Eponymous Author Disambiguation Method Based on Multi-scale and Clustering Properties in Graph Neural Networks

Yuan Fang
CQUPT
Chongqing,China
519623263@qq.com

Hao Deng
CQUPT
Chongqing,China
1922602772@qq.com

Xiangwu Yang
CQUPT
Chongqing,China
2675454740@qq.com

ABSTRACT

The WhoIsWho-IND dataset challenge requires developing a model to identify incorrectly assigned papers for each author profile, including name and publications. The KDD CUP OAG Challenge involves authenticating papers from a large dataset, combining clustering and classification without prior research.

This paper introduce that a graph neural network leveraging multi-scale and clustering to extract and splice diverse features, which are processed through a clustering layer and then classified. Trained on 515 authors, it achieved an AUC of 0.78 on a 370-author test set, outperforming a feature-enhanced tree model by 3% accuracy.

CCS Concepts

• Computing methodologies → Machine learning → Neural networks → Graph neural networks

KEYWORDS

Graph Neural Networks, Multi-Scale,Clustering,Feature-Enhanced Tree Model

ACM Reference Format:
Yuan Fang,Hao Deng,Xiangwu Yang.
Eponymous Author Disambiguation Method Based on Multi-scale and Clustering Properties in Graph Neural Networks. In Proceedings of KDD 2024 Workshop OAG-Challenge Cup (KDDCup’24).
ACM, New York, NY, USA, 4 pages.

* Corresponding author of this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s)
KDDCup’24, Aug 25 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.ACM ISBN 978-1-4503-XXXX-X/18/06

1 Introduction

Academic data mining aims to deepen our understanding of the development, essence, and trends of science, and to tap into its significant value in science, technology, and education. It aids governments in science policy-making, companies in talent discovery, and researchers in efficiently acquiring new knowledge. Despite applications like paper retrieval, expert finding, and journal recommendation, the lack of a data benchmark hinders the progress of academic knowledge graph miningclustering.

2 OAG-Challenge WhoIsWho-IND Task

Given each author's profile, including their name and published papers, participants are required to develop a model to detect papers incorrectly attributed to that author. Additionally, the dataset provides detailed attributes of all involved papers, including title, abstract, authors, keywords, venue, and year of publication.

3 Feature-Enhanced Tree Model

3.1 Data Cleaning

In the given paper data, the data format example is as follows:

Column	Type	Description
ID	string	Paper ID
title	string	Paper title
authors.name	string	Author’s name
author.org	string	Author’s organization
venue	string	Conference or Journal
year	int	Publication year
keywords	list of strings	Key words
abstract	string	Abstract of a paper

For the given data above, there are various languages of characters, such as Chinese, English, Japanese. We first convert all Chinese author names to Pinyin, and convert all uppercase letters to lowercase, and remove noise and stop words from each field.

3.2 Feature Engineering

In the aspect of feature engineering, we have mainly constructed statistical features of the length of various fields, cross-features, year-related features, and related organizational characteristics. At the same time, we have concatenated the processed paper fields and trained a word embedding model for each paper using the Word2Vec model. The aforementioned features are concatenated to obtain the vector representation of the paper.

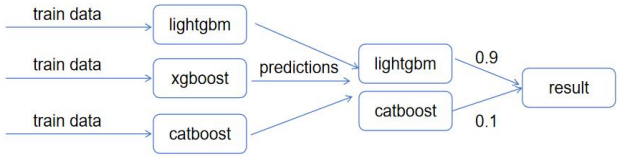
3.3 Tree Model Training

We primarily employ the GBDT model for tree-based algorithms. GBDT utilizes an additive model to iteratively reduce training residuals for classification or

regression tasks. Each iteration develops a weak classifier trained on the gradient of the previous
iteration's output. These weak classifiers are typically simple CART trees with high bias and low variance, which, when summed across iterations, form the final classifier. This approach refines the model's accuracy by incrementally reducing bias.The model can be described as:

$$F_m(x) = \sum_{m=1}^{M} T(x;\theta_m) \quad (1)$$

We construct GBDT tree models through lightgbm, xgboost, catboost, feed the constructed paper data into the model for training, and use 5-fold cross-validation, then through stacking, the predicted results of the above three models are used as features fed into lightgbm and catboost for prediction, and finally, the results obtained by the two models are weighted and averaged, with the final AUC of about 75%.



4 Author Disambiguation Based on Graph Neural Networks

Graph Convolutional Neural Networks (GCNs) are one of the most typical graph neural networks. Graph semi-supervised learning combines convolutional operations with the feature vectors of nodes and the graph structure between nodes. After each graph convolution operation, the feature vectors of the nodes are updated through the graph structure using the feature vectors of neighboring nodes, making similar nodes have similar feature vectors. This process is suitable for the author name disambiguation task, where papers to be disambiguated build a network through mutual associations and continuously update feature vectors through the graph convolutional network to achieve the paper classification task.

To use a graph neural network-based method, it is
necessary to establish the data provided to us as graph data. In the methods of other researchers, there are ways to build heterogeneous graphs, that is, for a cluster of papers with the same author's name, heterogeneous edges are constructed based on the common organizations and co-author relationships in the papers, thereby establishing a heterogeneous information network belonging to the author. Heterogeneous information networks can be represented as $G=\{V,E\}$,G={V,E}, where V and E represent nodes and edges in the network, corresponding to entities and relationships, as shown in Figure 1. Heterogeneous information networks contain various types of nodes (objects) and edges (relationships). Based on the node mapping function ϕ(v) and the edge mapping function ϕ(e), nodes and their complex relationships can be mapped to a low-dimensional space to generate node embedding representations. There are also other researchers who construct multiple graphs for an author according to different edges and input multiple graphs
into the neural network to obtain representations.

In this paper, we did not adopt the above two methods, but used a homogeneous network, that is, we constructed edges in the graph in different ways, but did not make special distinctions for these edges in the graph. Then we built our multi-scale and clustering feature graph neural network for training.

4.1 Establishment of Paper Graph Data Based on Relationships Between Papers

We have defined four types of relationships between papers:

(1) CoAuthor: Each paper usually has more than one author. If there are more than one co-authors between two papers (excluding the author himself), it can be considered that the two papers are related;

(2) CoOrg: If there are more than one common organizations between two papers (excluding the author'sown organization),

(3) CoKeyWord: The keywords of the literature can largely reflect the main content of the research. For two given papers, if there are more than zero common keywords between the two papers, it can be considered that the two papers are related;

(4) CoVenue: If two papers are published in the same journal, it can be considered that the two papers are related

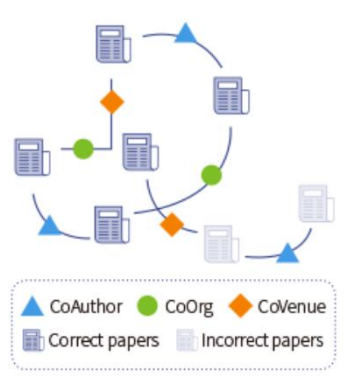


Figure 1 Example of Paper Relationship

4.1.1 Solution to the Problem of Multiple Names for One Person

In the above-mentioned process of discovering co-authors, there is the problem of a person's name having multiple forms of expression. which will seriously affect the discovery of co-authors, thereby affecting the effect of graph construction, resulting in poor graph construction effects. To solve this problem, we have taken the following steps:

1. If it is a Chinese name, convert it to Pinyin;
2. Convert all characters to lowercase strings;
3. Remove all symbols except spaces;

4. Split the string according to spaces;
5. Compare the list of strings after splitting for both strings;
6.If the comparison result is consistent, it is considered to be the same person.

At the same time, the above 5 steps can use edit distance for comparison to achieve better results, but it was abandoned due to high time cost.

4.2 Semantic Representation of Papers Based on BERT Pre-trained Model and Word2Vec

the text features of the paper's title, keywords, abstract, publication name, etc., can be used to characterize the author's research content and distinguish between different authors with the same name. The widely used methods for constructing text vectors include n-gram, NNLM, word2vec, etc. In 2018, Google released the BERT pre-trained language model, which greatly refreshed the accuracy on natural language processing tasks. Subsequently, Beltagy and others launched the SciBERT pre-trained language model specifically trained for scientific papers, which is more suitable for natural language processing tasks of scientific papers. To fully utilize the text features of the paper, this paper takes the title and keywords of the paper as text input and uses the SciBERT model to obtain the semantic representation vector of each paper. The process example is shown in Figure 2.At the same time, on the basis of using BERT, we also used Word2Vec to obtain a unique Word2Vec representation for these papers.

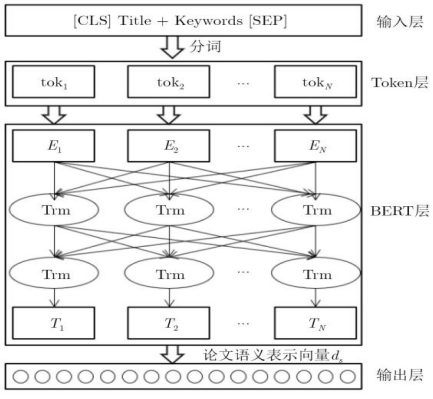


Figure 2 Semantic Representation of Papers Based on BERT Pre-trained Model

4.3 Construction of Neural Network with Multi- scale Features and Clustering Features

The current mainstream graph neural networks used for node classification include SAGE, GAT, SGC,ChebConv,GCNConv,GNNtransformers,etc.However,in the experi-ments for this task, we found that almost any single graph neural network could not achieve good results, so we tried to fuse the features extracted by multiple different neural network layers, using a multi-scale approach, using different graph neural network layers to provide different representations in the graph, and using this representation as the graph's representation.In the extracted graph representation, we designed an MLP layer for classification. In this task, we can actually find that these papers may have n authors. By adding a clustering feature layer, the model can more easily learn which papers are more similar, and thus assign similar papers to an author, making a better binary classification decision.The algorithm flow of the model is roughly shown in the following figure:

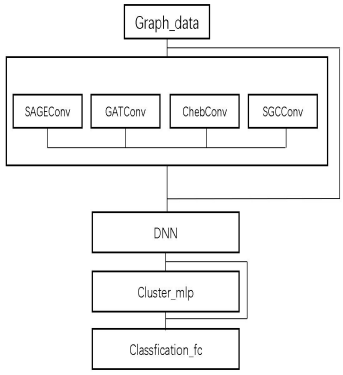


Figure 3 Design of Graph Neural Network with Multi-scale and Clustering Features

Based on the above model structure, using the characteristics of multi-scale and residual, the model can converge in about 5 to 6 epochs. We use AdamW as the learner, with a learning rate of 1e-3, and the final effect can reach about 78%.

4.3.1 Multi-scale graph convolution module

In the graphs we build, the representation of each graph should be different, because each graph represents the relationship between an author to be disambiguated. The connected edges in each graph have the same meaning, but the number and meaning of each connected node are different. In the previous task of node classification, the graph to be classified is often a subgraph of the larger graph, and their nodes have the same meaning, and the connection of the edges also represents the same meaning. However, in our task, the node classification of each graph can be regarded as a separate task, because there is no intersection relationship between any two graphs. Therefore, if only a single feature is extracted from the graph, it is difficult for the model to learn a model that can be truly used for generalization. In this inference, the author tried a variety of separate graph neural networks, such as GATConv, SAGEConv, SGC, to complete the node classification task, but the auc can only reach 0.5+. This shows that the effect of convolution by a single graph neural network cannot accurately represent all the different relationships in the graph. Therefore, in this paper, the author uses multiple different graph neural networks and concatenates the features extracted by them to make

up for the problem that a single graph neural network cannot accurately represent, which has been greatly improved.

4.3.2.Clustering features module

In the eponymous author disambiguation task, there are only two classes for each author with the same name, one is the real author, and the other is the non-real author, which represents other authors with the same name. However, there may be more than one author among the non-real authors, so we can regard the original task as a clustering problem, that is, there may be n different authors in the original task. We need to classify these n authors into two classes based on feature learning. Here are some thoughts on this task.

In the specific implementation, the author uses MLP to classify the features extracted by multi-scale graph neural network at a time n, and concatenate the category features with the previously extracted features and send them to the classification layer. That's a 1% improvement.

5 Experimental Results

5.1Metrics

We adopt the AUC, widely used in anomaly detection, as the evaluation metric.

For each author,

$$Weight = \frac{\#ErrorsOfTheAuthour}{\#TotalErrors} \quad (2)$$

For all authors (M is the number of authors):

$$\text{WeightedAUC} = \sum_{i=1}^{M} \text{AUC}_i \times \text{weight}_i \quad (3)$$

5.2 Source of experimental data

Our experimental data comes from the KDD CUP2024-OAG-Challenge, which includes a training set of 765 authors and over 10,000 papers, and the test set provided by the platform, which includes over 400 authors.

5.3 Experimental Hardware

Our development environment uses Python 3.10, CUDA 12.4,
PyTorch 2.0, with a GPU of RTX 3090 24G

5.4 Experimental Results Analysis

From the experimental results, it can be seen that the effect of our designed network is far better than the ordinary model without multi-scale graph feature extraction, and better than the model with a large number of manually designed features, which shows the superiority of our designed network results. Compared with the best effect of using large model, the effect of 0.8 is not much difference, but it is more memory saving and hardware requirements are smaller, and it has the characteristics of lightweight and simple.

6 Conclusion

In this paper, we design a graph neural network based on multi-scale and clustering characteristics for the disambiguation task of the author of the paper. We also design a set of processes that can be transferred and save memory. Through this set of processes and models, we can achieve good results in this task.

Through comparative experiments, the effectiveness of the components of the clustering layer and the multi-scale graph neural network layer in our model is verified, which provides more modeling ideas for the eponymous author disambiguation task through graph neural network.

The semantic vector acquisition method mentioned in this paper is still insufficient, and a more appropriate semantic vector model can be used to achieve better results.

REFERENCES

[1] ZHOU Qian. Research on disambiguation method of the same author based on Graph neural Network[D].SoochowUniversity2022.DOI:10.27351/d.cnki.gszhu.2022.001962.

[2] WU Xiaona. Research on author disambiguation with the same name based on Multi-feature fusion [D]. Beijing Jiaotong University.2023.DOI:10.26944/d.cnki.gbfju.2023.003304