将这些概率按照一定的权值加权组合在一起。此外，理论上这些文本的性质各不相同，适用于它们的最佳的 α 和 β 也可能是不一样的，所以也需要分别调整。这些参数并不是很适合应用一般的训练算法，如果要自动化的话其实也只能进行网格搜索。

3.2 聚类合并

经测试 GSDMM 聚类的结果簇数过多，这样召回率就比较不理想，所以考虑进一步合并聚类结果。

聚类合并部分，先将一个簇中的文本合并为一篇长文本，用一个 TF 向量来表示它，用向量之间的余弦距离来表示相似度，如果相似度超过一定的阈值就合并两个簇。这里同样会有类似上面的问题，即这个方法只适用于一组文本，但是一篇论文中有多篇文本。使用的方法也是一样的，即计算多个相似度，然后按照一定权值加权组合在一起。

最终结果评分在 0.8 左右，多次调整参数也没能达到更好的结果了。

3.3 该方案的不足

测试结果证明这种方法的聚类结果得分不如第一种，这里简单分析一下它有哪些不足之处：

1. 没有使用词向量，对词的利用仅限于等于关系。虽然预处理阶段做了词干化，但这仍然不能利用词义相近但词干不同的两个词的相似性。
2. 原 GSDMM 算法仅限于处理一组文本，我们直接将其加权推广到多段文本，这其实并没有理论基础，而且权值必须手动调整，也很难达到最优的效果。

3. 聚类合并部分也有类似的问题，需要手动调整权值。

4 总结

在本项目中，我们经历了从数据预处理到模型训练的全过程，摸索尝试不同的方法，对冷启动论文同名消歧问题给出了两个解决方案。在这个过程中，我们不仅锻炼与提升了工程能力，而且把数据挖掘课堂中学到的特征提取和聚类的相关知识应用到了解决实际问题上。最后感谢老师与助教为这门课程的付出。

参考文献

[1] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[2] Jianhua Yin and Jianyong Wang. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242, 2014.