LEGOBENCH: Scientific Leaderboard Generation Benchmark

Anonymous ACL submission

Abstract

The ever-increasing volume of paper submissions makes it difficult to stay informed about the latest state-of-the-art research. To address this challenge, we introduce LEGOBENCH, a benchmark for evaluating systems that generate scientific leaderboards. LEGOBENCH is curated from 22 years of preprint submission data on arXiv and more than 11k machine learning leaderboards on the PapersWithCode portal. We present four graph-based and two language model-based leaderboard generation task configurations. We evaluate popular encoder-only scientific language models as well as decoder-only large language models across these task configurations. State-of-the-art models showcase significant performance gaps in automatic leaderboard generation on LEGOBENCH. The code is available on GitHub[1] and the dataset is hosted on OSF[2].

1 Introduction

Comparison of results with prior state-of-the-art (SOTA) is a standard practice in experimental research papers. Performance on a task using a specific metric establishes the efficacy of the paper's proposed method. However, one of the primary challenges in scientific research is keeping up with the rapid volume of research progress and staying updated with the latest SOTA to compare with one's work. The increasing number of manuscripts (depicted by arXiv submissions in Appendix A.1) showcases the severity of information overload. With the continuous stream of submission, revision, and acceptance timelines of conferences and journals, researchers often struggle to keep up with the latest methods and developments. Thus, being acquainted with the latest papers, sieving through the massive set, and deciding which baselines to compare with can be challenging and time-consuming. Moreover, the latest papers with novel methods may have low visibility, and upcoming papers may overlook those for result comparison as citations are biased towards old compared to new papers.

To address the information overload and to facilitate the comparison with meaningful baseline works, multiple previous works mine scientific tables from papers (Kayal et al., 2022; Zhong et al., 2020; Deng et al., 2019; Liu et al., 2007) and construct scientific leaderboards (Yang et al., 2022; Kabongo et al., 2023, 2021; Kardas et al., 2020; Hou et al., 2019). A scientific leaderboard curates performance scores of competitive models against the triple <dataset, task, metric>. One of the most actively maintained platforms, PapersWithCode (PwC) (Stojnic et al., 2018), hosts leaderboards in empirical machine learning. Figure 4 in Appendix A.2 shows a representative leaderboard sample for the image clustering task on the MNIST dataset, available in PwC. While leaderboards are helpful for researchers to track the latest models, a majority of leaderboard curation initiatives are manually maintained (Stojnic et al., 2018), or are dormant (Eckersley, 2017; Tao, 2017; Ruder, 2018). Hence, the need to automate the generation of leaderboards is imperative.

To streamline the process of automating leaderboard generation, we create the arXiv Papers Collection (APC), a curated collection of research papers and graph data (citation network and performance comparison network) from arXiv. We also create a dataset sourced from PapersWithCode (PwC), consisting of leaderboards mapped with arXiv papers. We combine APC and PwC datasets to develop a benchmark framework called LEGOBENCH, that facilitates evaluation and assessment of automatic leaderboard generation models. LEGOBENCH introduces the leaderboard generation task in three configurations, (i) Ranking Papers based on Content and Graph [RPG], (ii) Ranking Papers by Prompting Language Models

[1] code would be made public post acceptance

[2] https://osf.io/9v2py/?view_only=6f91b0b510df498ba01595f8f278f94c

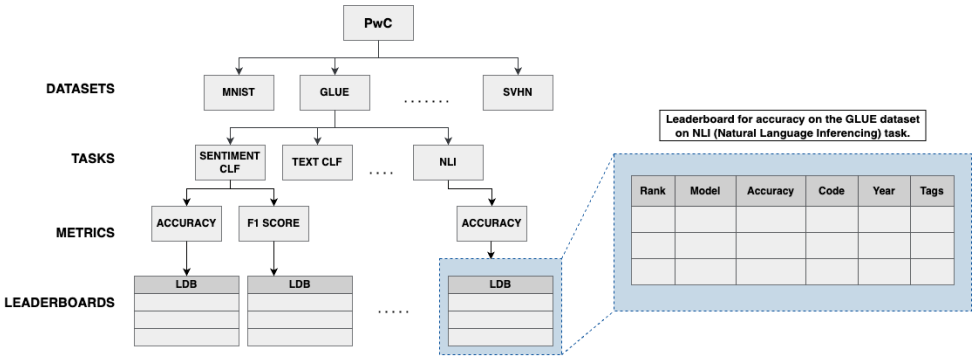


Figure 1: Organization of leaderboards in PwC. A leaderboard is constructed for a <dataset, task, metric> tuple. Leaderboards can contain additional metadata, such as the code repository link and model description tags.

[RPLM], and (iii) Leaderboard Entries Generation by Prompting Language Models [LGPLM]. Our contributions can be summarized as follows:

- We curate the first leaderboard generation framework, LEGOBENCH, where we provide datasets and metrics for evaluating scientific leaderboard generation. Our dataset consists of 22 years of arXiv data and 11k leaderboards from PwC, available publicly on OSF[2].
- We present six leaderboard generation task configurations, including four that are graph-based and two that utilize language models. The diverse task configurations allow for a comprehensive evaluation of systems, showcasing our framework's adaptability and breadth across differing methodologies.
- We assess the ability of the existing off-the-shelf encoder-only scientific LMs and decoder-only LLMs in the context of leaderboard generation. Our results showcase the severe limitations of existing models, uncovering avenues for future models to address.

2 Dataset

We curate two datasets, (i) PwC Leaderboards (PwC-LDB) and (ii) arXiv Papers Collection (APC), which are utilized for the construction of LEGOBENCH. PwC-LDB is curated from Papers With Code repository[3] and APC is curated from arXiv preprint repository[4].

2.1 PwC-LDB

PwC-LDB is a dataset of leaderboards for various ML tasks curated by the PwC repository. The PwC repository is curated and updated regularly. It is organized in a hierarchy, with ML datasets forming the parent nodes and tasks associated with dataset nodes being the children. Each task is evaluated using certain metrics, and hence each leaderboard is associated with a dataset (D), task (T), and metric (M), as represented in Figure 1. Formally, a leaderboard $\mathcal{L}$(D, T, M) is defined for a triplet <D, T, M>, where an algorithm/method/model (hereafter denoted as A) is evaluated against D, T, and M. <D, T, M, A>, thus, represents the addition of algorithm/method/model to the DTM triple. Every leaderboard consists of multiple A's that compare their performance scores against each other. Information such as code repository, year, model parameters, etc., could also be present.

Artifact	PwC-LDB	AP-LDB
Datasets	3666	1697
Tasks	1660	675
Metrics	2958	1381
DTMA	70559	43105
Leaderboards	11470	4409

Table 1: Statistics of PwC-LDB & AP-LDB dataset. DTMA refers to <data, task, metric, method> tuple representing an entry in the leaderboard. AP-LDB is curated by mapping PwC-LDB with APC.

We curated 3666 machine learning (ML) datasets, 1660 tasks, and their corresponding leaderboards from the PwC repository. 11,470 leaderboards contain 70,559 <D, T, M, A> tuples. On average, a leaderboard contains six entries, i.e., a performance comparison of six models. The maximum number of entries in a leaderboard is 863 for the 'Image classification on ImageNet using Top-1 Accuracy' leaderboard. The statistics of the PwC-LDB dataset are presented in Table 1.

2.2 APC

APC is a collection of datasets curating diverse paper information from arXiv[4], a research-sharing

[3]https://paperswithcode.com/. Dataset curation - 06/2023.
[4]https://arxiv.org/. Dataset curation - 09/2022.

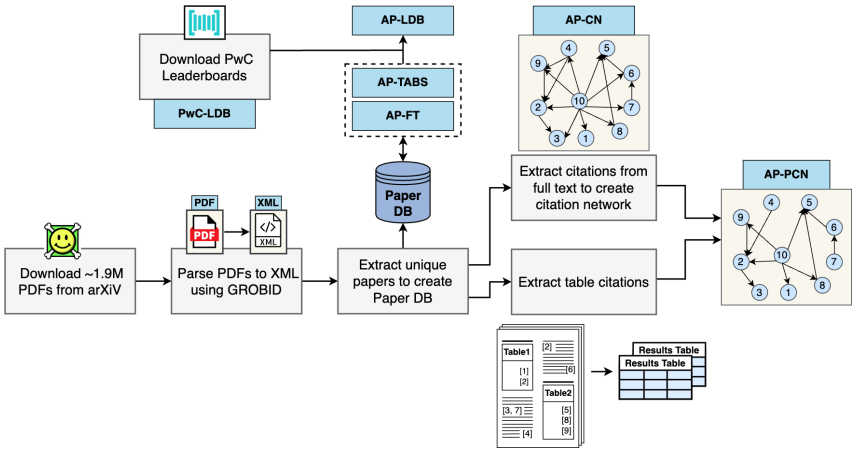


Figure 2: Pipeline for constructing the APC datasets. Blue boxes denote various datasets in the APC collection and the PwC-LDB dataset.

Dataset	Summary	Size
AP-TABS	Titles and Abstract of arXiv papers extracted from metadata	1.9M Papers
AP-FT	Full text of arXiv papers extracted after parsing PDFs with GROBID	1.9M Papers
AP-CN	Citation network of arXiv papers extracted using regex from AP-FT	18M nodes & 59M edges
AP-PCN	Performance comparison network extracted from table citations using AP-FT and AP-CN	280k nodes & 309k edges
AP-LDB	PWC-LDB mapped with the arXiv Papers Collection	9.8k leaderboards & 41k <TDM>

Table 2: Summary of datasets in the LEGOBENCH benchmark.

platform that hosts preprints of scientific papers in eight domains. We curate titles and abstracts (AP-TABS), full-texts (AP-FT) from arXiv, and process the data to extract the citations (AP-CN) and performance comparisons (AP-PCN). Next, we discuss the stages (denoted by [S*i*]) in the APC curation pipeline. Figure 2 illustrates the pipeline.

[S1] arXiv Paper Curation: We curate the arXiv data by collecting metadata and paper PDFs from Jan 2000 to July 2022, consisting of 1,942,301 papers categorized into eight broad domains. The domain-wise statistics of the curated papers are presented in Appendix A.3. In the remainder of this paper, we refer to the title and abstract metadata obtained directly as the **AP-TABS** dataset (arXiv Papers' Titles and Abstracts).

[S2] Parsing PDFs: We parse PDFs into TEI-XML format using GROBID (Lopez, 2009). GROBID successfully extracts full-text information from 1,940,910 papers, which is referred to as **AP-FT** (arXiv Papers' Full Text) dataset.

[S3] Constructing the Unique Paper Index: We construct a unique index of all papers present in our dataset. This index includes 1.9M papers present in AP-FT along with their references.

[S4] Construction of the Citation Network: The arXiv Papers' Citation Network dataset (**AP-CN**) consists of citations in the AP-FT dataset (details in Appendix A.3). However, the leaderboards in the PwC-LDB dataset are present only for computer science, specifically in ML.

[S5] Table Extraction and Construction of the Performance Comparison Network: Finally, we extract tabular information from parsed TEI-XML paper format in the AP-FT dataset. Citations in the table are extracted and mapped to the papers in the unique index. It should be noted that the table citation data of a paper is a subset of the references of the paper. We only include papers containing at least one table with at least one citation in the table text or the caption to construct the **AP-PCN** (arXiv Papers' Performance Comparison Network) data.

[S6] Mapping PwC-LDB with APC: PwC-LDB dataset consists of leaderboards for a task, dataset, and metric triple, denoted by <T, D, M>. Each model A is listed with the title of the paper and its unique arXiv Identifier, if available. We leverage these identifiers to map to the AP-FT dataset. The paper's metadata or full-text information is originally absent in the PwC-LDB, which we collate and map in the AP-LDB dataset.

A summary and statistics of datasets in LEGOBENCH are presented in Table 2 and size, domain, and modality statistics of the AP-CN and the

AP-PCN dataset are presented in Appendix A.3.

3 LEGOBENCH: Automatic Scientific Leaderboard Generation Benchmark

We present LEGOBENCH, a benchmark specifically developed for the evaluation of scientific leaderboard generation. This benchmark tasks systems with generating a leaderboard in response to a natural language query that specifies a dataset (D), a task (T), and a metric (M), utilizing a collection of arXiv Papers' datasets. To comprehensively assess the capabilities of scientific leaderboard generation systems, we design three tasks, resulting in a total of six configurations. These configurations employ multiple frameworks and use various datasets from the arXiv Papers' Collection to assess the generation of diverse leaderboard formats.

The three leaderboard generation tasks are: (i) Ranking Papers based on Content and Graph [RPG], (ii) Ranking Papers by Prompting Language Models [RPLM], and (iii) Leaderboard Entries Generation by Prompting Language Models [LGPLM]. The benchmark for the first two ranking tasks comprises 4409 leaderboards for 675 empirical ML tasks on 1697 datasets. The LGPLM task comprises 9847 leaderboards from the AP-LDB dataset. For each task, the format is: Given query q = <D, T, M>, and arXiv dataset $\mathcal{D}$, the task is to generate a leaderboard $\mathcal{L}$ corresponding to the query q. The format of q, $\mathcal{D}$ and $\mathcal{L}$ depend on the task configuration. Next, we describe each task in detail and discuss the four task configurations for the RPG task and one configuration for the RPLM and LGPLM tasks.

3.1 Ranking Papers based on Content and Graph [RPG]

Given a short natural language query q (consisting of D, T, and M details), this task format requires ranking candidate papers based on the performance score. For RPG tasks, $\mathcal{L}$ is a ranked list of papers, where the best rank indicates the best performance on the <D, T, M> triple. The first step is retrieving a set of candidates from the arXiv Papers' Collection. It leverages the network structure as well as the paper content, to generate a ranked list of papers such that the papers with the best performance are ranked highest, and ranks increase as performance decreases. We encourage the evaluation of graph models for this task format as we present multiple configurations of this task with different text and network datasets.

1. **Ranking Papers in the Citation Network with title and abstract (RPG[CN-TABS]):** Given the title and abstract of each paper in the arXiv Papers' Citation Network, i.e., $\mathcal{D}$ = AP-CN $\bigcup$ AP-TABS, construct the leaderboard based on the citation network properties and the content.
2. **Ranking Papers in the Performance Comparison Network with title and abstract (RPG[PCN-TABS]):** Given the title and abstract of each paper in the arXiv Papers' Performance Comparison Network, i.e., $\mathcal{D}$ = AP-PCN $\bigcup$ AP-TABS, construct the leaderboard based on the comparison network properties and the content present in TABS. The AP-PCN dataset encodes which paper compares results with which papers; however, if the performance is better or worse, it is not present in the graph dataset. The improvement needs to be extracted from the TABS dataset and used in addition to the comparison data to generate the correct ranking of papers in the leaderboard.
3. **Ranking Papers in the Citation Network with Full texts (RPG[CN-FT]):** Given access to the full text of papers along with the citation network, i.e. $\mathcal{D}$ = AP-CN $\bigcup$ AP-FT, this task focuses on generating the ranked paper list by leveraging the network as well as the full text. For example, the full text can be utilized to learn node embeddings in the graph.
4. **Ranking Papers in the Performance Comparison Network with Full texts (RPG[PCN-FT]):** This task is similar to RPG[CN-FT], except that instead of the citation network, performance comparison network is provided. $\mathcal{D}$ = AP-PCN $\bigcup$ AP-FT.

3.2 Ranking Papers by Prompting Language Models [RPLM]

RPLM focuses on ranking the papers using language models (LMs). Given a natural language query q (consisting of D, T, and M details), and $\mathcal{D}$ is a randomly shuffled list of paper titles present in the leaderboard corresponding to <D, T, M>. LMs are expected to generate output $\mathcal{L}$ as a ranked list of paper titles, such that the best-ranked paper in the list achieves the best score on <D, T, M>. This task intends to leverage LMs to rank papers by retrieving the best performance score of the model discussed in the paper. This task opens

the possibilities for evaluating if a language model encodes/memorizes relevant information about the paper results in the model parameters as paper text is not provided. However, it is a challenge to automatically ascertain whether the rationale behind the generated rankings truly takes into account the extracted scores. We use the AP-LDB dataset to construct the queries. As this is a ranking task, we only use leaderboards where at least three unique papers are present so that it is practical to evaluate the generated paper title rankings.

3.3 Leaderboard Entries Generation by Prompting Language Models [LGPLM]

LGPLM is modeled as a QA task over documents, where given the query q (consisting of D, T, and M details), and $\mathcal{D}$ = AP-FT, a language model extracts method performance from papers experimenting on T and D and reporting scores with M. It focuses on the extraction of leaderboard entries, consisting of method/model/algorithm names (A in <D, T, M, A> tuple, and referred as method hereafter in the paper) and the performance scores for metric M and arranging them into a leaderboard. The ith leaderboard entry in $\mathcal{L}$ can be represented as <m_i, s_i>, where m_i is the method name and s_i is the score. In our dataset, $\mathcal{L}$ is stored as a markdown table containing <m_i, s_i> entries, in string format. We chose the markdown table format for storing the leaderboard entries as our preliminary evaluation highlighted that most LLMs are efficient at generating a uniform markdown table rather than any other format (e.g. tab or space separated columns). However, during evaluation, we do not use straightforward rouge-based evaluations to compare the LLM output with the ground truth. Instead, we parse the string to extract methods and scores. Metrics are discussed in Section 3.4.

The three tasks, RPG, RPLM, and LGPLM are designed for different purposes. While the RPG task focuses on paper graph representation models and models for ranking graph nodes, the RPLM task focuses on retrieving the score information from the memory of scientific language models. LGPLM focuses on the extraction of methods and their scores from the paper text and utilizing that for leaderboard generation.

3.4 Evaluation of Leaderboard Generation

For the RPG task, we use Kendall's Tau (KTau) (Kendall, 1938) to measure the rank correlation between the ranked list of paper titles generated by the candidate model and the ground truth ranks of papers in the leaderboard. It is in the range of -1 to +1, indicating perfect inverse or direct association or no association if 0. We also describe some custom-designed metrics. **Binary Exact Match (BEM)** takes binary values, i.e., one iff the two ranked lists are exactly similar; otherwise zero. To enhance readability, we present BEM percentage values, i.e. the percentage of ranked lists that are ordered. **Complete Inclusion Score (CIS)** CIS denotes the percentage of model-generated leaderboards that have all the titles present in the ground truth. **Concordant Pairs (CP)** measures the percentage of pairs of leaderboard entries ranked in the same order as in the ground truth. It lies in the range 0-100, with 100% indicating that all pairs are ranked in the same order. **Method Recall (MR)** is used for the LGPLM task. The output of the LGPLM task is a leaderboard consisting of method/algorithm names and their corresponding scores, arranged in a markdown table. This metric computes the percentage of correct method names in the model-generated leaderboard with respect to the method names in the ground truth. **Method Precision (MP)** is similar to MR, and computes the percentage of correct methods in the model-generated leaderboard with respect to the total number of generated methods. **Score Precision (SP)** computes the percentage of correct scores in the model-generated leaderboard with respect to the total number of generated methods.

4 Preliminary Baselines and Results

We present preliminary baselines for each of the tasks. Due to space constraints, the schematic diagrams for the baselines are presented in Appendix A.4.

Task →	RPG[CN-TABS]		RPG[PCN-TABS]	
Model ↓	BEM	KTau	BEM	KTau
SciBERT	2.66	-0.010	8.26	-0.187
SPECTER	2.60	-0.009	8.74	-0.177
SciNCL	2.25	-0.015	8.51	-0.175
OAG-BERT	2.62	-0.009	8.34	-0.201

Task →	RPG[CN-FT]		RPG[PCN-FT]	
Model ↓	BEM	KTau	BEM	KTau
SciBERT	0.184	-0.006	6.743	-0.140
SPECTER	0.851	-0.008	6.978	-0.137
SciNCL	0.888	-0.006	6.283	-0.015
OAG-BERT	0.665	-0.010	7.201	-0.132

Table 3: Performance of PageRank for ranking nodes. Candidates Retrieval selected intersecting candidate documents retrieved by query unigram search.

4.1 RPG Baselines and Results

We follow a retrieve-then-rank procedure for ranking papers based on content and network as depicted in Figure 5. The first step is candidate retrieval, which selects a subset of papers from the arXiv dataset, followed by a ranker module that ranks the papers to generate leaderboard entries.

Candidate Retrieval Module: We present a straightforward methodology to retrieve candidate papers given the query. We tokenize the query and obtain unigrams. The query unigrams are searched in papers (title and abstract if AP-TABS, and full-text for AP-FT configuration), and the papers containing all the unigrams are selected.

Ranker Module: After candidate retrieval, we construct a graph of the retrieved papers. The candidate papers retrieved in the previous step are considered nodes, and directed unweighted edges are added from the citation or performance comparison network depending on the dataset provided with the task. Weights are added to the existing edges by encoding paper content (TABS) and computing cosine similarity between the paper nodes. We experimented with popular scientific encoder models such as SciBERT (Beltagy et al., 2019), SPECTER (Cohan et al., 2020), SciNCL (Ostendorff et al., 2022), and OAG-BERT (Liu et al., 2022), to encode the paper content. Node scores are calculated using PageRank (Page et al., 1999) and are used to rank the nodes. We report the leaderboard generation performance results in Table 3. KTau values close to 0 for all baselines indicate that there is no correlation between the generated lists. The maximum BEM values are in the range of 6-8%, indicating less than random chance papers being ranked correctly. Among the baselines, the usage of AP-PCN shows promise over AP-CN as performance comparison is a substantial signal compared to citations. In the current settings, usage of the AP-FT leads to inferior performance than AP-TABS, however, we posit the large candidate set retrieved from AP-FT to be the actual reason. AP-FT usage could potentially enhance performance as it is more likely to discuss performance scores exhaustively in the full-text, contingent upon the integration of a more effective candidate retrieval module. Performance comparison networks and paper full-text datasets have long been unexplored and have the potential to improve leaderboard generation performance. Existing baselines however perform poorly, thereby presenting an opportunity to leverage the benchmark for the assessment of novel graph and encoder LMs.

4.2 RPLM Baselines and Results

For the RPLM task, we prompt language models by adding an instruction to the provided natural language query q and paper titles. The pipeline and complete prompt are presented in Figure 6 and Table 10 respectively. We present results with open-source LLMs (Falcon (Almazrouei et al., 2023), Galactica (Taylor et al., 2022), LLama 2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), Vicuna (Chiang et al., 2023), and Zephyr (Tunstall et al., 2023)) and two closed models Gemini (Google et al., 2023) and GPT-4 (Achiam et al., 2023). We evaluate 7B and 13B models and exclude bigger models due to resource constraints.

Results for the models are presented in Table 4. LLama 2 Chat 7B and Vicuna 7B perform best among the open-source LLMs in generating a ranked list that consists of all the titles provided in the input prompt (i.e. CIS metric) for around 16% of the instances in the dataset. The 7B versions of Llama 2 and Vicuna perform better at generating the same titles (and hence understanding the instruction correctly) than their counterpart 13B models. Overall, GPT-4 has the highest CIS of 21.89%. BEM and KTau are calculated for the subset of instances for which the LLM-generated ranked list contains all the papers in the ground truth, hence we omit these values for papers with CIS less than 1%. BEM values are less than 1% for all models, indicating less than 1% of the LLM-generated ranked paper titles are exactly the same as the original ranking of papers. Similarly, all models' KTau values are close to zero, indicating no association between the LLM-generated paper ranks and the original leaderboard ranks. Finally, we present CP, indicating the percentage of concordant pairs in the LLM-generated titles and original leaderboard titles. Note that LLM-generated titles absent from the original prompt input are ignored while computing CP. GPT-4 has the highest CP with 51.93% concordant title pairs, followed by Gemini with 46.06% concordant title pairs. The best performing open-source LLMs, Llama 2 Chat 7B and Vicuna 7B lag behind GPT-4 with around 16 points. Instruction-tuned models perform better than their regular counterparts across all the open-source LLMs.

Manual analysis of generated ranks revealed that Gemini follows the prompt instruction efficiently

by generating a well-formatted ranked paper list. Most open-source models such as Falcon, Llama, Vicuna, and Mistral generate titles absent in the prompt and often also repeatedly generate the same title. Manual inspection of the results indicates that most models are unable to follow the instructions and often end up generating paper titles that were not provided for ranking in the input. These titles are often hallucinated and no papers with such titles exist in the public domain. Falcon 7B, Llama 2 7B, and Llama 2 13B often keep generating the same paper title multiple times in the ranked list, however, the instruction-tuned counterparts of these models generally did not face this issue. The majority of the Galactica-generated answers have HTML tags (specifically <s>, <p>, and sometimes <li>), or the first generated title is repeated in the entire generated text. Galactica model also often ends up generating long text in the format of a paper title and abstract instead of paper title ranks. We also observe that the Vicuna 13B model is chattier in comparison to the Vicuna 7B model despite the instruction clearly stating to only generate the ranked titles and skip any additional text. We posit this as the reason for the slightly better performance of the Vicuna 7B model compared to the 13B model.

Model	CIS	BEM	KTau	CP
7 B Models				
Falcon	1.24	0.20	-0.17	1.65
Falcon Instruct	7.46	0.23	0.00	16.38
Galactica	0.50	-	-	0.00
LLama 2	0.25	-	-	1.36
LLama 2 Chat	16.17	0.11	0.02	35.87
Mistral	1.74	0.29	-0.04	3.25
Mistral Instruct	3.48	0.07	-0.05	7.97
Vicuna	16.92	0.10	0.05	36.87
Zephyr Beta	7.21	0.10	-0.14	13.25
13 B Models				
LLama 2	0.25	-	-	0.46
LLama 2 Chat	10.20	0.07	0.01	29.71
Vicuna	14.18	0.09	0.07	34.66
Closed Models				
Gemini Pro	18.91	0.09	0.06	46.06
GPT-4	21.89	0.08	0.10	51.93

Table 4: Performance of LLMs on the RPLM task.

4.3 LGPLM Baselines and Results

We follow a retrieval-augmented-generation setup for the LGPLM baseline. We use a BM25 ranker module that takes the leaderboard query and full-text paper chunks as input and selects top-k chunks. For our experiments, only the papers present in the leaderboard (i.e. papers that report performance on the specified task and dataset using the specified metric) are split into chunks and provided to the BM25 ranker. We use k=10 for our experiments. The top-k chunks and the query are then provided to a language model, which generates the leaderboard consisting of methods and performance scores. We include the top-10 chunks iteratively in the prompt and keep including the chunks till the model context length is exhausted. The pipeline is presented in Figure 7 and prompt details in Table 11. We use the same set of LLMs as used in the RPLM task.

We present the results for this configuration in Table 5. The Method Recall (MR) is less than 1% for 10 out of the 14 evaluated models (Falcon 7b, Falcon Instruct 7B, Galactica 6.7B, Llama 2, Llama 2 Chat 7B, Llama 2 13B, Llama 2 Chat 13B, Vicuna 7B, and Vicuna 13B). A manual examination of the results reveals that the majority of models merely produce the leaderboard table header in the format "Method | Metric," thereby precluding the possibility of assessing Score Precision (SP). Further, as the models generate table headers only or ill-formatted noisy text, it is infeasible to calculate Model Precision (MP).

MR presents the percentage of ground truth method names that are present in the LLM-generated leaderboard. GPT-4 performs the best, indicating that merely 25.24% original methods are generated by the LLM on average. Gemini performs poorly in comparison to GPT-4 with only 3.3% MR. Among the open-source LLMs, Mistral Instruct 7B has a MR of 20.4%, close to GPT-4. The regular Mistral 7B model performs poorly, attaining less than 1% recall, indicating that instruction tuning helps the model follow instructions to generate a leaderboard table. The second-best open-source LLM Zephyr Beta 7B has 10.8% MR. Interestingly, Mistral 7B was the starting point for the Zephyr Beta 7B model, with distilled direct preference optimization applied to learn a chat model with improved intent alignment. Galactica, which is the only LLM trained specifically on scientific texts (research papers, references, LaTeX, code, DNA sequences, and knowledge bases) performs poorly on both RPLM and LGPLM tasks.

We present the MP scores for GPT-4, Gemini, Mistral Instruct 7B, and Zephyr Beta 7B, which computes the precision of correctly generated methods in the LLM-generated leaderboard. Mistral Instruct 7B and Zephyr Beta 7B generate long text with multiple repetitive titles, leading to low MP compared to GPT-4. Gemini, on the other hand, generates reasonable-looking method names, but

Model	MR	MP	SP
7 B Models			
Falcon	0.010	-	-
Falcon Instruct	0.002	-	-
Galactica	0.000	-	-
LLama 2	0.024	-	-
LLama 2 Chat	0.077	-	-
Mistral	0.351	-	-
Mistral Instruct	20.423	5.75	2.13
Vicuna	0.023	-	-
Zephyr Beta	10.870	1.49	1.81
13 B Models			
LLama 2	0.014	-	-
LLama 2 Chat	0.020	-	-
Vicuna	0.061	-	-
Closed Models			
Gemini Pro	3.382	2.73	13.87
GPT-4	25.244	17.14	13.06

Table 5: Performance of LLMs on the LGPLM task.

often incorrect, leading to low MP. GPT-4 has the highest precision, with roughly generating 17% correct method names on average. However, it still indicates that the model hallucinates and generates several method names that are not present in the papers. SP computes the percentage of correctly generated scores with respect to the correctly generated methods in the leaderboard. Gemini which only has 3.38% method recall, generates 13.87% exactly correct scores for the 3.38% correctly generated methods. GPT-4 has a similar SP of 13.06% for the 25.24% correctly generated methods. Overall, the model GPT-4 only correctly generates 25.24% correct methods, and further, the scores for these methods are incorrect 74.76% times. Upon further inspection, we identify two major challenges in our pipeline design: (i) poor efficiency of the OCR tool (Grobid) in extracting tables data, and (ii) low recall of the BM25 ranker.

5 Related Works

Information Extraction from Scientific Papers: Several works (Viswanathan et al., 2021; Jain et al., 2020; Luan et al., 2018; Augenstein et al., 2017; Luan et al., 2017) extract dataset, task, method, and metric (DTMM) entities for leaderboards. Bedi et al. (2022) annotate 4.9k references as baselines similar to AP-PCN, and Kabongo et al. (2021) curate DTM triples from 4.5k articles. However, these datasets are significantly smaller than our dataset, which contains 1.9M articles, citation and comparison networks from the last 22 years of arXiv, and 70k TDM triples. Parallely, multiple works extract table data from images (Kayal et al., 2022; Zhong et al., 2020), LaTeX (Kardas et al., 2020; Li et al., 2020) and PDFs (Liu et al., 2007). However, IE works extract entities from the text and tables and don't focus much on leaderboard construction due to several challenges such as entity normalization, determination of metric directionality, merging results, and organizing results into leaderboards.

Automatic Leaderboard Generation Hou et al. (2019) presents two datasets and a framework called TDMS-IE for automatically extracting DTM entities and score information from papers. IBM-TDMS (Hou et al., 2019) and ORKG-TDM (Kabongo et al., 2021) use an RTE (recognizing textual entailment) task, where given the paper context, the task determines if TDM tuples are entailed, contradicted, or can't be deduced. However,Kabongo et al. (2023) show that RTE models for DTM identification are not generalizable to new data. In a parallel direction, AxCell Kardas et al. (2020) and Yang et al. (2022) present an end-to-end ML pipeline for extracting results from papers. Singh et al. (2019) consolidate tables from multiple papers into a graph that illustrates performance improvements. Every individual performance edge is extracted from a table with citations to other papers. In contrast, our setting is flexible and realistic, as it starts with a natural language query of <D, T, M>.

6 Conclusion

We curate two dataset collections, APC and PwC-LDB, to construct LEGOBENCH benchmark for automatic scientific leaderboard generation task. LEGOBENCH largely caters to graph-based rankers and language models for leaderboard generation. We design three tasks, with six configurations to comprehensively evaluate diverse systems for automatic scientific leaderboard generation. Across all three tasks, we find that existing models severely lack the capabilities to generate scientific leaderboards leveraging paper texts and paper network datasets. This opens up a new avenue for foundation models to focus on, which also helps the community by reducing the overload of comparing and organizing scientific output by generating leaderboards. In addition to the automatic leaderboard generation problem, our proposed datasets and LEGOBENCH can also be used in traditional tasks such as citation recommendation and intent identification, impact prediction, novelty assessment in review generation, citation count prediction, and venue recommendation for manuscript submission.

7 Limitations

LEGOBENCH benchmark currently only includes ranking tasks, which we plan to extend to various formats such as score extraction and question-answering. Our performance comparison network (AP-PCN) is currently based on the identification of citation patterns in tables in papers. However, it should be noted that not all tables are result comparison tables. Similarly, it is not necessary that all papers whose results are compared are included in tables, and sometimes, results can be compared in the text alone. Such comparison papers won't be present in our performance comparison network. Our leaderboard dataset was curated from PapersWithCode (PwC), and any papers missing in the PwC or arXiv dataset (APC) were excluded from the ranking task dataset. Our ranking task dataset excluded papers with less than three entries and also removed multiple models from the same paper. Hence, the ranking task dataset is not exhaustive. Additionally, our ranking task only focuses on ranking without verifying if the rankings were based on actual scores presented in the candidate papers. Our PwC-LDB dataset is curated from the PaperWithCode repository. An adversary can add incorrect results, leading to poor performance of good ranking models. Similarly, adversaries can also add incorrect results to existing leaderboards to favor a specific group or an individual organization. Lastly, we rely on PwC to correctly map the scores to the corresponding papers, even if they are reported as baselines in other papers.

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*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Manjot Bedi, Tanisha Pandey, Sumit Bhatia, and Tanmoy Chakraborty. 2022. Why did you not compare with that? identifying papers for use as baselines. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I*, pages 51–64. Springer.

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 (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Yuntian Deng, David Rosenberg, and Gideon Mann. 2019. Challenges in end-to-end neural scientific table recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 894–901. IEEE.

Peter Eckersley. 2017. Eff AI Metrics.

Gemini Team Google, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2019. Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. *arXiv preprint arXiv:1906.09317*.

Sarthak Jain, Madeleine Van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. Scirex: A challenge dataset for document-level information extraction. *arXiv preprint arXiv:2005.00512*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Salomon Kabongo, Jennifer D'Souza, and Sören Auer. 2023. Zero-shot entailment of leaderboards for empirical ai research. *arXiv preprint arXiv:2303.16835*.

Salomon Kabongo, Jennifer D'Souza, and Sören Auer. 2021. Automated Mining of Leaderboards for Empirical AI Research. In *Towards Open and Trustworthy Digital Societies: 23rd International Conference on Asia-Pacific Digital Libraries, ICADL 2021, Virtual Event, December 1–3, 2021, Proceedings 23*, pages 453–470. Springer.

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. Axcell: Automatic extraction of results from machine learning papers. *arXiv preprint arXiv:2004.14356*.

Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. 2022. Tables to latex: structure and content extraction from scientific tables. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–10.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2020. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1918–1925.

Xiao Liu, Da Yin, Jingnan Zheng, Xingjian Zhang, Peng Zhang, Hongxia Yang, Yuxiao Dong, and Jie Tang. 2022. OAG-BERT: Towards a unified backbone language model for academic knowledge services. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3418–3428.

Ying Liu, Kun Bai, Prasenjit Mitra, and C Lee Giles. 2007. Tableseer: automatic table metadata extraction and searching in digital libraries. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 91–100.

Patrice Lopez. 2009. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Research and Advanced Technology for Digital Libraries: 13th European Conference, ECDL 2009, Corfu, Greece, September 27-October 2, 2009. Proceedings 13*, pages 473–474. Springer.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602*.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651, Copenhagen, Denmark. Association for Computational Linguistics.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11670–11688, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking : Bringing order to the web. In *The Web Conference*.

Sebastian Ruder. 2018. NLP Progress.

Mayank Singh, Rajdeep Sarkar, Atharva Vyas, Pawan Goyal, Animesh Mukherjee, and Soumen Chakrabarti. 2019. Automated early leaderboard generation from comparative tables. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*, pages 244–257. Springer.

Robert Stojnic, Ross Taylor, Marcin Kardas, Elvis Saravia, Guillem Cucurull, Andrew Poulton, and Thomas Scialom. 2018. Paperwithcode.

Yudong Tao. 2017. Reddit SOTA.

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*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. CitationIE: Leveraging the Citation Graph for Scientific Information Extraction. *arXiv preprint arXiv:2106.01560*.

Sean Yang, Chris Tensmeyer, and Curtis Wigington. 2022. Telin: Table entity linker for extracting leaderboards from machine learning publications. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, pages 20–25.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: Data, model, and evaluation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI*, page 564–580, Berlin, Heidelberg. Springer-Verlag.

A Appendix

A.1 Exponential Growth in Monthly Paper Submissions on arXiv: 1995-2022

Figure 3 presents monthly submissions to arXiv from 1995 to 2002, indicating the exponential growth in submitted manuscripts.

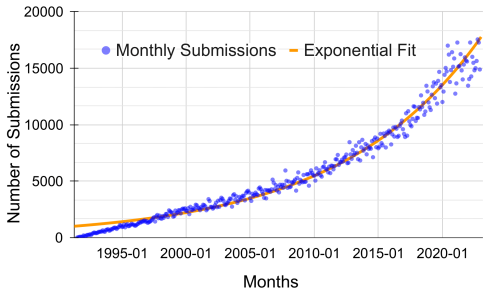
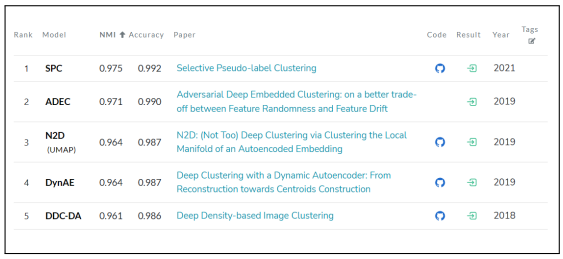


Figure 3: The graph illustrates the exponential growth in the number of papers published monthly on arXiv from 1995 to 2022. This trend showcases the continuous expansion of research and knowledge in the academic community.

A.2 Representative leaderboard taken from PapersWithCode

We present a snapshot of a leaderboard taken from PapersWithCode in Figure 4.



Rank	Model	NMI ↑	Accuracy	Paper	Code	Result	Year	Tags
1	SPC	0.975	0.992	Selective Pseudo-label Clustering			2021	
2	ADEC	0.971	0.990	Adversarial Deep Embedded Clustering: on a better trade-off between Feature Randomness and Feature Drift			2019	
3	N2D (UMAP)	0.964	0.987	N2D: (Not Too) Deep Clustering via Clustering the Local Manifold of an Autoencoded Embedding			2019	
4	DynAE	0.964	0.987	Deep Clustering with a Dynamic Autoencoder: From Reconstruction towards Centroids Construction			2019	
5	DDC-DA	0.961	0.986	Deep Density-based Image Clustering			2018	

Figure 4: A snapshot of the leaderboards from PwC showcasing top-performing models for Image Clustering on MNIST Dataset and ranked based on NMI (Normalized Mutual Information) metric.

A.3 Dataset Statistics

The size and the connected components in the citation and performance comparison network are presented in Table 6. The statistics of different dataset modalities and tasks in the dataset are presented in Table 7 and Table 8 respectively. We finally also present the statistics of the Citation network and the Performance Comparison network for each domain in Table 9.

Citation Network	**Property**
Directed	Yes
Nodes	18,325,578
Edges	59,890,375
$\lvert\mathcal{CC}\rvert$	0.993436
$\lvert\mathcal{SCC}\rvert$	0.001794
Comparison Network	**Property**
Directed	Yes
Nodes	280444
Edges	309483
$\lvert\mathcal{CC}\rvert$	0.518481
$\lvert\mathcal{SCC}\rvert$	0.000043

Table 6: Statistics of the Citation network and Comparison network. $\lvert\mathcal{CC}\rvert$ and $\lvert\mathcal{SCC}\rvert$ denote the number of connected components and strongly connected components, respectively.

Dataset Modality	Frequency
Images	498
Texts	364
Videos	166
Graphs	101
Environment	69
Audio	41
Lidar	39
Medical	34
3d	26
Point Cloud	21
RGB-D	19
Speech	15
Time Series	14
Tracking	12
Tabular	10
Others	44

Table 7: Frequency of various dataset modality of datasets in the AP-LDB dataset. The modality information is taken from PapersWithCode repository.

Task Category	Frequency
Computer Vision	1180
NLP	350
Graphs	109
Playing Games	74
Miscellaneous	42
Medical	40
Time Series	33
Reasoning	30
Speech	28
Knowledge Base	13
Audio	12
Computer Code	11
Robots	7
Music	4
Adversarial	2
Others	146

Table 8: Frequency of various task categories in the AP-LDB dataset. The category information is taken from PapersWithCode repository.

A.4 Baseline System Design

In this section, we present a detailed overview of our baselines for the RPG, RPLM, and LGPLM tasks. For inferencing with LLMs, we use the vLLM library. We had access to a 64 core Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, running Ubuntu 20.04 with 355GB RAM, and one 32GB Nvidia Tesla V100 GPU. We utilized the GPUs for embedding the paper content using the SciBERT, SPECTER, SciNCL, and OAG-BERT, and for LLM inferencing for RPLM and LGPLM tasks. We used NLTK for preprocessing text.

The RPG pipeline is presented in Figure 5. It is provided with the natural language query (consisting of D, T, M) and papers dataset and network dataset from the arXiv Papers' Collection. The candidate retriever generates a set of initial candidate papers that report performance on the <D, T, M> triple. The Ranker module leverages the AP-CN or AP-PCN dataset depending on the RPG task configuration and generates a ranking of the papers.

RPLM pipeline is straightforward as it involves prompting a language model with the natural language query (<D, T, M>) and a list of paper titles. We provide succinct instructions in the prompt to explain the task to LLM. The pipeline is presented in Figure 6 and a representative prompt is presented in Table 10.

Our baseline for LGPLM is a RAG setup. For the natural language query consisting of <D, T, M> details, we select relevant papers. We split these paper texts into chunks and a BM25 ranker function selects top-10 chunks with respect to the query. The top-10 chunks, query, and an instruction are fed to an LLM, and asked to extract leaderboard entries (methods and scores) from the chunks. The pipeline is presented in Figure 7 and a representative prompt is presented in Table 11.

Domain	Papers	PC Network		Citation Network	
		Nodes	Edges	Nodes	Edges
Computer Science	618010	61677	18738	188269	2510609
Economics	5924	639	47	1633	10394
Electrical Engineering and Systems Science	50456	6864	375	14089	95145
Mathematics	530067	5225	20350	171825	2068125
Physics	1123204	360071	37952	360071	4296484
Quantitative Biology	36298	1313	700	11098	97310
Quantitative Finance	14881	699	294	4630	41599
Statistics	157189	9973	3485	50761	87695
TOTAL PAPERS	1938693	613333	4644	107715	5419356

Table 9: Distribution of arXiv papers from Jan 2000-July 2022 in each category. The details of the Performance Comparison network and the Citation network are listed. This data includes the papers not present on arXiv but present in AP-CN.

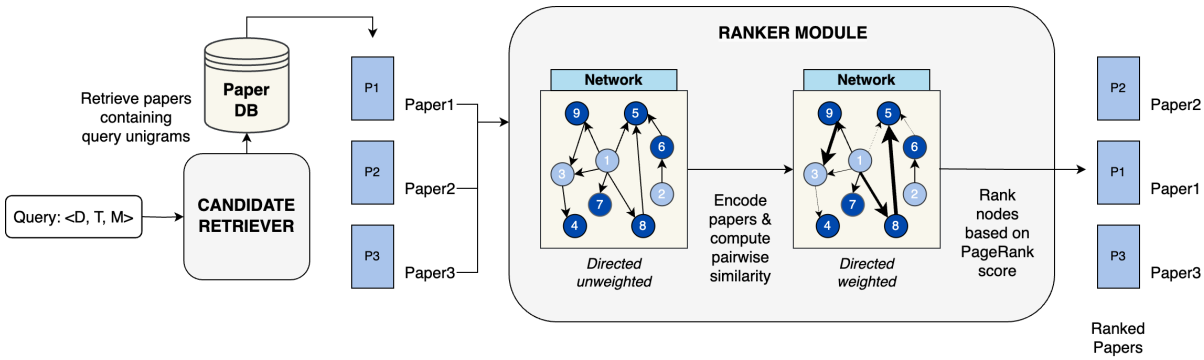


Figure 5: Pipeline for ranking papers with content and graph for leaderboard generation (RPG).

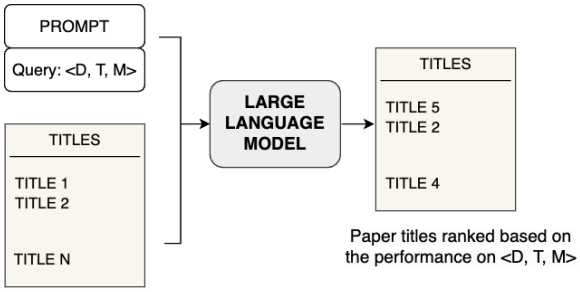


Figure 6: Pipeline for ranking papers by prompting language models (RPLM).

PROMPT:
You are provided with a list of paper titles in the machine learning domain. Your task is to rank them based on their performance on a specific task and dataset using a metric mentioned in the query(the best-performing model should be listed first and the worst should be listed last). Use only the best-performing model proposed in the papers below to compute the ranks. Only include the ranked list of titles in your response and skip any additional text. Query - Rank the performance of the following papers on the <TASK> on dataset <DATASET> using metric <METRIC>. **T1:** ... **T2:** ... **T3:** ... **T4:** ...

Table 10: Prompt Template for Ranking by Prompting Language Models [RPLM]. <TASK>, <DATASET>, and <METRIC> are replaced by appropriate T, D, and M values in the prompt.

PROMPT:
Excerpts: .... You are provided with a dataset, task, and metric. You need to create a leaderboard which lists the performance of various methods on the provided dataset and task using the provided metric. Excerpts from research papers are provided above which report the performance of methods on these task, dataset and metric. Extract the performance from the excerpt to create the leaderboard. The output should be a single table listing each method and performance only. Do not include any explanation or additional text in the output, only include method name and performance scores. Query - List the performance scores of various methods on the <DATASET> dataset on the <TASK> task using metric <METRIC>.

Table 11: Prompt Template for Leaderboard Generation by Prompting Language Models [LGPLM]. <TASK>, <DATASET>, and <METRIC> are replaced by appropriate T, D, and M values in the prompt.

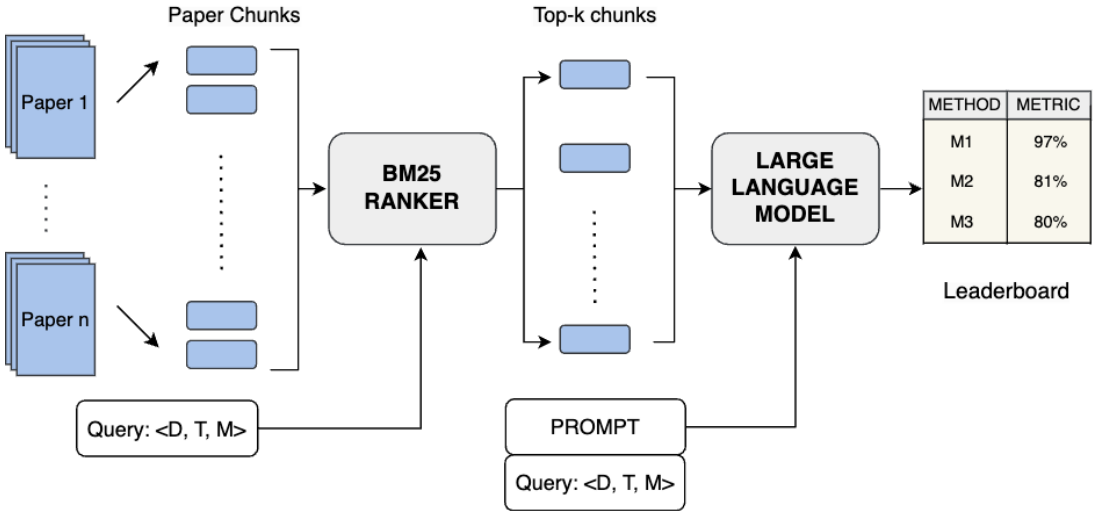


Figure 7: RAG pipeline for leaderboard generation by prompting the language model (LGPLM). Given paper chunks and a query, a BM25 ranker selects the top-10 chunks that are used by an LLM for generating the leaderboard.