Leveraging Hybrid Embeddings and Data Augmentation for Identifying Significant References

Kenji Shinoda
Toyota Motor Corporation
Tokyo, Japan
kenji_shinoda@mail.toyota.co.jp

Abstract

In this technical report, we describe the solution that achieved 5th place in the Paper Source Tracing task of the KDD Cup OAG-Challenge. This task involves estimating the most significant references for each academic paper. We extracted information from the provided XML files and academic databases, and performed named entity resolution using natural language processing to acquire data. We generated features using text embedding and graph embedding techniques. Due to the small data volume, we augmented the dataset through oversampling and trained multiple Gradient Boosted Decision Tree (GBDT) models on hyperparameter tuning. By ensembling the trained models, we produced the prediction results. Finally we achieved a score of 0.44278 on the final submission leaderboard for the Mean Average Precision (MAP) metric, securing 5th place in this task. The sample source code is publicly available at https://github.com/ToyotaInfoTech/kddcup2024-oagpst-solution.

CCS Concepts

• **Applied computing → Document analysis**; • **Computing methodologies → Classification and regression trees**.

Keywords

data mining, academic citation network, text embedding, graph embedding, imbalanced data learning, KDD Cup

ACM Reference Format:
Kenji Shinoda. 2024. Leveraging Hybrid Embeddings and Data Augmentation for Identifying Significant References. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXX.XXXXXX

1 Introduction

Academic papers are published daily on a large scale. According to the DBLP database, the number of published academic papers each year has increased dramatically, from about 14,000 in 1995 to over 550,000 in 2023 [1]. Accordingly, *academic data mining*, which refers to extracting necessary information from large datasets of academic papers based on scientific methods, has become increasingly important. For example, the ArnetMiner system analyzes the results of automatically extracting researcher profiles from the web and provides academic network search services [13].

The KDD Cup Open Academic Graph Challenge (OAG-Challenge) was a competition on academic data mining held at the KDD Cup 2024 [2]. This competition was based on the extensive benchmarking of academic data mining conducted by Zhang et al. [15, 16]. This competition included the following three tasks. In these tasks, the "Paper Source Tracing" (PST) task aims to trace the most significant references in the full texts of given papers.

This competition was held on the data analysis competition platform, biendata, from March 20th to June 7th, 2024. The period until May 31st served as a preliminary validation phase, followed by the final submission period. A total of 236 participants from 203 teams participated, and 20 teams competed in the final submission period. We participated in the PST task and achieved the 5th place result in the final submission phase. Notably, we had the best performance among those who participated solo without forming a team. The main techniques to achieve this result are summarized as follows:

- Information extraction through text mining and querying academic databases for missing information.
- Generation of features using text and graph embedding techniques.
- Data augmentation through oversampling and learning with GBDT models, as well as their ensemble.

In the following chapters, firstly we describe the datasets that were available for PST task. Next, we sequentially explain the methods used for feature processing on the data, and describe the learning methods for the generated feature set. Finally, in the discussion section, we also refer to implementations that did not prove effective during the competition period, such as fine-tuning with LoRA for a local Large Language Model (LLM), and other methods that seem effective for enhancing the result.

2 Dataset

For this task, two types of data were available: datasets prepared by the organizers and academic paper databases that were allowed for external use. Table 1 describes the summary of the data files.

DBLP and OAG are both academic databases that typically contain similar types of data; however, there are instances where records may exist in OAG but not in DBLP. Therefore, I extracted unique paper IDs from OAG and DBLP datasets, and then merged these two datasets. We refer to this merged dataset simply as the "academic database" in this report.

KDD '24, August 25–29,2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXX.XXXXXX

Table 1: Description of data files

Name	Prepared or External	Description
`paper_source_trace_train_ans.json`	prepared	Including paper IDs, their references, and IDs of significant papers for training
`paper_source_trace_valid_wo_ans.json`	prepared	Including paper IDs for validation
`paper_source_trace_test_wo_ans.json`	prepared	Including paper IDs for test
`paper-xml`	prepared	containing XML files of paper
`DBLP and OAG`	external	open-source academic paper databases

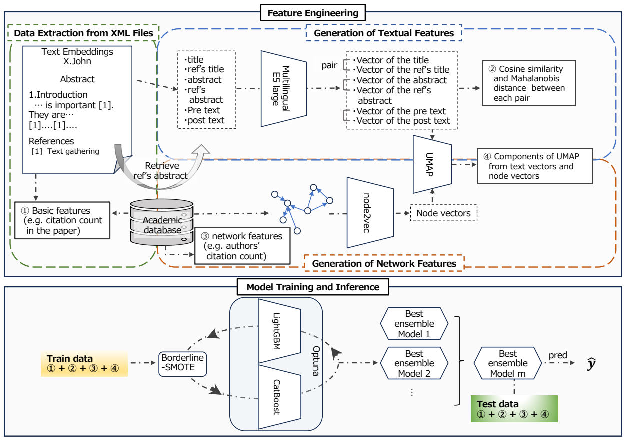

Figure 1: Overview of the processing

3 Feature Engineering

In this section, we explain how to create features for supervised learning. All of the features are listed Table 2, and Figure 1 shows the overall data processing.

3.1 Data Extraction from XML Files

Each XML file not only contains the title and abstract but also the full text, allowing for the extraction of various pieces of information. To create features, we extracted paper ID, title, authors and abstract. In addition, we extracted reference titles, authors, sections, place of citation, citation count and text surrounding the reference citation in the paper.

Since the references in the XML files do not contain abstracts, it was necessary to retrieve them from the academic database. However, since references in the XML files do not have IDs, it was necessary to perform entity resolution based on titles and authors. The entity resolution was conducted as follows:

- Normalize the titles and authors in both the XML file references and the academic database.
- Retrieve records from the academic database that exactly match the reference title.
- In cases where multiple records were linked, calculate the Levenshtein Distance for the authors and link to the record with the shortest distance. The Levenshtein Distance is a metric for measuring two different strings.

This process enabled the retrieval of the reference IDs and abstracts, as well as publication years, page information and publication venues. Since papers can be identified by their IDs, it is also possible to index the academic database by paper ID to similarly retrieve publication years, etc. Based on the information obtained through these processes, simple feature engineering was performed to generate features that are described in the subsequent subsections.

In Table 2,the rows where the Section column is 3.1 correspond to features created in this section.

3.2 Generation of Textual Features

Papers can be influenced by the academic keywords and topics of their references. Text embedding is used to embed texts in a vector space, enabling numerical comparisons and similarity calculations between texts. The multilingual E5 text embedding models are designed to handle text in multiple languages, with the multilingual-e5-large model being the largest, embedding text into 1024 dimensions [14]. It was assumed that the titles and abstracts of the target papers and the significant papers have a high degree of similarity. These texts were embedded using the multilingual-e5-large model, and their similarities were calculated. Additionally, the embedding vectors were dimensionally reduced and used as features. We chose UMAP for dimension reduction, which compresses a high-dimensional vector to a lower-dimensional space while preserving topological structures and has advantages over similar compression techniques like t-SNE in terms of speed [11]. The specific steps are as follows:

- Normalize the extracted text (titles, abstracts).
- Embed the normalized text into 1024 dimensions using the multilingual-e5-large model.
- Calculate the following two distances for the embedded results, the cosine similarity and Mahalanobis distance.
- Use UMAP to reduce the embedding vectors to two dimensions.

In Table 2, rows where the Section column is 3.2 correspond to features created in this section.

3.3 Generation of Network Features

The academic database forms a large network based on citation relationships. While the citation count is a primary feature, it is believed that the citation counts of more significant nodes may increase due to network effects. We considered how to incorporate the effects by quantifying the influence of each node. The academic

Table 2: Main features for machine learning

Section	Name	Dimension	Description
3.1	citation count	2	Number of citations in the academic database for each paper and its references
3.1	citation count in the paper	1	Number of citations within the paper for references
3.1	appearance order in the paper	1	Order in which cited within the paper for references
3.1	appearance order ratio in the paper	1	Ratio of citation's rank to the number of citations
3.1	year difference	1	difference between the paper's published year and its references
3.1	flag of appearance in {e.g. Introduction}	7	Flag of section of the references where cited
3.1	flag of context containing {e.g. important}	5	Flag indicating presence of target phrases in context of the references
3.1	flag of the acceptance by {e.g. CVPR}	12	Flag of accepted conference of the paper and its references
3.2	UMAP components of text embedding vectors	12	Components reduced via UMAP from embedding vector of title, abstract, context
3.3	authors' citation count	1	Number of citations of the authors in the academic database
3.3	UMAP conponents of node2vec vectors	4	Components reduced via UMAP from node2vec embedding vectors
3.2,3.3	cosine similarity and mahalanobis distance	6	between text embedding vectors of the paper and its references

database provides the number of times each paper is cited, based on which the citation counts of the references and the authors' citation counts were calculated. Additionally, a citation network was created from the database, and as a feature representing higher-order relationships, node embeddings were generated using node2vec [6]. Node2vec is an algorithm that explores complex paths between nodes using random walks. This algorithm generates node embeddings based on these paths. After generating the features of each node in a 64 dimensinal space, they were dimensionally reduced to a two dimensional space using UMAP, in a similar way to the text features.

In Table 2, rows where the Section column is 3.3 correspond to features created in this section.

4 Model Training and Inference

We trained machine learning models for supervised binary classification using the generated features, where papers with a strong influence were labeled as positive examples. The models, trained using LightGBM and CatBoost with hyperparameter tuning via Optuna, were used for inference to generate submission files [4, 9, 12]. An average ensemble of the inference results was taken as the final prediction.

The dataset was highly imbalanced for binary classification, with 746 positive and 3583 negative examples. Various strategies exist for handling imbalanced data, such as under-sampling the majority class or over-sampling the minority class. In our solution, negative cases were under-sampled to a fixed amount, and then positive examples were over-sampled to match this number. SMOTE, especially Borderline-SMOTE, was used for over-sampling, which synthesizes new data points near decision boundaries to enhance the model's discriminative power [5, 7]. However, SMOTE can fail to create synthetic data if no neighboring positive samples are available, or if the base sample size is too small to ensure quality. A random seed value was chosen to run over-sampling with Borderline-SMOTE, and if successful, GBDT models were trained. If it failed, a new seed value was selected, and over-sampling was performed again. The performance of the GBDT models, including LightGBM and CatBoost, was optimized by tuning the seed value of Borderline-SMOTE, testing on a prepared dataset to achieve the best AUC. The best performing ensemble model was then used for final predictions, with results averaged for the GBDT models' outputs. Algorithm 1 describes the overall process for training the model.

Algorithm 1 Significant References Classification Method

```
Input: Training dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^F$ is a set of features and $y_i \in \{0, 1\}$ is a class label. Test dataset $\hat{D}$ is as set of features with no class label. A constant value determining the size of negative samples is $a$. The number of trials using different seed is $L$. The number of trials for hyperparameter tuning is $N$.
$D' \leftarrow \{x_i \in D \mid x_i \neq \text{NULL}\}$
$D'_1 \leftarrow \{x_i \in D' \mid y_i = 1\}$
$D'_0 \leftarrow \text{Sample}(D' \mid y_i = 0, \text{size} = a \times |D'_1|)$
$D'' \leftarrow D'_0 \cup D'_1$
$D''_{\text{train}}, D''_{\text{test}} \leftarrow \text{TrainTestSplit}(D'')$
for $l = 1$ to $L$ do
    try: $D^{\dagger} \leftarrow \text{Borderline-SMOTE}(D''_{\text{train}}, \text{random\_state} = \ell)$
    catch: $\text{AUC}_{\ell} \leftarrow 0$, continue
    $m_{\ell}^{\text{LG}} \leftarrow \text{HyperparameterTuning}(\text{LightGBM}(D^{\dagger}), N \text{ trial})$
    $m_{\ell}^{\text{CB}} \leftarrow \text{HyperparameterTuning}(\text{CatBoost}(D^{\dagger}), N \text{ trial})$
    $\text{AUC}_{\ell} \leftarrow \text{Evaluate}((m_{\ell}^{\text{LG}}(D''_{\text{test}}) + m_{\ell}^{\text{CB}}(D''_{\text{test}}))/2)$
end for
$\ell_{\text{best}} \leftarrow \text{argmax}(\{\text{AUC}_{\ell}\}_{\ell=1}^{L})$
$\hat{y} \leftarrow (m_{\ell_{\text{best}}}^{\text{LG}}(\hat{D}) + m_{\ell_{\text{best}}}^{\text{CB}}(\hat{D}))/2$
```

5 Experiments

We created submission file based on the methods described above. The computation environment for data processing, learning, and

Table 3: Experimental results

Method	MAP
Random Forest (Baseline)	0.21420
ProNE (Baseline)	0.21668
SciBERT (Baseline)	0.29489
Our model	**0.44278**

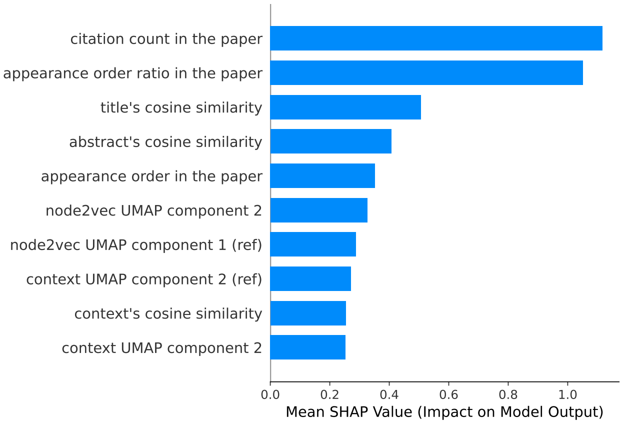

Figure 2: Mean SHAP value

inference was as follows : Linux, Python 3.10, CUDA 12.0, NVIDIA A100 80G.

Using the trained model to predict scores resulted in achieving a score of 0.44278 in the MAP evaluation on the validation and test sets. Table 3 shows that our model significantly improved over the baseline models, Random Forest, ProNE and SciBERT. With SHAP, it is possible to understand which features the trained model places importance on [10]. We show an example of SHAP mean value in Figure 2. This Figure indicates that 'citation count in the paper' of the references was the most important feature in the classification, and the feature sets created through embeddings were among the top performers.

6 Discussion

Due to time constraints during the competition, the following methods, which were not implemented and evaluated, could be effective in further improving the results presented in this report.

- Application of ranking algorithms: In this report, we treated the references that had a strong influence as positive examples in a binary classification problem. However, considering that the evaluation metric is MAP, it is also possible to design the problem using ranking algorithms. Specifically, treating each paper's references as queries, the goal would be to train models such that significant references are ranked higher.
- Finetuning of the local LLM: During the competition, we attempted LoRA training with Microsoft's phi3-medium local LLM [3, 8]. LoRA is a finetuning technique for the local LLM. The training involved inputting paper context information and performing binary classification to determine if a cited reference was influential. Although this did not result in improved the evaluation metric, further improvements could potentially enhance performance.

Acknowledgments

I would like to express my gratitude to Dr.Shintaro Fukushima and Mr.Takeyuki Sasai from Toyota Motor Corporation for their support in participating in this competition.

References

[1] DBLP Statistics. https://dblp.org/statistics/newrecordsperyear.html.

[2] KDD 2024 OAG-Challenge. https://www.biendata.xyz/kdd2024/.

[3] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadallah, Nguyen Bach, Amit Bahree ..., and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL] https://arxiv.org/abs/2404.14219

[4] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2623–2631. https://doi.org/10.1145/3292500.3330701

[5] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *J. Artif. Int. Res.* 16, 1 (jun 2002), 321–357.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 855–864. https://doi.org/10.1145/2939672.2939754

[7] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*, De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 878–887.

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL] https://arxiv.org/abs/2106.09685

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[10] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[11] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3, 29 (2018), 861. https://doi.org/10.21105/joss.00861

[12] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 6639–6649.

[13] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: extraction and mining of academic social networks *(KDD '08)*. Association for Computing Machinery, New York, NY, USA, 990–998. https://doi.org/10.1145/1401890.1402008

[14] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. arXiv:2402.05672 [cs.CL] https://arxiv.org/abs/2402.05672

[15] Fanjin Zhang, Kun Cao, Yukuo Cen, Jifan Yu, Da Yin, and Jie Tang. 2024. PST-Bench: Tracing and Benchmarking the Source of Publications. arXiv:2402.16009 [cs.DL] https://arxiv.org/abs/2402.16009

[16] Fanjin Zhang, Shijie Shi, Yifan Zhu, Bo Chen, Yukuo Cen, Jifan Yu, Yelin Chen, Lulu Wang, Qingfei Zhao, Yuqing Cheng, Tianyi Han, Yuwei An, Dan Zhang, Weng Lam Tam, Kun Cao, Yunhe Pang, Xinyu Guan, Huihui Yuan, Jian Song, Xiaoyan Li, Yuxiao Dong, and Jie Tang. 2024. OAG-Bench: A Human-Curated Benchmark for Academic Graph Mining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.