PST-Bench: Tracing and Benchmarking the Source of Publications

Anonymous ACL submission

Abstract

Tracing the source of research papers is a fundamental yet challenging task for researchers. The billion-scale citation relations between papers can hinder researchers from understanding the evolution of science. To date, there is still a lack of an accurate dataset constructed by professional researchers to identify the direct source of their studied papers, based on which automatic algorithms can be developed to expand the evolutionary knowledge of science. In this paper, we study the problem of paper source tracing (PST) and construct a high-quality and ever-increasing benchmark dataset PST-Bench in computer science. Based on PST-Bench, we also reveal several intriguing discoveries, such as the difference in the life force of papers in different areas (e.g., AI and HPC). An exploration of various methods validates the hardness of PST-Bench, pinpointing potential directions on this topic. The dataset and codes have been available[1].

1 Introduction

The pace of scientific evolution has accelerated like never before. For instance, since the launch of ChatGPT[2] on November 30, 2022, Google Scholar has indexed around 43,000 papers about ChatGPT in less than a year, in the sense that ChatGPT has inspired a significant amount of research works. However, some research works can be traced back to much earlier origins. In distributed systems, Raft (Ongaro and Ousterhout, 2014) is an alternative consensus algorithm proposed for better simplicity and understandability based on Paxos (Lamport, 2001). In computer architecture, temporal prefetcher (Wenisch et al., 2009), conceptually originated from Markov prefetcher (Joseph and Grunwald, 1997), was successfully applied to Arm N2 processor (Pellegrini, 2021) until recently.

[1] https://anonymous.4open.science/r/paper-source-trace-3598

[2] https://chat.openai.com/

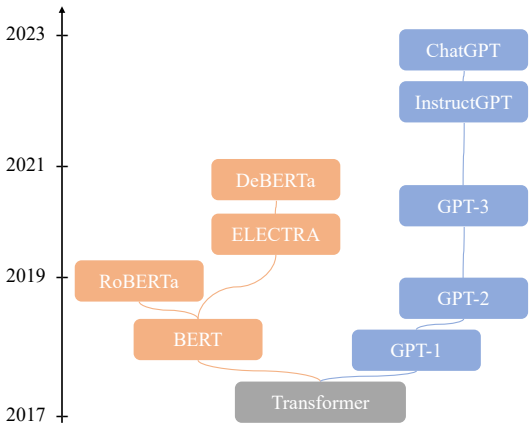


Figure 1: A subgraph of paper source tracing graph.

Tracing the source of research works is a challenging issue that has not been thoroughly studied. Valenzuela et al. (2015) classify citing relationships into incidental and important citations and propose a feature-engineering approach to predict important citations. However, their dataset only contains 450 annotated citing pairs. Algorithm Roadmap (Zha et al., 2019) aims to sketch the dynamics and development of algorithms automatically. It applies weak supervision in the citation contexts to generate datasets and proposes a cross-sentence attention-based model to extract comparative algorithms from texts. Further, MRT (Yin et al., 2023) is an unsupervised framework designed to generate fine-grained annotated evolution roadmaps for specific publications by utilizing text embeddings and node embeddings on citation graphs. MRT evaluates the generated important scores between papers and references based on user clicks on the generated roadmap, which may suffer from the sparsity and bias of user clicks.

Until now, to grasp the ins and outs of technological development from vast literature, it becomes indispensable to trace the source of papers. Otherwise, researchers may find themselves inundated with a multitude of papers and a vast array of references. However, this problem presents the fol-

lowing challenges: (1) How to formally define the source of a paper? (2) How to construct a high-quality and ever-increasing dataset for paper source tracing? (3) What are the underlying patterns behind the paper source tracing graph? (4) Is it feasible to automatically trace the source of papers?

Present Work. For this purpose, in this study, we formally define the problem of paper source tracing (**PST**) and present **PST-Bench**, a professionally-annotated PST dataset comprising 1,120 computer science papers and 49,367 associated references. Each target paper within this dataset has been meticulously annotated with its source papers. Moreover, we conduct a comprehensive analysis of this dataset, uncovering several interesting patterns. Lastly, we investigate the potential for automatically tracing the source of papers. To summarize, our contributions are as follows.

- We establish an accurate, diverse, and continually expanding paper source tracing dataset **PST-Bench**. To achieve this, we devise reward and punishment mechanisms to encourage graduate students to annotate the source of papers accurately and regularly.

- We perform an in-depth analysis of the PST graph based on PST-Bench, with an illustrative subgraph provided in Figure 1. For instance, our analysis uncovers the existence of the Matthew effect within the PST problem, indicating that a small number of source papers can significantly influence a vast array of subsequent works. Interestingly, the temporal gap between a paper and its source papers highlights differences among various subfields within computer science. For example, papers in high performance computing tend to draw inspiration from older papers, while the case is reverse for AI papers.

- We explore a range of approaches to automatically trace the source of papers, including statistical methods, graph-based methods, pre-trained content-based methods, and ensemble methods. Experimental results indicate that pre-trained language models (PLMs) exhibit the potential for addressing the PST problem. However, the best result of automatic methods is still far from satisfactory, leaving much room for future research, including long text understanding, the integration of PLMs and graph-based methods, and so forth.

2 Related Work

Paper source tracing is closely related to citation intention analysis, trend analysis, and citation impact evaluation, among others. The creation of a scalable benchmark dataset that quantifies and annotates the semantics of citation links presents a significant challenge. Tang et al. (2009) conduct a study on citation semantic analysis, defining three categories for each citation link: drill down, similar, and others. They construct a dataset comprising approximately 1,000 citation pairs in computer science. Hereafter, Valenzuela et al. (2015) propose a new dataset of 450 citation pairs designed to classify incidental and important citations. Jurgens et al. (2018) introduce a larger dataset of nearly 2,000 citation pairs in the NLP area, classifying citation intentions into categories such as background, uses, motivation, and comparison. Most of these datasets involve meticulous annotation of each paper, comparing one target paper with each reference, thus making them hard to scale up.

Some endeavors have been made to automatically identify the importance of references. Early attempts define hand-crafted features and then employ classifiers to determine the significance of references. Pride and Knoth (2017) argue that abstract similarity is one of the most predictive features. Hassan et al. (2017) incorporate several new features, such as context-based and cue words-based features, and utilize Random Forest to assess the importance of references. He et al. (2009) adapt the LDA (Blei et al., 2003) model to citation networks and develop a new inheritance topic model to depict the topic evolution. Färber et al. (2018) present a convolutional recurrent neural network based method to classify potential citation contexts. Jiang and Chen (2023) propose contextualized representation models based on SciBERT (Beltagy et al., 2019) to classify citation intentions. The predictive performance is optimistic on certain datasets, achieving over 90% AUC.

Paper source tracing has numerous practical applications, including understanding the evolution of a subfield (Shao et al., 2022) and assessing scholarly impact. Several online systems, such as MRT (Yin et al., 2023) and IdeaReader (Li et al., 2022), have been developed to assist researchers in better understanding the evolution of ideas or concepts. Characterizing important references enables a better evaluation of scholarly impact. Manchanda and Karypis (2021) propose CCI, a content-aware

citation impact measure, to quantify the scholarly impact of a publication.

In this study, we build an accurate and scalable benchmark PST-Bench for paper source tracing and investigate a variety of methods for automatic source tracing. Extensive experiments underscore the complexity of the task, which deserves more in-depth exploration in the future.

3 Problem Definition

In this section, we formally define the problem of paper source tracing (PST).

Problem 1 ***Paper Source Tracing (PST)****. Given a target paper p along with its full text, the objective is to identify the most important references, termed as "ref-sources", that have significantly contributed to the ideas or methods presented in the paper. For each reference within the paper p, an important score ranging from 0 to 1 should be assigned, indicating the degree of influence each reference has exerted on the paper. For each paper p, the predictive output is denoted as S_p.*

Note that a paper may draw inspiration from one or more *"ref-sources"*. The determination of whether a reference qualifies as a *"ref-source"* is based on one of the following criteria:

- Does the main idea of paper p draw inspiration from the reference?
- Is the fundamental methodology of paper p derived from the reference?

Namely, is the reference indispensable to paper p? Without the contributions of the reference, would the completion of paper p be impossible? It's vital to clarify that if paper p_c cites both papers p_a and p_b, with p_a serving as a *ref-source* for p_b and p_b in turn serving as a *ref-source* for p_c. In this case, p_a does not become a *ref-source* for p_c, even if p_c cites p_a. Our focus is solely on identifying *ref-sources* that **directly** inspire paper p.

4 Building the PST-Bench

Considering the specialized knowledge necessary for tracing the sources of academic papers, we engaged dozens of computer science graduate students to identify the sources of English papers within their respective fields of expertise.

Our data collection methodology is bifurcated into two approaches. The first approach involves each student marking the papers they had previously read, averaging around 20 papers per individual. To ensure a consistent influx of high-quality labeled data, the second approach requires each student to read and mark two new papers every week. This is conducted in the format of an *online WeChat paper reading group*, where students identify the source papers of the ones they read recently. A data collection example is shown in Figure 2. More specifics about data collection can be found in Section A.

A Filling Example

Title: Masked Autoencoders Are Scalable Vision Learners
Venue: CVPR 2022
Reading notes: This paper introduces an asymmetric encoder-decoder structure to reconstruct the original image by masking a significant portion of input image patches (e.g., 75%).
Ref-sources: BERT: Pre-training of deep bidirectional transformers for language understanding###An image is worth 16x16 words: Transformers for image recognition at scale
First Author: Kaiming He
Affiliation: Meta
Paper Field: computer vision
Keywords: image classification###self-supervised learning
Your Name: ZZZ
Date filled in: 20230606

Figure 2: A filling example. Multiple items are separated by "###" in the fields of *ref-sources* and keywords.

After gathering and preprocessing the data, we obtain a total of 1,120 labeled computer science papers. The dataset is then partitioned based on their publication year, with 560 papers allocated for training, 280 for validation, and the remaining 280 set aside for testing.

Quantity control & quality control. We devise several strategies to ensure a steady and quality growth of the dataset. First, each student only needs to read and mark two new papers every week, avoiding the attacks of perfunctory annotations to some extent. Second, we provide additional accumulated rewards to students once they have read and marked a certain number of papers (e.g., 20), and remove students who have not marked any papers for a long time, thereby improving long-term user retention. Third, we conduct both automatic and manual quality control on the labeled data, including verifying the existence of citation relationships between *ref-sources* and target papers and manually checking the rationality of the annotations.

Human evaluation. Senior researchers double-checked 100 papers in the test set and tried to identify those papers that were clearly annotated incorrectly. The sampled correct rate is 94%.

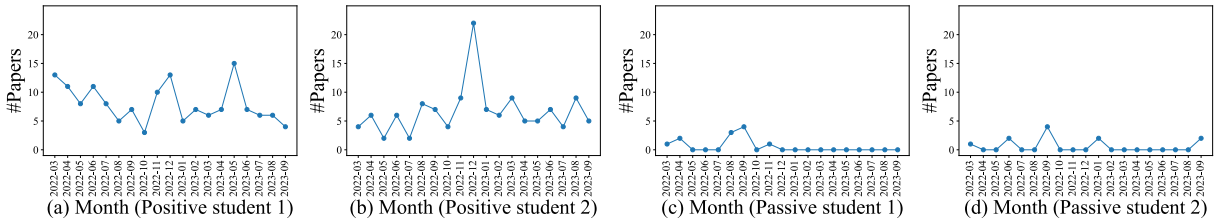


Figure 3: Positive and passive student patterns.

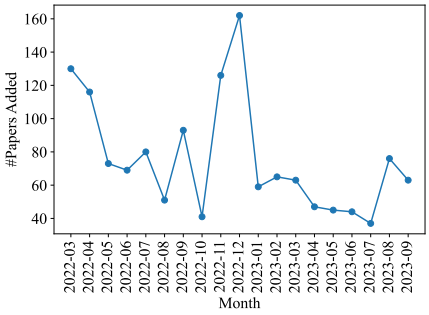


Figure 4: New papers added per month in the paper reading group.

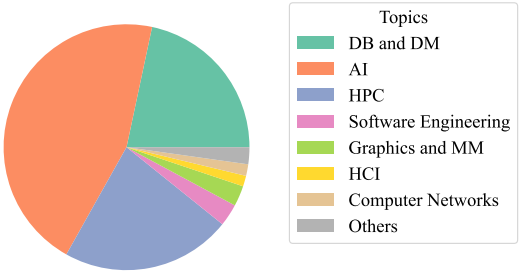


Figure 5: Paper topic distribution. DB and DM: Database and Data Mining, AI: Artificial Intelligence and Pattern Recognition, HPC: High Performance Computing, Graphics and MM: Computer Graphics and Multimedia, HCI: Human Computer Interaction and Pervasive Computing.

5 Preliminary Study

5.1 Student Behavior Patterns

The paper reading group was established in March 2022, running for around one year and a half until now. We track the number of new papers added to the dataset each month, as depicted in Figure 4. Several observations can be made below. (1) Students were actively reading and sharing papers when the group was just created, particularly in March and April 2022. After this initial period, the number of papers added in most months was less than those added in March/April 2022. (2) The number of newly added papers peaked in November and December 2022. The reason is two-fold. On the one hand, we additionally rewarded the students who had shared at least 20 papers in November 2022, which likely motivated more paper sharing among some students. On the other hand, we publicized our paper reading group in October 2022 and removed inactive students in November 2022. One needed to read and share new papers to prevent being removed. (3) The number of newly added papers tended to decrease during major holidays, such as the Chinese New Year in February 2023 and the National Day in October 2022.

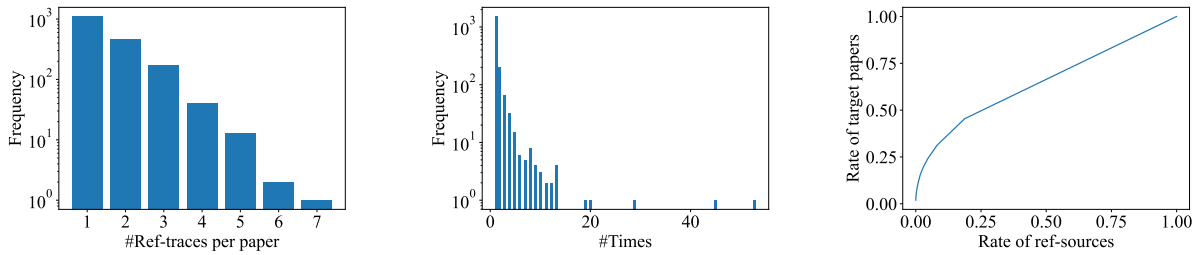
We also conduct an individual analysis for students in the paper reading group. Figure 3 illustrates the patterns of positive and passive students. For positive students who regularly shared and read papers, Figure 3(a) depicts a student who steadily shared new papers with slight variance, while the positive student in Figure 3(b) shared the most papers in December 2022, potentially motivated by the accumulated reward mechanism. Figure 3(c) and Figure 3(d) represent two passive students. The student in Figure 3(c) actively shared papers for a short period but lost interest subsequently. Figure 3(d) presents an interesting pattern. This student shared papers only occasionally. Instead, we observed that (s)he gave red packets to group members proactively and commonly when not reading papers. It implies that (s)he viewed the paper reading group as an incentive mechanism to motivate one's reading habit.

5.2 Paper Statistics and Patterns

Paper topic distribution. Figure 5 visualizes the topic distribution of the collected papers, which are categorized into 8 subtopics[3]. This figure reveals that the majority of papers fall within the AI field, followed by *high performance computing (HPC)* and *database and data mining*. This distribution is largely due to the fact that our paper reading group initially expanded from students in the HPC and AI groups. Collected papers cover diverse fields but are short of areas of *network and information security*, *theoretical computer science*, and *system software and software engineering*.

Paper source tracing graph (PST Graph). The

[3] https://numbda.cs.tsinghua.edu.cn/~yuwj/TH-CPL.pdf



(a) Distribution of the number of *ref-sources* per paper.

(b) Frequency of a paper being regarded as *ref-sources*.

(c) Cumulative distribution between *ref-sources* and target papers.

Figure 6: Analysis of the distribution of *ref-sources*.

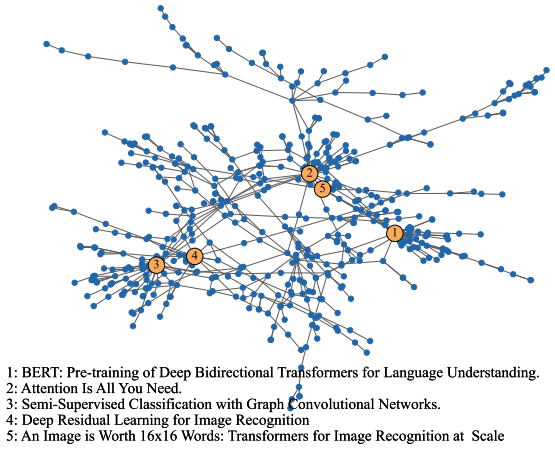


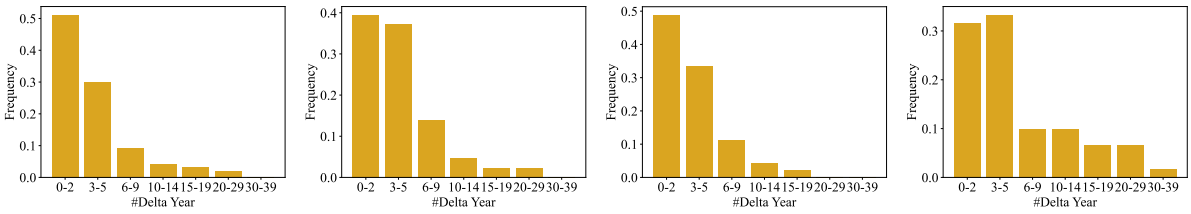
Figure 7: Paper source tracing graph. Papers with more than 100 citations are plotted. The edges represent the relations between papers and their *ref-sources*. The five nodes with the largest degree are enlarged.

PST graph, denoted as $\mathcal{G}_{\text{pst}} = \{\mathcal{P}, \mathcal{E}\}$, consists of a paper set $\mathcal{P}$ and edge set $\mathcal{E}$. Each edge $e \in \mathcal{E}$ represents the relations between one paper and its *ref-sources*. For better visualization, we plot the largest connected component of the PST graph, including paper nodes with over 100 citations, in Figure 7. We discover that papers are scattered in several "communities", each containing a "super node". This figure vividly illustrates the research threads of several fields in computer science. For instance, on the right, Transformers (Vaswani et al., 2017) and BERT (Devlin et al., 2019) inspired a significant body of pre-training works, including ViT (Dosovitskiy et al., 2020). ViT, in turn, inspired numerous research works in computer vision. On the left, graph convolutional networks (GCN) (Kipf and Welling, 2017) and ResNet (He et al., 2016) are two pioneering works that inspired a lot of studies in graph mining and deep learning.

***Ref-sources* per paper.** Figure 6(a) presents the histogram of the number of *ref-sources* per paper. It demonstrates that most annotated papers have only one *ref-source*, with about 10% of papers having more than three *ref-sources*. This could reflect the actual distribution of *ref-sources* per paper to some extent, but may also be caused by the annotation bias among students, who may be more inclined to annotate papers with fewer *ref-sources*.

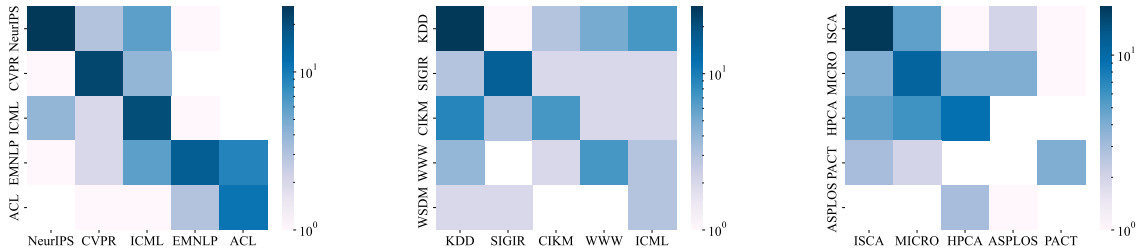
Matthew effect of *ref-sources*. Figure 6(b) and Figure 6(c) display the frequency of a paper being considered as a *ref-source* and the cumulative distribution between *ref-sources* and target papers, respectively. We observe that the majority of papers are regarded as *ref-sources* only **once** in our dataset, while only a few dozen papers are regarded as *ref-sources* more than 10 times. In Figure 6(c), the rate of *ref-sources* is sorted by the times of a paper being treated as a *ref-source*. We observe that the top 20% of papers inspire more than 40% of other papers, and the top 40% of papers inspire about 60% of papers. Papers ranked in the bottom 20% largely maintain a one-to-one mapping with their *ref-sources*, demonstrating the diversity of related research as well as our datasets.

How soon will one *ref-source* inspire subsequent works? We examine the year gap between a paper and its *ref-sources* across different fields. Figure 8 shows the distribution of the year gap in four fields with the most papers. We have the following intriguing observations. (1) Across all studied fields, most papers are inspired by *ref-sources* published within the past 5 years. Papers are less likely to be influenced by older publications. (2) There exist clear differences between fields in terms of the distribution of the year gap. For example, in HPC and computer graphics, roughly the same order of magnitude of papers are inspired by papers from 0-2 years ago and papers from 3-5 years ago. However, in AI and *database and data mining*, almost



(a) AI (b) Computer Graphics (c) Database and Data Mining (d) HPC

Figure 8: Year gap between a paper and its *ref-sources* in different fields.



(a) AI (b) Database and Data Mining (c) HPC

Figure 9: Influence between computer science venues.

half of papers are inspired by papers from 0-2 years ago. Some HPC papers are even inspired by papers published more than 30 years ago, a phenomenon rarely seen in other fields. It reveals that some areas, such as AI, are developing rapidly, while for fields such as HPC, papers in these fields tend to have a relatively longer life force.

Influence between computer science venues. For target venues in each subtopic, we study *ref-sources* in which source venues are more likely to inspire papers in target venues. We count pairwise influence relationships between venues, selecting the subtopics with the most annotated papers, including AI, database and data mining, and HPC. For each subtopic, we select the top-5 target venues with the most papers and top-5 source venues that inspired most papers in target venues. Figure 9 displays the heatmaps of pairwise venue influence on these subtopics. We highlight several observations below. (1) AI venues are mostly influenced by AI venues. NLP conferences (e.g., ACL and EMNLP) can be influenced by ML conferences (e.g., ICML), but the reverse is not the case. (2) In addition to being affected by data mining (DM) conferences, DM conferences are also influenced by AI conferences (e.g., ICML). (3) HPC conferences are primarily influenced by HPC conferences. These figures clearly demonstrate the cross-influence between different fields in computer science.

6 PST Approach

With the vast proliferation of research papers, manually annotating the source of each paper is impractical. Can we automatically identify the *ref-sources* of a paper? In this section, we explore various approaches to address the PST problem. PST approaches can be broadly categorized into the following classes: (1) statistical methods, (2) graph-based methods, (3) pre-trained content-based methods, and (4) ensemble methods.

6.1 Statistical Methods

Rule. An intuitive method to discover *ref-sources* is the rule-based method, which extracts references that appear near signal words like "motivated by" or "inspired by". Nevertheless, a limitation of this method is that not all *ref-sources* are explicitly mentioned in proximity to these signal words.

Random Forest (RF). Alternatively, we can define statistical features related to each reference to indicate its importance. Following (Valenzuela et al., 2015), we define features including citing count, citing position, author overlap, text similarity, etc. We then employ RF to classify the importance of each reference. RF is adopted due to its effectiveness in filtering out unrelated features.

6.2 Graph-based Methods

The paper citation graph can also deliver the structural importance or structural similarity of each reference to the target paper. For instance, an extension paper p_e and its original paper p probably share many references. Thus, their structural similarity should be high. To this end, we extract the paper citation graph in computer science[4] and learn paper embeddings with network embedding methods, such as **LINE** (Tang et al., 2015), **ProNE** (Zhang et al., 2019), **NetSMF** (Qiu et al., 2019). We adopt these methods owing to their effectiveness and efficiency in handling large-scale graphs. Next, we measure the importance of references to the target paper by calculating the cosine similarity between the paper representation and the reference representation.

6.3 Pre-trained Content-based Methods

Imagine how researchers judge whether a reference is a *ref-source*. They may read the context where the reference appears in the full text of the paper and then decide whether the reference is a *ref-source* based on content comprehension. Recently, pre-trained language models (PLMs) have achieved great success in various natural language understanding tasks. Hence, we can extract the contextual texts where each reference appears in the full text and then encode these texts with the pre-trained models, which are then followed by an MLP classifier for binary prediction. We use the annotation results in the training set as supervision information to fine-tune the parameters of pre-trained models and the classifier layers. Then, fine-tuned models are used to predict the *ref-sources* of papers in the test set. The considered PLMs include BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019), and GLM (Du et al., 2022). For comparison, we also evaluate these pre-trained models without any fine-tuning.

6.4 Ensemble Methods

To leverage the strengths of each category of methods, we employ an ensemble method to combine the predictions of different methods. Specifically, we select the best performer from each category of methods and average their predictions as the final prediction. We opt for average instead of voting to avoid specifying thresholds for each method.

[4] https://www.aminer.cn/citation

Table 1: Accuracy results of paper source tracing.

	Method	MAP
Stat	Rule	0.0565
	RF	0.1268
Graph	LINE	0.1140
	ProNE	0.1273
	NetSMF	0.1364
PLM	BERT-base	0.1418
	SciBERT	0.1220
	GLM-2B	0.0961
	GLM-10B	0.0754
PLM-FT	BERT-base	0.1294
	SciBERT	0.2634
	GLM-2B	0.1465
	GLM-10B	0.1558
Ensemble	RF + NetSMF + SciBERT-FT	**0.2709**

Stat: statistical methods, PLM: pre-trained language models, PLM-FT: fine-tuned PLM.

7 Experiments

7.1 Experimental Setup

For the full texts of papers, we use the GROBID[5] API to convert PDF to XML format for convenient processing of citation contexts. We employ regular expression to identify the contexts of each reference. For graph-based methods, the node embedding size is set to 128. We utilize the CogDL (Cen et al., 2023) framework to implement graph-based methods. For pre-trained content-based methods, the context length is set to 200. More implementation details can be found in Section B.

Evaluation Metrics. We adopt mean average precision (MAP) to evaluate the prediction results. Concretely, for each paper p in the test set,

$$\mathrm{AP}(p) = \frac{1}{R_p} \sum_{k=1}^{M_p} \mathrm{Prec}_p(k) \mathbb{1}_k, \tag{1}$$

where R_p is the number of *ref-sources* of paper p, M_p is the number of references of paper p, $\mathrm{Prec}_p(k)$ is the precision at cut-off k in the ranked output list $S_p(k)$, and $\mathbb{1}_k$ is the actual annotation, with the values 0 or 1.

$$\mathrm{MAP} = \frac{1}{|\mathcal{P}_{\text{test}}|} \sum_{p \in \mathcal{P}_{\text{test}}} \mathrm{AP}(p), \tag{2}$$

where $\mathcal{P}_{\text{test}}$ is the paper set in the testing set.

[5] https://grobid.readthedocs.io/en/latest/

Table 2: The feature contribution analysis for RF.

Feature description	Weight
citation number of the reference	0.48
reciprocal of the number of references	0.26
number of paper citations / all citations[1]	0.17
appearing near signal words[2]	0.02
author overlap[3]	0.02

[1] This feature computes the number of direct citation instances for the cited paper over all the direct citation instances in the citing work.
[2] Signal words include "inspired by" and "motivated by".
[3] Set to true if the citing and the cited works share at least one common author.

7.2 Main Results

Table 1 presents the results of paper source tracing. Among all the methods evaluated, Random Forest (RF) surpasses the Rule method, emphasizing the efficacy of feature engineering. NetSMF outperforms LINE and ProNE, likely due to its ability to capture higher-order proximity of nodes via sparse matrix factorization. The Rule-based approach underperforms, likely due to the absence of signal words such as "inspired by" around many crucial references, leading to a low recall rate. Notably, NetSMF performs comparably to several fine-tuned pre-trained models without utilizing supervision information, underscoring the importance of the citation network structure. Fine-tuned SciBERT significantly surpasses other single models, demonstrating the effectiveness of pre-training on domain-specific data. Fine-tuning BERT impairs the performance, possibly owing to the mismatch between pre-trained models and the target tasks. The ensemble method achieves the best performance, indicating that each category of methods has its unique advantages for this problem. However, the current methods' results are not yet optimal, suggesting significant potential for further research in this field.

7.3 Feature Analysis

We conduct a feature importance analysis for random forest, with the most significant features shown in Table 2. We observe that the most important feature is the citation number of the reference, aligning with our previous analysis. In addition, the number of direct citations of a reference also matters, which makes sense as the more times a reference is cited, the more important it might be. Surprisingly, the feature of appearing near signal words is not that important, possibly due to the sparsity of this feature. Author overlap is weakly positively correlated with being a *ref-source*, which is intuitive since some authors are likely to extend the ideas or methods from their previous works.

Target Paper 1: ProteinBERT: A universal deep-learning model of protein sequence and function
Ref-source 1: Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences
Contexts: … loss continues to improve on the training set (i.e., does not saturate), even after multiple epochs (Fig. 2), in accordance with other studies [20].

Target Paper 2: PeCo: Perceptual Codebook for BERT Pre-training of Vision Transformers
Ref-source 2: The unreasonable effectiveness of deep features as a perceptual metric.
Contexts: It has been shown in [71] that the internal activations of a network trained for classification task surprisingly coincide with human judgment.

Target Paper 3: xMoCo: Cross Momentum Contrastive Learning for Open-Domain Question Answering
Ref-source 3: Momentum contrast for unsupervised visual representation learning
Contexts: Momentum contrastive learning (MoCo) is originally proposed by He et al. (2020). He et al. (2020) learns …

Figure 10: Predictive error analysis.

7.4 Error Analysis

We conduct a case study of the prediction errors made by our best-performing model, with several examples shown in Figure 10. We list each target paper with its *ref-source* and the corresponding contexts. We have the following observations. For target paper 1, the relationship between the target paper and its *ref-source* is weak, as indicated by the signal words "in accordance with", making it hard to identify the *ref-source* based on the context. For target paper 2, the *ref-source* appears as a background explanation of the target paper, resulting in a loose semantic correlation between them. For target paper 3, the *ref-source* is introduced in the related work section and is not explicitly associated with the target paper. However, familiar researchers can easily identify the *ref-source* based on the title similarity of the two papers. Thus, the general understanding of main ideas of papers might be omitted in the current contextual methods.

8 Conclusion

In this paper, we present PST-Bench, a novel, professionally-annotated, and ever-growing benchmark for paper source tracing. PST-Bench enables further analysis of the evolution of science and a deep understanding of the crux of research works, and so on. Through extensive experiments, we highlight that the PST-Bench presents significant challenges for existing machine learning methods, pointing out potential directions of lengthy text understanding and citation graph structure mining.

9 Ethical Considerations

For online publications, PST-Bench provides publicly available metadata and very few parsed full-texts of open-access papers for research purposes. For data annotation, all annotators gave their informed consent for inclusion before they participated in this study.

10 Limitations

While PST-Bench provides an accurate and scalable benchmark for paper source tracing, its current format has the following limitations. First, the topics covered in PST-Bench are not even, with most topics related to AI, data mining, and high performance computing. Second, annotating the source of papers is subjective to some degree. Different readers may hold different views on selecting *ref-sources* for the same paper. This might be alleviated by cross-checking from different readers, but sometimes identifying the source of a paper may be an open question with no standard answer. Third, annotators might tend to annotate fewer *ref-sources* than actual ones, which is deferred to future work by cross-checking from multiple annotators.

11 Broader Impact

PST-Bench can be used by various communities, such as NLP, graph mining, science of science, etc. One can use them to discover the evolution of science or develop automatic methods to trace the source of papers. However, since there may be no standard answer for the sources of some papers, users can leverage PST-Bench dialectically.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Yukuo Cen, Zhenyu Hou, Yan Wang, Qibin Chen, Yizhen Luo, Zhongming Yu, Hengrui Zhang, Xingcheng Yao, Aohan Zeng, Shiguang Guo, et al. 2023. Cogdl: A comprehensive library for graph deep learning. In *Proceedings of the ACM Web Conference 2023*, pages 747–758.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Michael Färber, Alexander Thiemann, and Adam Jatowt. 2018. To cite, or not to cite? detecting citation contexts in text. In *Advances in Information Retrieval: 40th European Conference on IR Research*, pages 598–603.

Saeed-Ul Hassan, Anam Akram, and Peter Haddawy. 2017. Identifying important citations using contextual information from full text. In *2017 ACM/IEEE Joint Conference on Digital Libraries*, pages 1–8.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. 2009. Detecting topic evolution in scientific literature: how can citations help? In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 957–966.

Xiaorui Jiang and Jingqiang Chen. 2023. Contextualised segment-wise citation function classification. *Scientometrics*, 128(9):5117–5158.

Doug Joseph and Dirk Grunwald. 1997. Prefetching using markov predictors. In *Proceedings of the 24th annual international symposium on Computer architecture*, pages 252–263.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Leslie Lamport. 2001. Paxos made simple. *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pages 51–58.

Qi Li, Yuyang Ren, Xingli Wang, Luoyi Fu, Jiaxin Ding, Xinde Cao, Xinbing Wang, and Chenghu Zhou. 2022. Ideareader: A machine reading system for understanding the idea flow of scientific publications. *arXiv preprint arXiv:2209.13243*.

Saurav Manchanda and George Karypis. 2021. Evaluating scholarly impact: Towards content-aware bibliometrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6041–6053.

Diego Ongaro and John Ousterhout. 2014. In search of an understandable consensus algorithm. In *2014 USENIX annual technical conference*, pages 305–319.

Andrea Pellegrini. 2021. Arm neoverse n2: Arm's 2 nd generation high performance infrastructure cpus and system ips. In *2021 IEEE Hot Chips 33 Symposium*, pages 1–27.

David Pride and Petr Knoth. 2017. Incidental or influential?-challenges in automatically detecting citation importance using publication full texts. In *Research and Advanced Technology for Digital Libraries: 21st International Conference on Theory and Practice of Digital Libraries*, pages 572–578.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. 2019. Netsmf: Large-scale network embedding as sparse matrix factorization. In *The World Wide Web Conference*, pages 1509–1520.

Zhou Shao, Ruoyan Zhao, Sha Yuan, Ming Ding, and Yongli Wang. 2022. Tracing the evolution of ai in the past decade and forecasting the emerging trends. *Expert Systems with Applications*, 209:118221.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077.

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. In *2009 Ninth IEEE International Conference on Data Mining*, pages 1010–1015.

Marco Valenzuela, Vu Ha, and Oren Etzioni. 2015. Identifying meaningful citations. In *AAAI workshop: Scholarly big data*, volume 15, page 13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Paper Reading Group Rules

Each student needs to read **2** papers every week. After reading, you need to share reading notes and fill in relevant info on the form.

Check mechanism:
Reading notes will be checked by group members and programs to check whether ref-source is authentic.

Punishment mechanism:
Students who didn't share their notes last week need to give ￥2*Y red packets to those who completed paper sharing. Students who didn't share papers for four weeks will be removed from the reading group.

Reward mechanism:
Students who added a new qualified unique paper can receive ￥Y rewards. For every 20 valid papers for each student, (s)he will receive an additional ￥20*Y reward.

Statement:
The collected data will be public for research purposes only.

Figure 11: Reading group rules.

Thomas F Wenisch, Michael Ferdman, Anastasia Ailamaki, Babak Falsafi, and Andreas Moshovos. 2009. Practical off-chip meta-data for temporal memory streaming. In *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, pages 79–90.

Da Yin, Weng Lam Tam, Ming Ding, and Jie Tang. 2023. Mrt: Tracing the evolution of scientific publications. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):711–724.

Hanwen Zha, Wenhu Chen, Keqian Li, and Xifeng Yan. 2019. Mining algorithm roadmap in scientific publications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1083–1092.

Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019. Prone: fast and scalable network representation learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4278–4284.

A Data Collection

The detailed paper reading group rules are shown in Figure 11. Currently, each paper is annotated by one student. We periodically hold paper reading groups on WeChat every week and publicize the reading group on the public forums of several universities and familar labs. The recruited students usually read papers even without the reading group. Thus, their workload is primarily to annotate the source of papers they have read and fill in the form we provide. In this case, the payment is relatively reasonable. We don't know many demographics of volunteering students, but most of them are from China, studying in universities or research institutes, including Tsinghua University, Chinese Academy of Sciences, Harbin Institute of Technology, Southeast University, Nankai University, etc.

Table 3: Parameters and running time of main methods.

Method	#Parameters	Running hours
RF	12	0.05
LINE	1.47B	14
ProNE	1.47B	10
NetSMF	1.47B	16
BERT-base	110M	2
SciBERT	110M	2
GLM-2B	2B	5
GLM-10B	10B	18

B Implementation Details

The parameters and running time of the main methods are listed in Table 3. All experiments are conducted on a Linux server with 56 Intel(R) Xeon(R) Platinum 8336C CPU, 1.88T RAM, and 8 NVIDIA A100 GPUs, each with 80GB memory.

For each fine-tuned pre-trained model, we search for the best learning rate in the range of $\{1e^{-5}, 3e^{-5}, 1e^{-4}, 3e^{-4}\}$, and the best learning rate is set to $1e^{-4}$ according to the performance on the validation set. For LINE in CogDL, we set the `walk_length` and `walk_num` to 5 and 5, respectively. For NetSMF in CogDL, we set the `window_size` and `num_round` to 5 and 5, respectively. For ProNE in CogDL, we use its default parameters. For graph-based methods, the constructed citation graph includes $11{,}478{,}633$ nodes and $167{,}161{,}322$ edges. For supervised methods, we keep all positive instances and sample negative instances randomly, keeping their ratio at $1:10$. For the ensemble model, we use MinMax normalization to scale the outputs of different methods.

C Responsible NLP Checklist

A For every submission

☑ A1. Did you discuss the *limitations* of your work?
In Section 10.

☒ A2. Did you discuss any potential *risks* of your work?
Work doesn't have immediate ethical risk.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1 and Abstract.

B ☑ Did you use or create *scientific artifacts*?
In Section 4.

☒ B1. Did you cite the creators of artifacts you used?
N/A.

☑ B2. Did you discuss the *license or terms* for use and/or distribution of any artifacts?
Yes, we discussed the distribution of our dataset, which has been made public under ODC-BY.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their *intended use*, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The created dataset and original data is used for research purposes only.

☑ B4. Did you discuss the steps taken to check whether the data that was collected/used contains any *information that names or uniquely identifies individual people* or *offensive content*, and the steps taken to protect / anonymize it?
We anonymize the annotators' information.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
In Section 4 and Section 5.

☑ B6. Did you report relevant statistics like the number of examples, details of train/test/dev splits, etc. for the data that you used/created?
In Section 4.

C ☑ Did you run *computational experiments*?
In Section 7.

☑ C1. Did you report the *number of parameters* in the models used, the *total computational budget* (e.g., GPU hours), and *computing infrastructure* used?
In Section B.

☑ C2. Did you discuss the experimental setup, including *hyperparameter search* and *best-found hyperparameter* values?
In Section B.

☒ C3. Did you report *descriptive statistics* about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Since the fine-tuning process and network embedding training process are time-consuming, we perform a single run for each method. Meanwhile, our focus is not to develop a best-performing method but to explore the potential of different methods for the PST problem.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
In Section 7.1 and Section B.

D ☑ Did you use *human annotators* (e.g., crowdworkers) or *research with human subjects*?
In Section 4.

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
In Section 4 and Section A.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such *payment is adequate* given the participants' demographic (e.g., country of residence)?
In Section A.

☑ D3. Did you discuss whether and how *consent* was obtained from people whose data you're using/curating (e.g., did your instructions explain how the data would be used)?
In Section A.

☒ D4. Was the data collection protocol *approved (or determined exempt)* by an ethics review board?
N/A.

☑ D5. Did you report the basic demographic and geographic characteristics of the *annotator* population that is the source of the data?
In Section A.

E ☒ Did you use *AI assistants* (e.g., ChatGPT, Copilot) in your research, coding, or writing?
Left blank.