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 hinder researchers from understanding the evolution of science efficiently. To date, there is still a lack of an accurate and scalable 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 dataset PST-Bench in computer science. Based on PST-Bench, we reveal several intriguing discoveries, such as the differing evolution patterns across various topics. An exploration of various methods underscores the hardness of PST-Bench, pinpointing potential directions on this topic. The dataset and codes have been available[1].

1 Introduction

Comprehending the patterns of scientific evolution, such as the trends of topics and the flow of ideas, are critical for funding agencies in policy development and for researchers in knowledge discovery (Fortunato et al., 2018). The trajectory of scientific evolution can be discerned through citation relationships. However, a notable gap persists between the large-scale and semantically rich citation relations (Zhang et al., 2019a; Tang et al., 2009) and the backbone structure of scientific evolution.

To reveal the essence of scientific development, how can we simplify the citation graph to trace the source of publications and uncover the relationships of inspiration between papers? One might intuitively consider the most cited references of each paper as the sources, discarding other citation relations. However, this is not the case. Based on around 1,500 computer science papers and their professionally annotated source papers, we visualize the cumulative distribution function (CDF) of the references' citation ranks of source papers in Figure 1. It shows that if we consider the most cited reference as the source paper, the accuracy is less than 10%. Further, nearly 70% of papers have source papers that are not among the top-5 cited references. This challenges the intuition that the citation number is the primary indicator to identify the source of publications. For example, Random Forest (Breiman, 2001), Scikit-learn (Pedregosa et al., 2011), and ImageNet (Deng et al., 2009) are among the most cited papers. However, they are not frequently regarded as the direct sources of annotated papers as these works are popular and classic methods/tools/benchmarks. In contrast, the TAGE branch predictor (Seznec and Michaud, 2006), a performance-critical component in modern CPUs, receives less than 20 citations per year on average, but its inspired variants are applied to most high-end ARM processors (Pellegrini, 2021) and AMD Zen processors (Suggs et al., 2020).

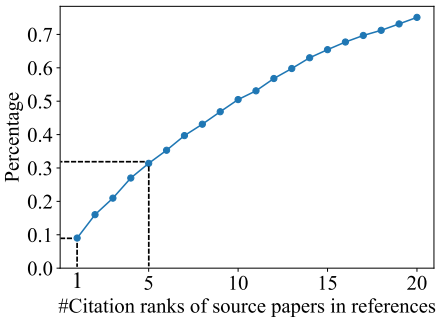


Figure 1: The cumulative distribution function (CDF) of the references' citation ranks for source papers.

Tracing the source of publications is a challenging issue that remains under-explored. 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 com-

[1] https://anonymous.4open.science/r/paper-source-trace-017D

prises less than 100 annotated important citing pairs. Given the high level of expertise required for annotation, many related works employ automated methods to generate datasets. Algorithm Roadmap (Zha et al., 2019) applies weak supervision in the citation contexts to generate datasets and 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 assesses 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. Consequently, relevant resources suffer from *small scale*, *lack of diversity due to machine generation*, or *absence of professional annotation*.

Present Work. For this purpose, in this study, we first formally define the problem of paper source tracing (**PST**) and introduce **PST-Bench**, a professionally annotated PST dataset comprising 1,576 computer science papers and 55,014 associated references, supplemented by additional 4,800 papers and their rule-generated source papers. Each target paper within this dataset has been meticulously annotated with its source papers. We devise a new data annotation strategy via an online paper reading group to ensure high-quality and ever-increasing professional annotations. Second, we perform a comprehensive analysis of this dataset, examining aspects such as the year gap and cross-venue influence between papers on different topics and their source papers, 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 develop a novel strategy that leverages a reading group of graduate students to share papers and mark the sources of papers accurately and regularly.
- We perform in-depth analyses of the PST graph, revealing several intriguing discoveries. For instance, papers in high performance computing (HPC) tend to draw inspiration from less-cited papers than AI papers, even though the former are inclined to be influenced by older papers.
- We explore a variety of methods to automatically trace the source of papers, including statistical methods, graph-based methods, and pre-trained language model (PLM) based methods. Experiments indicate that 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.

PST-Bench can be used for various research topics, such as understanding scientific evolution, studying automatic paper source tracing, and measuring paper impact, aiming to boost innovation through analogy mining and thinking ultimately.

2 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.

3 Building the PST-Bench

Considering the specialized knowledge necessary for tracing the sources of academic papers, we engaged dozens of computer science graduate stu-

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.

dents 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 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.

After gathering and preprocessing the data, we obtain a total of 1,576 labeled computer science papers. The dataset is then partitioned based on their publication year, with 788 papers allocated for training, 394 for validation, and the remaining 394 set aside for testing.

Furthermore, we additionally generate a supplementary dataset by extracting references that appear near signal words like "motivated by" and "inspired by" as source papers, resulting in 4,800 papers with their rule-generated source papers.

Quantity control & quality control. We devise several strategies to ensure a steady and high-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 offer 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

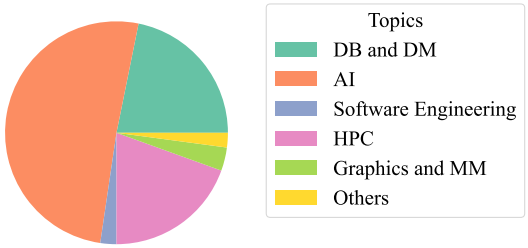


Figure 3: Paper topic distribution. DB and DM: Database and Data Mining, AI: Artificial Intelligence and Pattern Recognition, HPC: High Performance Computing, Graphics and MM: Computer Graphics and Multimedia.

data, including verifying the existence of citation relationships between *ref-sources* and target papers, identifying the perfunctory annotations via the quality of the reading notes (e.g., incoherent abstract translation without modifications), 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%.

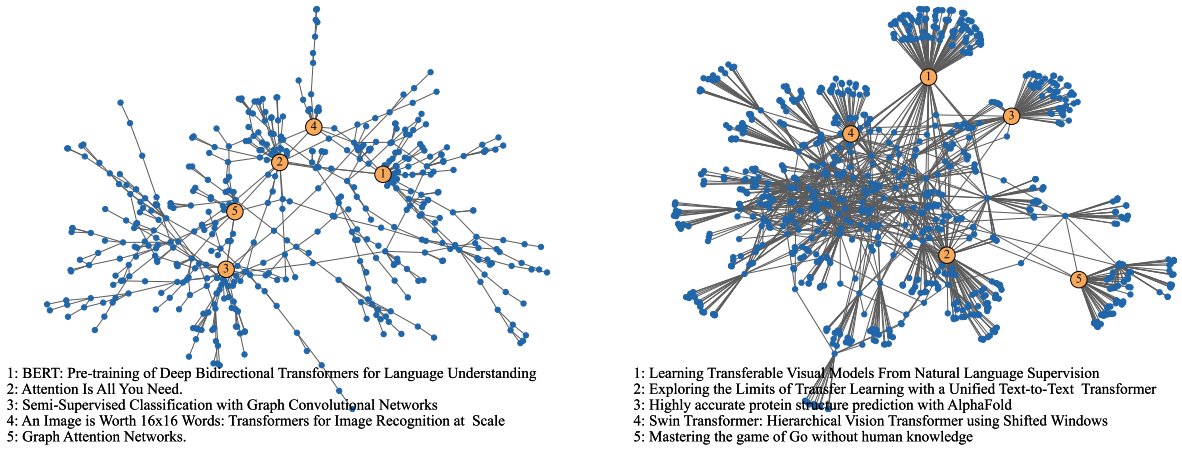
4 Preliminary Study

4.1 Overall Analysis of PST-Bench

Paper topic distribution. Figure 3 visualizes the topic distribution of the collected papers, which are categorized into five subtopics[2]. This figure reveals that the majority of papers fall within the AI field, followed by *database and data mining* and *high performance computing (HPC)*. This distribution is largely due to the fact that our paper reading group initially expanded from students in the HPC and AI groups. Papers in other fields can be added to the dataset in a similar way in the future.

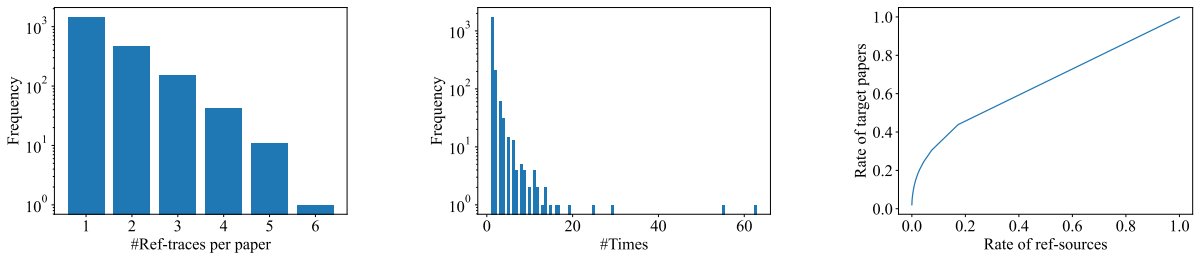
PST graph vs. citation graph. The PST graph, denoted as $\mathcal{G}_{\text{pst}} = \{\mathcal{P}, \mathcal{E}\}$, consists of a paper set $\mathcal{P}$ and edge set $\mathcal{E}$. Each edge $e \in \mathcal{E}$ represents the relations between one paper and its *ref-sources*. For better visualization, we plot the largest connected component of the PST graph, including paper nodes with over 100 citations, in Figure 4(a). We discover that papers are scattered in several "communities", each containing a "super node". This figure vividly illustrates the research threads of several fields. For instance, Transformers (Vaswani et al., 2017) (*node 2*) and BERT (Devlin et al., 2019) (*node 1*) inspired a significant body of pre-training works, including ViT (Dosovitskiy et al.,

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



(a) PST graph (b) Citation graph

Figure 4: Visualization of the simplified PST graph and the simplified citation graph.



(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 5: Analysis of the distribution of *ref-sources*.

2020) (*node 4*). ViT, in turn, inspired numerous research works in computer vision. On the left, graph convolutional networks (GCN) (Kipf and Welling, 2017) (*node 3*) and its subsequent inspired graph attention networks (GAT) (Veličković et al., 2018) (*node 5*) are two pioneering works that inspired a lot of studies in graph mining.

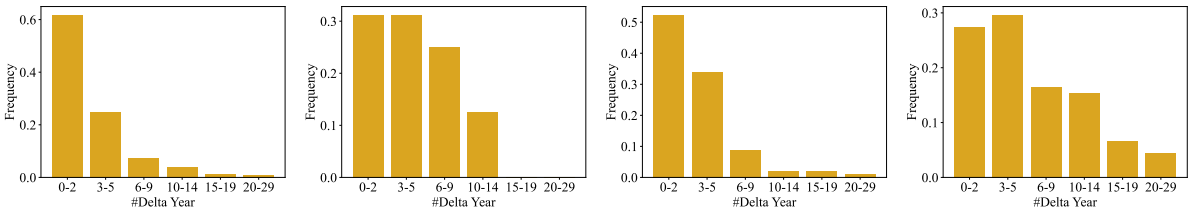
Additionally, we plot the corresponding citation graph in Figure 4(b) for comparison. Given the density of citations, we visualize paper nodes with at least 10,000 citations. Despite this simplification, Figure 4(b) is much denser than Figure 4(a). Figure 4(b) presents more diverse research fields, including language-image pretraining, natural language pretraining, protein pretraining, vision pretraining, etc. In Figure 4(b), Swin Transformer (Liu et al., 2021) (*node 4*) cites CLIP (Radford et al., 2021) (*node 1*), and CLIP cites T5 (Raffel et al., 2020) (*node 2*). However, these citation relationships exist primarily due to background introductions and don't represent the evolution of relevant fields. Thus, it is arduous to identify the evolution of these research works from the intricate citation graph.

4.2 Distribution Analysis of *ref-sources*

In the following subsection, we conduct a detailed analysis of the distribution of *ref-sources*.

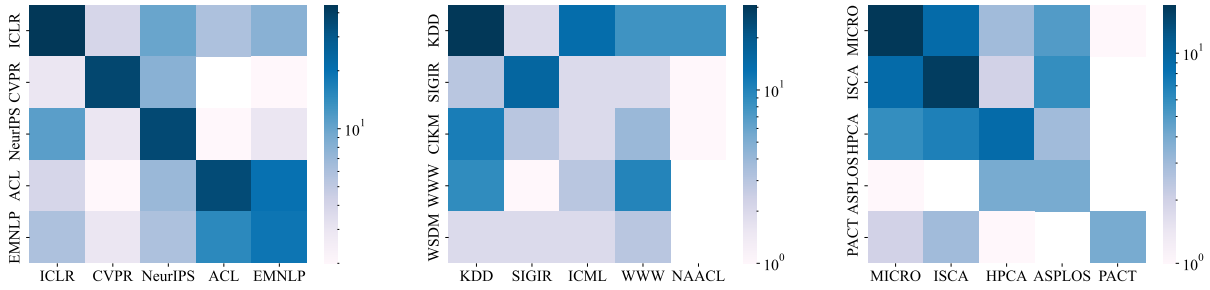
***Ref-sources* per paper.** Figure 5(a) depicts the histogram of the number of *ref-sources* per paper. It demonstrates that most annotated papers have only one *ref-source*, with about 10% of papers having more than three *ref-sources*. This could reflect the actual distribution of *ref-sources* per paper to some extent, suggesting that the majority of annotated papers are inspired by one significant idea.

Matthew effect of *ref-sources*. Figure 5(b) and Figure 5(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 5(c),



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

Figure 6: Year gap between a paper and its *ref-sources* in different fields.



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

Figure 7: Influence between computer science venues.

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.

4.3 Analysis of Different Topics

What are the underlying evolution patterns of different topics? In the following subsection, we conduct analyses from multi-faceted aspects.

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 6 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 five years. Papers are less likely to be influenced by older publications. (2) Clear differences between fields exist 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 half of the papers are inspired by papers from 0-2 years ago. Some HPC papers are even inspired by papers published more than 20 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 7 displays the heatmaps of pairwise venue influence on these subtopics. We highlight several observations below. (1) AI venues are mostly influenced by AI venues. (2) In addition to being affected by data mining (DM) conferences, DM conferences are also influenced by AI conferences (e.g., ICML and NAACL). (3) HPC conferences are primarily influenced by HPC conferences. These figures clearly demonstrate the cross-influence between different fields in computer science.

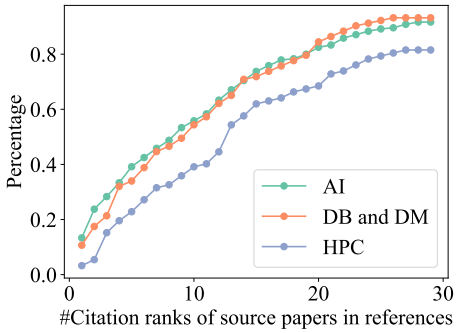


Figure 8: The cumulative distribution function (CDF) of the references' citation ranks for *ref-sources* with respect to different topics.

Are papers inclined to be inspired by the most cited references? Figure 8 plots the CDF of the references' citation ranks for being *ref-sources* w.r.t. different topics. The HPC curve is clearly below the curves of AI, DB and DM. That is, compared with the other two fields, despite the fact that HPC papers tend to be motivated by older publications, it doesn't mean that HPC papers are more likely to be inspired by most cited references. The correlation between the citation number of a reference and its probability of being a *ref-source* is weaker in HPC than in AI, DB and DM.

5 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, and (3) pre-trained language model (PLM) based methods.

5.1 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.

5.2 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[3] and learn paper embeddings with network embedding methods, such as **LINE** (Tang et al., 2015), **ProNE** (Zhang et al., 2019b), **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.

5.3 PLM-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), **GLM** (Du et al., 2022), and **Galactica** (Taylor et al., 2022). We also adopt three state-of-the-art closed-source models: **GPT-3.5** (OpenAI, 2022), **GPT-4** (Achiam et al., 2023), and **Claude** (Anthropic, 2023).

6 Experiments

6.1 Experimental Setup

For the full texts of papers, we use the GROBID[4] API to convert PDF to XML format for convenient

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

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

Table 1: Accuracy results of paper source tracing.

	Method	MAP
Stat	Rule	0.0616
	RF	0.1821
Graph	LINE	0.1047
	ProNE	0.1050
	NetSMF	0.1231
PLM	BERT-base	0.2775
	SciBERT	**0.3240**
	GLM-2B	0.1503
	Galactica-standard	0.1472
	GPT-3.5	0.0781
	GPT-4	0.0519
	Claude-instant	0.0536

Stat: statistical methods.

processing of citation contexts. We employ regular expressions 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 PLM-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,

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

where R_p is the number of *ref-sources* of paper p, M_p is the number of references of paper p, $\text{Prec}_p(k)$ 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.

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

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

6.2 Main Results

Table 1 presents the results of paper source tracing. Among statistical methods, Random Forest (RF) surpasses the Rule method, emphasizing the efficacy of feature engineering. 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.

In terms of graph-based methods, NetSMF outperforms LINE and ProNE, likely due to its ability to capture higher-order proximity of nodes via sparse matrix factorization.

As for PLM-based methods, SciBERT significantly surpasses other models, demonstrating the effectiveness of pre-training on domain-specific data. Surprisingly, finetuned SciBERT and BERT-base outperform larger models like GLM-2B, Galactica-standard, and closed-source PLMs. The reason may lie in two aspects. First, the training objective of the mask language model is more suitable for this context understanding task. Second, API-based models may not be well-trained on similar tasks. However, the results of current methods are not yet optimal, suggesting significant potential for further research in this field, such as combining multiple categories of methods.

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.

6.3 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.

6.4 Error Analysis

We conduct a case study of the prediction errors made by our best-performing model, with several examples shown in Figure 9. We list each target paper with its *ref-source* and the corresponding contexts. We have the following observations. For

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 9: Predictive error analysis.

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 the main ideas of papers might be omitted in the current contextual methods.

7 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 with both incidental and important citations. Jurgens et al. (2018) introduce a larger dataset of nearly 2,000 citation pairs in the NLP area, in which less than 100 citation pairs are annotated as the motivation. 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 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.

8 Conclusion

In this paper, we present PST-Bench, a novel, professionally annotated, and ever-growing benchmark for paper source tracing. We conduct detailed analyses on PST-Bench and offer several insights, such as the differing evolution patterns of papers across different topics. PST-Bench facilitates further analysis of the evolution of science and a deep understanding of the crux of research works, and so on. We plan to expand the coverage of PST-Bench to more topics and design elaborate methods to improve the accuracy of the PST problem.

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. (1) The topics covered in PST-Bench are not even, with most topics related to AI, data mining, and high performance computing. In the future, we plan to call for students majoring in different areas to expand the coverage of PST-Bench. (2) Although we explore various types of methods for automatic paper source tracing, more advanced methods tailored for the PST problem are absent. However, the elaborate method design for the PST problem is not the main focus of this paper. We plan to optimize the methods and call for contributions to improve the performance of automatic paper source tracing.

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 mine and understand the evolution of science or develop automatic methods to trace the source of papers, etc.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2023. Introducing claude. https://www.anthropic.com/news/introducing-claude.

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.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255.

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.

Santo Fortunato, Carl T Bergstrom, Katy Börner, James A Evans, Dirk Helbing, Staša Milojević, Alexander M Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, et al. 2018. Science of science. *Science*, 359(6379):eaao0185.

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.

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.

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*.

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*.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

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.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

André Seznec and Pierre Michaud. 2006. A case for (partially) tagged geometric history length branch prediction. *The Journal of Instruction-Level Parallelism*, 8:23.

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.

David Suggs, Mahesh Subramony, and Dan Bouvier. 2020. The amd "zen 2" processor. *IEEE Micro*, 40(2):45–52.

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 *9th IEEE International Conference on Data Mining*, pages 1010–1015.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations*.

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.

Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, et al. 2019a. OAG: Toward linking large-scale heterogeneous entity graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2585–2595.

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 10: Reading group rules.

Jie Zhang, Yuxiao Dong, Yan Wang, Jie Tang, and Ming Ding. 2019b. 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 10. Currently, each paper is annotated by one student. Recruited group members are told that the collected data will be public and used for research purposes only. Next, we detail the components of data annotation to ensure data quality.

Maintenance of the paper reading group. We periodically hold paper reading groups on WeChat every week and publicize the reading group on the public forums of several universities and familiar labs. Inactive group members are removed every four weeks. We remove group members immediately once they have made perfunctory annotations.

Reward and punishment mechanism. The reward mechanisms are divided into immediate and long-term rewards. As shown in Figure 10, students receive rewards each week or once they mark every 20 papers. In contrast, students who didn't share their paper reading notes need to give red packets to those who completed paper sharing. 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.

Demographics of group members. Until February 2024, there have been 101 members who participated in the effective annotation. We don't know many demographics of volunteering students, but most of them are from China, studying in renowned universities or research institutes, including Chinese Academy of Sciences, Tsinghua University, Harbin Institute of Technology, Southeast University, Nankai University, etc.

PST-Bench has been made public under the ODC-BY license. The created dataset and the original data are used for research purposes only. We have anonymized the annotators' information.

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
Galactica	6.7B	13

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 the fine-tuned BERT, SciBERT, and GLM 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 the Galactica model, we adopt the Xturing[5] framework and use the default parameters. As for API-based methods, we use the same input contexts as other open-sourced pre-trained models for a fair comparison. We have submitted the paper PDF files and asking the GPT/Claude which references are the most significant to inspire the given papers, but the responses are basically unreasonable. 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 $1:10$.

[5] https://github.com/stochasticai/xTuring/

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 3.

☒ 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 3 and Section 4.

☑ 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 3.

C ☑ Did you run *computational experiments*?
In Section 6.

☑ 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 6.1 and Section B.

D ☑ Did you use *human annotators* (e.g., crowdworkers) or *research with human subjects*?
In Section 3.

☑ 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 3 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.