The Solution for The PST-KDD-2024 OAG-Challenge

Shupeng Zhong
Nanjing University of Science and Technology
Nanjing Shi, China
zspnjlgdx@njust.edu.cn

Xinger Li
Nanjing University of Science and Technology
Nanjing Shi, China

Shushan Jin
Nanjing University of Science and Technology
Nanjing Shi, China

Yang Yang
Nanjing University of Science and Technology
Nanjing Shi, China
yyang@njust.edu.cn

Abstract

In this paper, we introduce the second-place solution in the KDD-2024 OAG-Challenge paper source tracing track. Our solution is mainly based on two methods, BERT and GCN, and combines the reasoning results of BERT and GCN in the final submission to achieve complementary performance. In the BERT solution, we focus on processing the fragments that appear in the references of the paper, and use a variety of operations to reduce the redundant interference in the fragments, so that the information received by BERT is more refined. In the GCN solution, we map information such as paper fragments, abstracts, and titles to a high-dimensional semantic space through an embedding model, and try to build edges between titles, abstracts, and fragments to integrate contextual relationships for judgment. In the end, our solution achieved a remarkable score of 0.47961 in the competition.

ACM Reference Format:
Shupeng Zhong, Xinger Li, Shushan Jin, and Yang Yang. 2024. The Solution for The PST-KDD-2024 OAG-Challenge. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

With the rapid development of science and technology, the number of papers has exploded. For academic papers in various fields, it is very difficult to find the technical development process from a large amount of literature. The source paper tracing task aims to find important reference papers in the original paper, that is, whether they have made inspiring contributions to the ideas or main methods of the paper. At present, the solutions to this task are mainly divided into three categories: large models, graph convolutional neural networks (GCNs), and machine learning. Among them, thanks to the stronger robustness and generalization of large models [24, 21], and the powerful relationship modeling and feature aggregation capabilities of graph convolutional neural networks [10, 22], their excellent performance has attracted more attention from researchers.

Permission to make digital or hard copies of all or part 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 components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

, ,
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Under this background, the KDD-2024 OAG-Challenge PST paper source tracing task requires contestants to design an efficient and accurate algorithm to identify the source paper [27, 28]. Here we will introduce our second place solution.

Our main contributions are as follows:

- For extremely redundant XML format context data, we designed data cleaning methods such as removing XML characters with unclear semantic information, embedding text content prompts and setting dynamic context acquisition length, so that the context used to train BERT-like model will contain richer classification semantic information.
- In terms of GCN, we constructed a relationship network between reference title information, paper title information, paper abstract information, and paper context information, and performed node classification with the help of GCN's powerful relationship modeling capabilities.

Finally, we integrated the reasoning results of BERT and GCN to achieve complementary performance. We believe that the combination of BERT's powerful context understanding ability and GCN's ability to integrate full-text structural information has greatly improved classification accuracy.

2 Related Work

Understanding scientific trends and the flow of ideas is critical to both policy development by funding agencies and knowledge discovery by researchers. Citation relationship analysis reveals the trajectory of scientific evolution, but there is a significant gap between large-scale semantically rich citation relationships and the backbone structure of scientific evolution. To reveal the context in which science develops, we need to simplify citation graphs to track the sources of publications and reveal heuristic relationships between papers.

Tang et al. [11] studied citation semantics, defined three categories of citation links, and constructed a computer science dataset containing approximately 1000 citation pairs. Valenzuela et al. [13] present a new dataset of 450 citation pairs and distinguish between incidental and important citations. Jurgens et al. [7] proposed a larger dataset in the NLP field that contained nearly 2000 citation pairs, but less than 100 citation pairs were annotated as motivations.

Pride and Knoth believe that abstract similarity is one of the most indicative features for predicting citation importance. Hassan et al. [5] used random forest to evaluate the importance of citations and combined context-based and clue word-based features. He et

al. [6] adapted the LDA model to the citation network and developed a new inherited topic model to describe topic evolution. With the rapid development of large models[23, 16, 19, 20, 21], image and text features are constantly being learned by large models[4, 14, 21, 17, 14, 25, 18]. Färber et al. [3] proposed an approach based on convolutional recurrent neural networks to classify potential citation contexts. Jiang and Chen proposed a SciBERT-based contextual representation model to classify citation intent and achieved over 90% AUC on some datasets. Yin et al. [26] designed MRT, an unsupervised framework to identify important previous work or evolutionary trajectories by building an annotated evolutionary roadmap. The framework utilizes generative expression embedding and clustering methods to group publications to capture potential relationships between publications.

3 Method

The training dataset released for this task is mainly given in the form of XML papers. According to our observation, XML data contains a lot of redundant information, so extra attention needs to be paid to data cleaning. Inspired by the [27], we divides the paper tracing task into two aspects: BERT-based text classification task and GCN-based node classification task .

The BERT-like text classification task performs text classification by understanding the cleaned reference context which appears in the original paper. The GCN node classification task focuses on understanding the relationship between the reference title and the paper title based on the content of the reference fragment and the paper abstract.

3.1 Text sequence classification task based on BERT-like model

This section directly regards the paper tracing task as a text sequence binary classification task, that is, the probability that a certain paper is the source paper. This article mainly cleans the XML format training data and does not modify the model module. The official Json file for training contains 788 papers including source papers and reference papers. We only use 2/3 of the papers for training and the remaining 1/3 for verification.

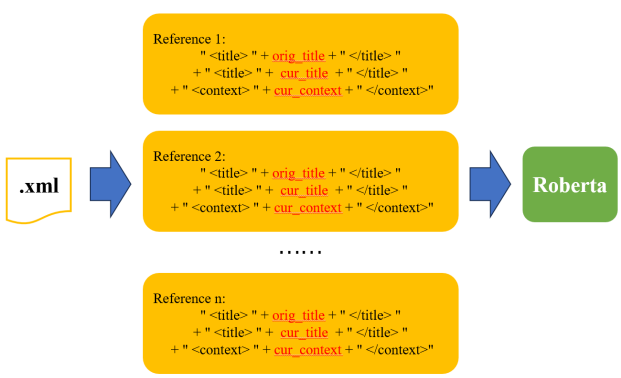

Figure 1: Train Dataset For Roberta

In the text cleaning process, we mainly perform the following operations:

- When finally forming the training data, the original title, the citation title, and the context of the citation in the original paper are inserted in the following format, as show in figure 1.
- The XML format context is cleaned , such as removing line breaks, "<", ">", special characters and other characters whose semantic information is unclear to humans.
- Considering that the same citation appears more than once in the original text, we split the multiple citations by spaces and concatenate them into the final citation context, namely the cur_context part .
- We have added a hint for the reference section, such as "[Introduction]".
- We used two text truncation methods, one is to use semantics to truncate the complete sentence, and the other is to use the absolute number of characters before and after. We found that the combination of the two methods can improve the performance of the model .
- For multiple references appearing in the same segment, we set the references other than the target reference to null.

3.2 GCN-based node classification task

Graph Convolutional Network (GCN) is a neural network model for processing graph structured data. GCN extracts features on the graph through convolution operations, which can effectively capture the relationship between nodes and their neighbors. Unlike traditional convolutional neural networks, GCN is specially designed for graph data and can process non-Euclidean structured data, such as social networks, knowledge graphs, and citation networks.

The core idea of GCN is to aggregate the features of each node with the features of its neighboring nodes through convolution operations, thereby updating the representation of the node. The forward propagation formula of GCN is as follows:

$$H^{(l+1)} = \sigma\left(D^{-1/2}AD^{-1/2}H^{(l)}W^{(l)}\right) \tag{1}$$

where:

- $H^{(l)}$ denotes the node feature matrix at the l-th layer.
- A is the adjacency matrix of the graph.
- D is the degree matrix of the adjacency matrix.
- $W^{(l)}$ is the weight matrix at the l-th layer.
- σ is an activation function (such as ReLU).

Drawing on the research of Yao et al. [8] and Wu et al. [15] on GCN text classification and combining GCN's powerful relationship modeling and feature aggregation capabilities with its excellent efficiency, we used GCN to classify text in the competition.

Figure 2 shows how we use GCN to build graphs for papers. We use the Spacy library to semantically segment each paragraph of the paper, then build a unidirectional edge between the paper abstract and the paper title, a bidirectional edge between the reference title and the paper title, and a unidirectional edge between the reference and the text block where the reference appears. The node

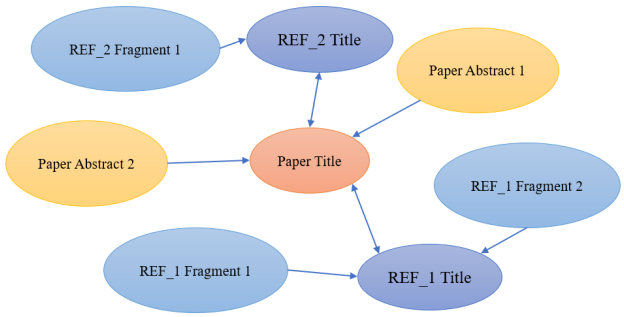

Figure 2: Our GCN Mapping Method

information is a high-dimensional vector of the text processed by the embedding model.

As for the direction of edge building, according to our experiments, the effect of using all bidirectional edge building or all unidirectional edge building is reduced to varying degrees.

4 Experiment

We conducted experiments on GCN and BERT-like models respectively.

4.1 BERT-like models

Here we present our experimental results on BERT-like models. We mainly tried two models: SCIBERT [1, 2] and his variants CS_ROBERTA_BASE [9].

Dataset. The data is provided by the official. The data includes training Json files and a large number of XML format papers. The Json files contain the basic information of 788 papers such as ID, title, references, and source papers; the XML format papers are converted from the PDF files of the original papers using Grobid. It is worth noting that although the official allows the use of OAG and DBLP Citation data, we do not use them.

Details. Considering that the task background is academic papers in various fields, we tried to use models such as SCIBERT and CS_ROBERTA_BASE that have been trained on a large amount of scientific text. Eventually, we found that CS_ROBERTA_BASE achieved the best results on the locally partitioned Train, Val and official Val, Test. We only used one NVIDIA RTX 4090 to train the CS_ROBERTA_BASE model, set the initial learning to 5e-6, batch_size=24, epoch=10 and implemented early stopping after 5 epochs when performance did not continue to improve. Finally, we selected the model with the highest local AP value as the optimal model.

Results. Table 1 shows the performance of our method on the official final test set. We add the method step by step according to the order of the labels in the table. The experimental results fully confirm the effectiveness of the method. The content cleaning of the XML format context removes a lot of redundant information. The embedding of the paper title and the corresponding prompt words makes the model pay more attention to and understand the corresponding content. At the same time, the CS_ROBERTA_BASE model has larger parameters and stronger learning ability.

Table 1: Experimental results of different methods on BERT-like models

Method	Score on test set
SCIBERT	0.38+
+Redundancy removal	0.408+
+CS_ROBERTA_BASE	0.43+
+Title	0.447+
+Content prompt words	0.452+

Table 2: Experimental results of different methods on GNN

Model	Number of Layers	Encoder	Score on test set
GCN	2	BGE_M3	0.447
GCN	5	BGE_small	0.426
GCN	2	BGE_small	0.438
GAT	1	BGE_small	0.440
RGCN	4	BGE_small	0.437

4.2 GCN

Here we mainly introduce our experimental results on GCN. We tried GCN models with various architectures and different text embedding models.

Details. We tried GCN models with different architectures, different numbers of layers, and different numbers of hidden units. We also tried GAT and RGCN based on heterogeneous relational graphs. In terms of experimental results, they seem to be not much different. We first divide the paper into semantic-based chunks with a chunk size of 300 characters. Then these text chunks are mapped to a high-dimensional semantic space as the node information of the GNN through an embedding model. We train our model on only one NVIDIA RTX 4090 and epoch is 100.

Results. The experimental results are shown in Table 2. We found that the main factors affecting the effect of GCN are the way of graph construction and the choice of embedding model. The mapping dimension of the embedding model should not be greater than 1024, and the graph construction should not be too complex. The number of GCN layers should not be too many.

In addition to the information shown in Table 2. We also made other attempts on the size of the text chunk and the way of graph construction. The results show that the size of the text chunk is best between 200 and 300. We also tried to decompose the abstract into more chunks and build edges with more references, but the effect showed a downward trend.

4.3 Ensemble

At the end, we integrated the results from different solutions. Compared with the results of the original single solution, the results were significantly improved, which reflects the advantages of ensemble learning. The integration results are shown in Table 3.

In the final stage, our solution mainly uses the integration of three solutions, including ROBERTA text classification based on semantic segmentation (Roberta-spacy), ROBERTA text classification

Table 3: The results of integrating different solutions in test stage

Method	Score on test set
GCN + Cs-Roberta-spacy	0.4783
GCN + Cs-Roberta-spacy + Cs-Roberta-absolute	0.4796

Table 4: The results of the large language model

stage	Score
Val	0.47+
Test	0.40+

based on absolute position segmentation (Roberta-absolute), and GCN-based text node classification.

5 Other attempts

In addition to the above two solutions based on BERT and GCN, we also tried a solution based on a large language model [12] in the validation set stage. The effect of this solution was reduced in the test set stage, but due to the lack of hardware and time resources, we did not continue to try the large model in the test set stage. In the validation set stage, we input reference fragments and paper abstracts into the large language model, and by modifying the prompts, the large model can break through the original text limitations and output more in-depth text, and then input it into the BERT-like model for training. As shown in Figure 3.

The results of the large model processing are shown in Table 4.

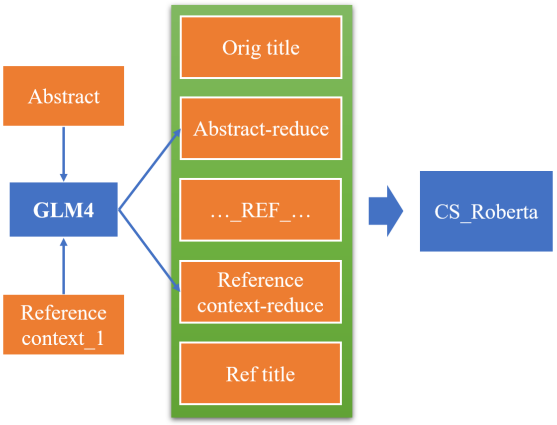

Figure 3: Training a BERT-like model using a large language model-processed dataset

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: a pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676.*

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

[3] Michael Färber, Alexander Thiemann, and Adam Jatowt. 2018. To cite, or not to cite? detecting citation contexts in text. (2018).

[4] Zhongtian Fu, Kefei Song, Luping Zhou, and Yang Yang. 2024. Noise-aware image captioning with progressively exploring mismatched words. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 11. Vol. 38, 12091–12099.

[5] Saeed-Ul Hassan, Anam Akram, and Peter Haddawy. 2017. Identifying important citations using contextual information from full text. (2017).

[6] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. 2009. Detecting topic evolution in scientific literature: how can citations help? (2009).

[7] David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. (2018).

[8] Thomas N. Kipf and Max Welling. 2016. Semi-supervised text classification with graph convolutional networks. (2016).

[9] Yinhan Liu et al. 2019. Roberta: a robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692.*

[10] Yucheng Liu, Zipeng Gao, Xiangyang Liu, Pengfei Luo, Yang Yang, and Hui Xiong. 2023. Qtiah-gnn: quantity and topology imbalance-aware heterogeneous graph neural network for bankruptcy prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1572–1582.

[11] Jie Tang, Jing Zhang, Jeffrey Xu Yu, Zi Yang, Keke Cai, Rui Ma, Li Zhang, and Zhong Su. 2009. Topic distributions over links on web. (2009).

[12] GLM Team et al. 2024. Chatglm: a family of large language models from glm-130b to glm-4 all tools. *arXiv e-prints*, arXiv–2406.

[13] Marco Valenzuela, Vu Ha, and Oren Etzioni. 2015. Identifying meaningful citations. (2015).

[14] Fengqiang Wan, WU Xiangyu, Zhihao Guan, and Yang Yang. [n. d.] Covlr: coordinating cross-modal consistency and intra-modal relations for vision-language retrieval.

[15] Kunze Wang, Yihao Ding, and Soyeon Caren Han. 2023. Graph sample-sigconf-authordraftneural networks for text classification: a survey. (2023).

[16] Yang Yang, Zhao-Yang Fu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. 2019. Semi-supervised multi-modal multi-instance multi-label deep network with optimal transport. *IEEE Transactions on Knowledge and Data Engineering*, 33, 2, 696–709.

[17] Yang Yang, Zhen-Qiang Sun, Hengshu Zhu, Yanjie Fu, Yuanchun Zhou, Hui Xiong, and Jian Yang. 2021. Learning adaptive embedding considering incremental class. *IEEE Transactions on Knowledge and Data Engineering*, 35, 3, 2736–2749.

[18] Yang Yang, Hongchen Wei, Zhen-Qiang Sun, Guang-Yu Li, Yuanchun Zhou, Hui Xiong, and Jian Yang. 2021. S2osc: a holistic semi-supervised approach for open set classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16, 2, 1–27.

[19] Yang Yang, Yi-Feng Wu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. 2018. Complex object classification: a multi-modal multi-instance multi-label deep network with optimal transport. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2594–2603.

[20] Yang Yang, Yi-Feng Wu, De-Chuan Zhan, Zhi-Bin Liu, and Yuan Jiang. 2019. Deep robust unsupervised multi-modal network. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 01. Vol. 33, 5652–5659.

[21] Yang Yang, Jia-Qi Yang, Ran Bao, De-Chuan Zhan, Hengshu Zhu, Xiao-Ru Gao, Hui Xiong, and Jian Yang. 2021. Corporate relative valuation using heterogeneous multi-modal graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 35, 1, 211–224.

[22] Yang Yang, De-Chuan Zhan, Yi-Feng Wu, and Yuan Jiang. 2018. Multi-network user identification via graph-aware embedding. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II 22*. Springer, 209–221.

[23] Yang Yang, De-Chuan Zhan, Yi-Feng Wu, Zhi-Bin Liu, Hui Xiong, and Yuan Jiang. 2019. Semi-supervised multi-modal clustering and classification with incomplete modalities. *IEEE Transactions on Knowledge and Data Engineering*, 33, 2, 682–695.

[24] Yang Yang, Chubing Zhang, Xin Song, Zheng Dong, Hengshu Zhu, and Wenjie Li. 2023. Contextualized knowledge graph embedding for explainable talent training course recommendation. *ACM Transactions on Information Systems*, 42, 2, 1–27.

[25] Yang Yang, Da-Wei Zhou, De-Chuan Zhan, Hui Xiong, Yuan Jiang, and Jian Yang. 2021. Cost-effective incremental deep model: matching model capacity with the least sampling. *IEEE Transactions on Knowledge and Data Engineering*, 35, 4, 3575–3588.

[26] Da Yin, Weng Lam Tam, Ming Ding, and Jie Tang. 2023. Mrt: tracing the evolution of scientific publications. (2023).

[27] Fanjin Zhang, Kun Cao, Yukuo Cen, Jifan Yu, Da Yin, and Jie Tang. 2024. Pst-bench: tracing and benchmarking the source of publications. *arXiv preprint arXiv:2402.16009.*

[28] Fanjin Zhang et al. 2024. Oag-bench: a human-curated benchmark for academic graph mining. *arXiv preprint arXiv:2402.15810.*