Automated Early Leaderboard Generation From Comparative Tables

Mayank Singh[¶], Rajdeep Sarkar[†], Atharva Vyas[†], Pawan Goyal[†], Animesh Mukherjee[†], and Soumen Chakrabarti[‡]

IIT Gandhinagar[¶], IIT Kharagpur[†], and IIT Bombay[‡]

Abstract. A *leaderboard* is a tabular presentation of performance scores of the best competing techniques that address a specific scientific problem. Manually maintained leaderboards take time to emerge, which induces a latency in performance discovery and meaningful comparison. This can delay dissemination of best practices to non-experts and practitioners. Regarding papers as proxies for techniques, we present a new system to automatically discover and maintain leaderboards in the form of partial orders between papers, based on performance reported therein. In principle, a leaderboard depends on the task, data set, other experimental settings, and the choice of performance metrics. Often there are also tradeoffs between different metrics. Thus, leaderboard discovery is not just a matter of accurately extracting performance numbers and comparing them. In fact, the levels of noise and uncertainty around performance comparisons are so large that reliable traditional extraction is infeasible. We mitigate these challenges by using relatively cleaner, structured parts of the papers, e.g., performance tables. We propose a novel *performance improvement graph* with papers as nodes, where edges encode noisy performance comparison information extracted from tables. Every individual performance edge is extracted from a table with citations to other papers. These extractions resemble (noisy) outcomes of 'matches' in an incomplete tournament. We propose several approaches to rank papers from these noisy 'match' outcomes. We show that our ranking scheme can reproduce various manually curated leaderboards very well. Using widely-used lists of state-of-the-art papers in 27 areas of Computer Science, we demonstrate that our system produces very reliable rankings. We also show that commercial scholarly search systems cannot be used for leaderboard discovery, because of their emphasis on citations, which favors classic papers over recent performance breakthroughs. Our code and data sets will be placed in the public domain.

1 Introduction

Comparison against best prior art is critical for publishing experimental research. With the explosion of online research paper repositories like arXiv, and the frenetic level of activity in some research areas, keeping track of the best techniques and their reported performance on benchmark tasks has become increasingly challenging. *Leaderboards*, a tabular representation of the performance

scores of some of the most competitive techniques to solve a scientific task, are now commonplace. However, most of these leaderboards are manually curated and therefore take time to emerge. The resulting latency presents a barrier to entry of new researchers and ideas, trapping "wisdom" about winning techniques to small coteries, disseminated by word of mouth. Thus, automatic leaderboard generation is an interesting research challenge. Recent work [4] has focused on automatic synthesis of reviews from multiple scientific documents. However, to the best of our knowledge, no existing system incorporates *comparative experimental performance* reported in papers into the process of leaderboard generation.

Limitations of conventional information extraction: The ordering of competing techniques in a leaderboard depends on a large number of factors, including the task being solved, the data set(s) used, sampling protocols, experimental conditions such as hyperparameters, and the choice of performance metrics. Further, there are often tradeoffs between various competing metrics, such as recall vs. precision, or space vs. time. In fact, an accurate extraction, in conjunction with all the contextual details listed above, is almost impossible. We argue that conventional table and quantity extraction [2, 10] is neither practical, nor sufficient, for leaderboard induction. In fact, numeric data is often presented as combinations of comparative charts and tables embedded together in a single figure [11]. These may even use subplots with multicolor bars representing baseline and proposed approaches.

Table citations: A practical way to work around the difficult extraction problem is to focus on the relatively cleaner and more structured parts of a paper, viz., tables. Performance numbers are very commonly presented in tables. A prototypical performance table is shown in Singh et al. [11, Figure 3]. Each row shows the name of a competing system or algorithm, along with a citation. (A transposed table style is easily identified with simple rules.) Each subsequent column is dedicated to some performance **metric**. The rows make it simple to associate performance numbers with specific papers. In recent years, tables with citations (here, named **table citations**) and performance summaries have become extremely popular in arXiv.

Performance improvement graphs: We digest a multitude of tables in different papers into a novel **performance improvement graph**. Each edge represents an instance of comparison between two papers, labeled with the ID of the paper where the comparison is reported, the metric (e.g., recall, precision, F1 score, etc.) used for the comparison, and the numeric values of the metric in the two papers. Note that every individual performance edge is extracted from a table with citations to other papers. Each such extracted edge is noisy. Apart from the challenge of extracting quantities from tables and recognizing their numeric types [2, 10], there is no control on the metric names, as they come from an open vocabulary (i.e., the column headers are arbitrary strings). Processing one table is a form of 'micro' reading; we must aggregate these 'micro' readings into a satisfactory 'macro' reading comparing two papers. We propose several reasonable edge aggregation strategies to simplify and featurize the performance improvement graph, in preparation for ranking papers.

Ranking papers using table citation tournaments: Ranking sports teams into total orders, on the basis of the win/loss outcomes of a limited number of matches played between them, has a long history [5, 3, 9]. We adapt two widely-used tournament solvers and find that they are better than some simple baselines. However, we can further improve on tournament solvers using simple variations of PageRank [8, 12] on a graph suitably derived from the tournament. Overall, our best ranking algorithms are able to produce high-quality leaderboards that agree very well with various manually curated leaderboards. In addition, using a popular list of papers spanning 27 different areas of Computer Science, we show that our system is able to produce reliable rankings of the state-of-the-art papers. We also demonstrate that commercial academic search systems like Google Scholar (GS)[1] and Semantic Scholar (SS)[2] cannot be used (and, in fact, are not intended to be used) for discovering leaderboards, because of their emphasis on aggregate citations, which typically favors classic papers over latest performance leaders.

2 Emergence of leaderboards

Experts in an area are usually familiar with latest approaches and their performance. In contrast, new members of the community and practitioners need guidance to identify the best-performing techniques. This gap is currently bridged by "organically emerging" leaderboards that organize and publish the names and the performance scores of the best algorithms in a tabular form. Such leaderboards are commonplace in Computer Science, and in many other applied sciences.

The prime limitation of manually curated leaderboards is the natural latency until the performance numbers in a freshly-published paper are noticed, verified, and assimilated. This can induce delays in the dissemination of the best techniques to non-experts. In this paper, we build an end-to-end system to automate the process of leaderboard generation. The system is able to mine table citations, extract noisy performance comparisons from these table citations, aggregate the micro readings to a smooth macro reading and finally obtain rankings of papers.

In Table 1, we show an example leaderboard generated by our system (details of the system to be discussed later in the subsequent sections) for the PASCAL VOC Challenge (which involves semantic segmentation of images). Similar results reproducing other leaderboards are presented by Singh et al. [11]. We observe that our system is able to find many of the papers present in this human-curated leaderboard. Traditional academic search systems like GS and SS do not fare well in finding leaderboard entries; each returned only seven papers (see Table 1) in their top 50 results retrieved for the query 'semantic segmentation'. Systems that emphasize cumulative citations rather than performance scores cannot be used for leaderboard discovery. Citations to a paper that make incremental improvements, resulting in the best experimental perfor-

[1] https://scholar.google.com/
[2] https://semanticscholar.org/

Table 1: Ability of GS, SS, and our system to recall prominent leaderboard papers for the PASCAL VOC Challenge.

Paper	GS	SS	Our
Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation	×	×	×
Rethinking Atrous Convolution for Semantic Image Segmentation	×	×	✓
Pyramid Scene Parsing Network	×	×	✓
Wider or Deeper: Revisiting the ResNet Model for Visual Recognition	×	×	×
RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation	✓	×	×
Understanding Convolution for Semantic Segmentation	×	×	✓
Not All Pixels Are Equal: Difficulty-aware Semantic Segmentation via Deep Layer Cascade	×	×	✓
Identifying Most Walkable Direction for Navigation in an Outdoor Environment	×	×	×
Fast, Exact and Multi-Scale Inference for Semantic Image Segmentation with Deep ...	×	×	×
DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, ...	×	✓	✓
Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation	✓	×	✓
High-performance Semantic Segmentation Using Very Deep Fully Convolutional Networks	×	×	✓
Higher Order Conditional Random Fields in Deep Neural Networks	×	×	×
Efficient piecewise training of deep structured models for semantic segmentation	✓	✓	✓
Semantic Image Segmentation via Deep Parsing Network	×	✓	✓
Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs ...	✓	×	✓
Pushing the Boundaries of Boundary Detection using Deep Learning	×	×	✓
Attention to Scale: Scale-aware Semantic Image Segmentation	✓	✓	✓
BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks ...	✓	✓	×
Learning Deconvolution Network for Semantic Segmentation	✓	✓	✓
Conditional Random Fields as Recurrent Neural Networks	×	×	×
Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation	×	×	✓
Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures ...	×	×	×
Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs	×	✓	✓
Global Deconvolutional Networks for Semantic Segmentation	×	×	×
Convolutional Feature Masking for Joint Object and Stuff Segmentation	×	×	×

mance, may never catch up with the seminal paper that introduced a general problem or technique.

3 Limits of conventional table information extraction

Performance displays are implicitly connected to a complex context developed in the paper, including the task, the data set, choice of training and test folds, hyperparameters and other experimental settings, performance metrics etc. Millions of reviewer hours are spent each year weighing experimental evidence based on the totality of the experimental context. "Micro-reading" one table at a time is not likely to replace that intellectual process. Beyond contextual ambiguities, there are often trade-offs between different metrics like space vs. time, recall vs. precision, etc. In summary, leaderboard induction is not merely a matter of accurately extracting performance numbers and numerically comparing them.

One way to partly mitigate the above challenges is to use relatively cleaner, structured parts of the papers, e.g., single tables or single charts. We focus on tables in our first-generation system. However, with advanced visual chart mining and OCR [7, 1, 6], we can conceivably extend the system to charts as well.

We concentrate on (the increasing number of) tables that also cite papers, which are surrogates for techniques. Table 2 shows the average number of citations in a paper p that occur in tables, against the year of publication of p. Clearly, there is a huge surge in the use of table citations in the last five years, which further motivates us to exploit them for building our system.

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Average	0.0	0.0	0.12	0.17	0.082	0.18	0.40	0.46	0.57	1.04	3.22	3.61	4.06

Table 2: Average number of table citations made by an arXiv paper between 2005 and 2017.

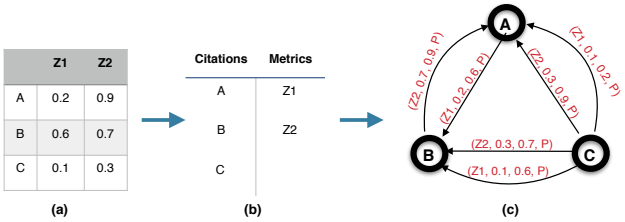


Fig. 1: First table extraction step toward performance tournament graph construction: (a) An example table present in paper P comparing three methods, A, B and C, for two evaluation metrics, $Z1$ and $Z2$. (b) Unique citations to the methods as well the evaluation metrics used are extracted, and (c) an abstract performance tournament graph is constructed.

4 Performance improvement graph

4.1 Raw performance improvement graph

The performance improvement graph $G(V, E, Z)$ is a directed graph among a set of research papers V that are compared against each other. Here, Z represents the set of all the evaluation metrics. An edge between two papers (A, B) (see Figure 1) is annotated with four-tuple (z, v_1, v_2, P), where $z \in Z$, v_1 and v_2 represent the metric value ('recall', 'F1', 'time') and lower and higher performing papers respectively. P denotes the paper that compared A and B. The directionality of an edge e ($e \in E$) is determined by the performance comparison between two endpoints. The paper with lower performance points toward better performing paper. Simple heuristic rules are used to orient the edges. E.g., large F1 but small running times[3] are preferred. Figure 1 shows a toy example of the construction of a raw performance improvement graph from an extracted table.

One table provides just one noisy comparison signal between two papers or techniques. Although table citations allow us to make numerical comparisons, there is no guarantee of the same data set or experimental conditions across different tables, leave alone different papers. Therefore, we process the raw performance improvement graph in two steps:

Local sanitization: All directed edges connecting a pair of papers in the raw performance improvement graph are replaced with one directed edge in the sanitized performance improvement graph. This is partly a denoising step, described through the rest of this section (4.2).

Global aggregation: In section 5, we present and propose various methods of analyzing the sanitized performance improvement graph to arrive at a total order for the nodes (papers) to present in a synthetic leaderboard.

[3] 'Time' is ambiguous by itself: a long time on battery but short training time are preferred. Our system is meant to take such errors it might make in stride.

4.2 Sanitized performance improvement graph

Relative edge improvement (REI) distribution: One unavoidable characteristic of the raw performance improvement graph is the existence of noisy edges from incomparable or botched extractions. We define

$$\text{REI}_z(u, v) = \frac{v_z - u_z}{u_z} \tag{1}$$

where (u, v) represents a directed edge from paper u to v; u_z and v_z denote performance scores of paper u and v respectively against a metric z. As described in previous section, u_z is lower than v_z.

We computed REIs from four leaderboards described in Section 6.1. These improvement scores are computed by considering all pairs of papers present in the respective leaderboards. We note that less than 0.5% of the edges have REI above 100%. In contrast, manual inspection of various erroneously extracted edges revealed that their REI was much larger than 100%. Therefore, we sanitize the raw performance improvement graph by pruning edges having improvement scores larger than 100%. This simple thresholding yielded graphs as clean as by using supervised learning (details omitted) to remove noisy edges.

Sanitizing multi-edges: Every comparison creates a directed edge with different tuple value. A pair of papers can be compared in multiple tables, resulting in (anti-) parallel edges or multi-edges. Two directed edges are termed as *anti-parallel* if they are between the same pair of papers, but in opposite directions. Whereas, two directed edges are said to be *parallel* if they are between the same pair of papers and in the same direction. In Figure 1c, two parallel edges exist between papers B and C and two anti-parallel edges exist between papers A and B.

Multiple strategies can be utilized to summarize and aggregate multi-edges into a condensed tournament graph. We consider the following variations. Note that all of these are directed graphs. In each case, we discuss if and how a directed edge (i, j) is assigned a summarized weight.

UNW — Unweighted Graph: The simplest variant preserves the directed edges without any weights. This is equivalent to giving a weight of 1 for each of these directed edge (i, j), if there is any comparison.

ALL — Weighted graph (total number of comparisons): This variation uses the total number of comparisons between two papers p_i and p_j as the weights of the directed edge. Thus, each time an improvement is reported, it is used as an additional vote to obtain the edge weight.

SIG – Sigmoid of actual improvements on edges: This variation takes into account the sigmoid value of the actual improvement score. If paper u having a score of u_z on a specific metric z, improves upon paper v which has a score of v_z in the same table and same metric, we compute the improvement score using Eq. (1). We then pass this score through a sigmoid function of the form:

$$\sigma_z(u, v) = \frac{1}{1 + e^{-\text{REI}_z(u,v)}} \tag{2}$$

To combine the multiple improvement scores of u over v on different metrics and, thereby, obtain the edge weights, we use the following two techniques.

Max: We set the weight of the edge pointing from v to u as the maximum of all the sigmoid values of the improvement scores across the different metrics.
Average: We set the weight of the edge pointing from v to u as the average of all the sigmoid values of the improvement scores across the different metrics.

Dummy winner and loser nodes: In the tournament ranking literature that we shall discuss in the next section, the most prominent factor that guarantees convergence is that the tournament must be connected. However, performance tournament graphs are mostly disconnected due to extraction inaccuracies, incomplete article collection, etc. Therefore, we introduce a dummy node that either wins or loses over all other nodes in the graph. A dummy node has a suitably directed edge to every other node.

5 Mining sanitized performance improvement graphs

In this section, we explore several ranking schemes to select the most competitive papers by analyzing the sanitized performance graph. We begin with basic baselines, then explore and adapt the tournament literature, and finally present adaptations of PageRank-style algorithms. Solving an incomplete tournament over n teams means to assign each team a score or rank inducing a total order over them, and presents a natural analogy with incomplete pairwise observations. The literature on tournaments seeks to extrapolate the anticipated outcome of a match between teams i and j (which was never played, say) in terms of the statistics of known outcomes, e.g., i defeated k and k defeated j.
Sink nodes: We can ignore the numeric values in table cells and regard each table as comparing some papers, a pair at a time, and inserting an edge from paper p_1 to paper p_2 if the table lists a better (greater or smaller depending on metric) number against p_2 than p_1. In such a directed graph, sink nodes that have no out-links are locally maximal. Thus, the hunt for leaders may be characterized as a hunt for sink nodes. We do not expect this to work well, because our graphs contain many biconnected components, thanks to papers being compared on multiple metrics.
Cocitation: An indirect indication that a paper has pushed the envelope of performance on a task is that it is later compared with many papers. We can capture this signal in a graph where nodes are papers, and an edge and its reverse edge (both unweighted) are added between papers p_1 and p_2 if they are cited by any paper. Edges in both directions are added without considering the numbers extracted from the tables.
Linear tournament: As described earlier, incomplete tournament presents a natural analogy to performance comparisons. [9] started with an incomplete tournament matrix M where $m_{ij} = m_{ji}$ is the number of matches played between teams i and j. $\boldsymbol{m} = (m_i)$ where $m_i = \sum_j m_{ij}$ is the number of matches played by team i. Abusing the division operator, let $\bar{M} = M/\boldsymbol{m}$ denote M after normalizing rows to add up to 1.

Of the m_{ij} matches between teams i and j, suppose i won r_{ij} times and j won $r_{ji} = m_{ij} - r_{ij}$ times. Then the *dominance* of i over j is $d_{ij} = r_{ij} - r_{ji}$ and the dominance of j over i is $d_{ji} = r_{ji} - r_{ij} = -d_{ij}$. Setting the dominance of a team

over itself as zero in one dummy match, we can calculate the average dominance of a team i as $\bar{d}_i = \left[\sum_j d_{ij}\right] \Big/ \left[\sum_j m_{ij}\right]$, and this produces a reasonable ranking of the teams to a first approximation, i.e., up to "first generation" or direct matches. To extrapolate to "second generation" matches, we consider all (i, k) and (k, j) matches, which is given by the matrix M^2. Third generation matches are likewise counted in M^3, and so on. [3] showed that a meaningful scoring of teams can be obtained as the limit $\lim_{T\to\infty} \sum_{t=0}^{T} \bar{M}^t \cdot \bar{\boldsymbol{d}}$, where $\bar{\boldsymbol{d}} = (\bar{d}_i)$.

Exponential tournament: The exponential tournament model [5] is somewhat different, and based on a probabilistic model. Given $R = (r_{ij})$ as above, it computes row sums $\rho_i = \sum_j r_{ij}$. Let $\boldsymbol{\rho} = (\rho_i)$ be the empirically observed team scores. Again, we can sort teams by decreasing ρ_i as an initial estimate, but this is based on an incomplete and noisy tournament. Between teams i and j there are (latent/unknown) probabilities $p_{ij} + p_{ji} = 1$ such that the probability that i defeats j in a match is p_{ij}. Then the MLE estimate is $p_{ij} = r_{ij}/m_{ij}$. [5] shows that there exist team 'values' $\boldsymbol{v} = (v_i)$ such that $\sum_i v_i = 0$ and

$$\rho_i = \sum_j m_{ij} p_{ij} = \sum_j \frac{m_{ij}}{1 + \exp(v_j - v_i)}. \tag{3}$$

Here M and $\boldsymbol{\rho}$ are observed and fixed, and $\boldsymbol{v}$ are variables. Values $\boldsymbol{v}$ can be fitted using gradient descent. Once the matrix $\boldsymbol{P} = (p_{ij})$ is thus built, it gives a consistent probability for all possible permutations of the teams. In particular, $\prod_j p_{ij}$ gives the probability that i defeats all other teams (marginalized over all orders within the other teams j). Sorting teams i by decreasing $\prod_j p_{ij}$ is thus a reasonable rating scheme.

PageRank: PageRank computes a ranking of the competitive papers in the (suitably aggregated) tournament graph based on the structure of the incoming links. We utilize standard PageRank implementation[4] to rank nodes in the directed weighted tournament graph. We found best results (see Table 5) when damping factor (α) is set at 0.90. We run this weighted variant of PageRank on each induced tournament graph corresponding to each query. The induced tournament graph consists of papers (P) relevant to the query along with the papers compared with P. These candidate response papers are ordered using PR values. These scores can also be used for tie-breaking sink nodes.

6 Experiments

6.1 Datasets

***ArXiv* dataset:** We downloaded (in June 2017) the entire arXiv document source dump but restricted this study to the field of Computer Science. Table 3 shows statistics of arXiv's Computer Science papers. ArXiv mandates uploading the source of DVI, PS, or PDF articles generated from LaTeX code resulting in a large volume of papers (1,181,349 out of 1,297,992 papers) with source code.

Preprocessing and extracting table citations: The curation process involves several sub-tasks such as reference extraction, reference mapping, table extraction, collecting table citations, performance metrics extraction and edge

[4] https://networkx.github.io

Table 3: Salient statistics about the arXiv and Computer Science data sets.

Full	Year range	1991–2017
	papers	1,297,992
	papers with LaTeX code	1,181,349
	Total fields	9

Comp. Science	Number of papers	107,795
	Year range	1993–2017
	Total references	2,841,554
	Total indexed papers	1,145,083
	Total tables	204,264
	Total table citations	98,943
	Unique extracted metrics	14,947

orientation. Due to space constraints, we present detailed description and evaluation of each sub-tasks elsewhere [11].

State-of-the-art deep learning papers: A representative example from the rapidly growing and evolving area of deep learning is https://github.com/sbrugman/deep-learning-papers. The website contains state-of-the-art (SOTA) papers on malware detection/security, code generation, NLP tasks like summarization, classification, sentiment analysis etc., as well as computer vision tasks like style transfer, image segmentation, and self-driving cars. This Github repository is very popular and has more than 2,600 stargazers and has been forked 330 times. The repository notes 27 different popular topics shown in Table 4. The table also shows that the SOTA papers curated by knowledgeable experts rarely find a place in the top results returned by the two popular academic search systems — GS and SS. To be fair, these systems were not tuned to find SOTA papers, but we argue that this is an important missing search feature. As fields saturate and stabilize, citations to "the last of the SOTA papers" may eclipse citations to older ones, rendering citation-biased ranking satisfactory. But we again argue that recognizing SOTA papers quickly is critical to researchers, especially new comers and practitioners.

Organic leaderboards: We identify manually curated leaderboards that compare competitive papers on specific tasks. The four popular leaderboards that we choose for our subsequent experiments are (i) The Stanford Question Answering Dataset (SQuAD)[5], (ii) Pixel-Level Semantic Labeling Task (Cityscapes)[6], (iii) VOC Challenge (PASCAL)[7], and (iv) MIT Saliency (MIT-300)[8]. Each leaderboard consists of several competitive papers compared against multiple metrics. For example, the SQuAD leaderboard consists of 117 competitive papers compared against two metrics 'Exact Match' and 'F1 score'. The tasks mostly include topics from natural language processing (e.g., question answering) and image processing (e.g., semantic labeling, image segmentation and saliency prediction).

6.2 Ranking state-of-the-art papers

Table 5 shows comparisons between Google Scholar (GS), Semantic Scholar (SS), and several ranking variations implemented in our testbed. Recall@10,

[5]https://rajpurkar.github.io/SQuAD-explorer/
[6]https://www.cityscapes-dataset.com/benchmarks/
[7]https://goo.gl/6xTWxB
[8]http://saliency.mit.edu/results_mit300.html

Table 4: Recall of human-curated state-of-the-art (SOTA) deep learning papers within top-10 and top-20 responses from two popular academic search engines (Google Scholar and Semantic Scholar). Both systems show low visibility of SOTA papers.

		Code Generation	Malware Detection	Summarization	Taskbots	Text Classification	Question Answering	Sentiment Analysis	Machine Translation	Chatbots	Reasoning	Gaming	Style Transfer	Object Tracking	Visual Q&A	Image Segmentation	Text Recognition	Brain Comp. Interfacing	Self Driving Cars	Object Recognition	Logo Recognition	Super Resolution	Pose Estimation	Image Captioning	Image Compression	Image Synthesis	Face Recognition	Audio Synthesis	**Total**
#SOTA		7	3	3	2	15	1	2	6	2	1	14	6	1	1	15	6	3	2	30	4	5	4	9	1	9	8	6	166
GS	Top-10	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0	1	0	0	0	0	6(3.6%)
	Top-20	0	0	0	0	1	0	0	1	0	0	0	3	0	1	1	1	0	1	1	0	0	0	1	0	0	0	1	12 (7.2%)
SS	Top-10	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	0	1	1	0	0	0	1	0	0	1	0	7 (4.2%)
	Top-20	0	0	0	0	0	0	0	1	0	0	0	2	0	1	1	1	0	1	1	0	1	0	1	0	0	1	0	11 (6.6%)

Recall@20, NDCG@10, and NDCG@20 are used as the evaluation measures, averaged over the 27 topics shown in Table 4. Since our primary objective is to find competitive prior art, recall is more important in case of Web search, where precision at the top (NDCG) is paid more attention.

Given the complex nature of performance tournament ranking, our absolute recall and NDCG are modest. Among naive baselines, sink node search led to generally worst performance, which was expected. The numeric comparison is slightly better, but not much.

GS and SS are mediocre as well. Despite the obvious fit between our problem and tournament algorithms, they are surprisingly lackluster. In fact, many of the tournament variants lose to simple cocitation. PageRank on unweighted improvement graphs performs beyond cocitation. However, the "sigmoid" versions of PageRank improve upon the unweighted case, almost doubling the gains beyond GS and SS, and are clearly the best choice.

6.3 Leaderboard generation

In this section, we demonstrate our system's capability to automatically generate task-specific leaderboards. We utilize four manually curated leaderboards for this study. Automatic leaderboard generation procedure is divided into two phases:

Obtaining list of candidate papers relevant to a task: We, first, obtain a list of candidate papers relevant to a given task. We utilize textual information such as title and abstract to find relevant candidate papers. These candidate papers are further ranked by utilizing best performing PageRank schemes (described in section 6.2). We consider top-50 ranked results and show comparisons between Google Scholar (GS), Semantic Scholar (SS), and top-3 high performing PageRank variations against two evaluation measures — Recall@50 and NDCG@50 — in Table 6. As expected, GS and SS performed poorly for

Table 5: Comparison between several ranking schemes. Recall@10, Recall@20, NDCG@10, NDCG@20 measures are averaged over the 27 tasks (queries). OS: Online Systems; LT: Linear Tournament; ET: Exponential Tournament; ALL: Weighted graph (total number of comparisons); UNW: Unweighted directed performance graph; SIG: Sigmoid of the actual performance improvement; DW: Dummy Winner; DL: Dummy Loser, DCC: Dense co-citation, NC: Numeric comparison.

		OS		LT				ET				PageRank				Sink	BS	
		GS	SS	DW	DL	DW	DL	DW	DL	DW	DL	UNW	ALL	Avg.	Max.	ALL	DCC	NC
				ALL		SIG		ALL		SIG		SIG						
T-10	Recall %	7.38	7.84	4.63	4.63	1.8	1.93	1.7	2.31	1.7	1.7	19.35	16.86	19.35	19.35	0.62	12.91	6.73
	NDCG	0.073	0.065	0.029	0.029	0.016	0.019	0.027	0.024	0.02	0.02	0.151	0.131	0.154	0.149	0.009	0.142	0.036
T-20	Recall %	10.48	10.08	5.86	5.86	6.5	6.63	4.17	2.93	2.93	2.93	21.74	21.95	22.36	22.09	0.62	19.25	7.35
	NDCG	0.086	0.074	0.034	0.034	0.028	0.03	0.036	0.026	0.025	0.025	0.159	0.151	0.164	0.159	0.009	0.152	0.037

Table 6: Recall@50 and NDCG@50 measures for four leaderboards. Green cells indicate best scores and red cells indicate worst scores.

Leaderboard name	GS		SS		PageRank UNW		PageRank SIG (Avg)		PageRank SIG (Max)	
	Recall (%)	NDCG	Recall (%)	NDCG	Recall (%)	NDCG	Recall (%)	NDCG	Recall (%)	NDCG
SQuAD	0	0	7.14	0.014	21.42	0.206	21.42	0.205	14.29	0.177
Cityscapes	25	0.067	37.5	0.159	62.5	0.303	62.5	0.310	62.5	0.295
PASCAL	26.92	0.12	26.92	0.179	57.69	0.497	57.69	0.500	57.69	0.502
MIT-300	42.86	0.115	14.28	0.036	50.00	0.465	50.00	0.437	50.00	0.438

all of the four leaderboards. PageRank variations have almost double the gains beyond GS and SS and are clearly the best choice. Some generated leaderboards are listed in Singh et al. [11].

Ranking candidate papers to generate leaderboard: Next, we compute the correlation between ranks in generated leaderboards with the ground-truth ranks obtained from the organic leaderboards. Table 7 presents the Spearman's rank correlation of rankings produced by PageRank variations, UNW, SIG (Avg) and SIG (Max), with the corresponding ground-truth rankings for the four leaderboards. SQuAD shows the highest correlation (0.94 for F1 and 0.89 for EM) for all of the three PageRank variations. CityScapes and PASCAL also exhibit impressive correlation coefficients for all the PageRank variants. For the MIT-300 leaderboard, while the correlation coefficient is decent for the SIM metric it is somewhat low for the AUC metric. The reason for the low correlation is existence of multiple weakly connected components. A local winner in one component is affecting the global ranks across all components.

Name	Nodes	Metric	UNW	SIG (AVG)	SIG (MAX)
SQuAD	9	F1	0.94	0.94	0.94
		EM	0.89	0.89	0.89
CityScapes	7	iIoU	0.7	0.7	0.7
PASCAL	26	AP	0.57	0.57	0.57
MIT-300	9	AUC	0.23	0.23	0.23
		SIM	0.53	0.45	0.45

Table 7: Spearman's rank correlation of rankings produced by UNW, SIG (Avg) and SIG (Max) with the corresponding ground-truth rankings for the four leaderboards.

6.4 Effect of graph sanitization

As described in section 4.2, graph sanitization is a necessary preprocessing step. In this section, we present several real examples that resulted in greater visibility of state-of-the-art after sanitization. As representative examples, we consider two tasks, "image segmentation" and "gaming", to show how graph sanitization results in noise reduction in the performance improvement graphs. We find several state-of-the-art papers that performed poorer than a competitive paper with high improvement score (>700%). This anomaly resulted in the poorer visibility of the state-of-the-art papers in top ranks. However, after sanitization, the visibility gets improved. For example, Table 8 shows four examples of high improvement edges whose removal resulted in the higher recall of the state-of-the-art papers.

Table 8: Effect of graph sanitization. The first two edges correspond to the task of "image segmentation" and the last two to the task of "gaming". Removal of these edges resulted in higher visibility of SOTA papers.

Source	Destination	Improvement %	Back-edge (Y/N)
1511.07122	1504.01013	775	Y
1511.07122	1511.00561	6597	Y
1611.02205	1207.4708	4012.3	N
1412.6564	1511.06410	928.8	N

6.5 Why is PageRank better than tournaments?

PageRank variations performed significantly better than tournament variations. Several assumptions of tournament literature do not hold true for scientific performance graphs; for instance, existence of disconnected components is a common characteristic of performance graphs. Unequal number of comparisons between a pair of papers in performance graphs is another characteristic that demarcates it from the tournament settings. We observe that in a majority of task-specific performance graphs, tournament-based ranking scheme is biased toward papers with zero out-degrees. Therefore the tournaments mostly converge to the global sinks; in fact, we observe more than half of the tournament based top-ranked papers are sink nodes. This is why recall and NDCG in Table 5 for these two methods are close.

7 Conclusion and future scope

We introduce performance improvement graphs that encode information about performance comparisons between scientific papers. The process of extracting tournaments is designed to be robust, flexible, and domain-independent, but this makes our labeled tournament graphs rather noisy. We present a number of ways to aggregate the tournament edges and a number of ways to score and rank nodes on the basis of this incomplete and noisy information. In ongoing work, we are extending beyond LaTeX tables to line, bar and pie charts [7, 1].

Acknowledgment: Partly supported by grants from IBM and Amazon.

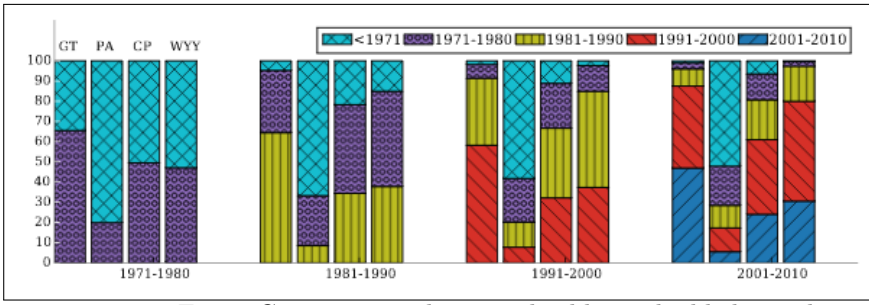
References

1. R. A. Al-Zaidy and C. L. Giles. Automatic extraction of data from bar charts. In *Proceedings of the 8th International Conference on Knowledge Capture*, page 30. ACM, 2015.
2. M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. WebTables: exploring the power of tables on the Web. *PVLDB*, 1(1):538–549, 2008. ISSN 2150-8097. https://doi.org/http://doi.acm.org/10.1145/1453856.1453916. URL http://www.eecs.umich.edu/~michjc/papers/webtables_vldb08.pdf.
3. H. A. David. Ranking from unbalanced paired-comparison data. *Biometrika*, 74(2):432–436, 1987. URL https://academic.oup.com/biomet/article-pdf/74/2/432/659083/74-2-432.pdf.
4. H. Hashimoto, K. Shinoda, H. Yokono, and A. Aizawa. Automatic generation of review matrices as multi-document summarization of scientific papers. In *Workshop on bibliometric-enhanced information retrieval and natural language processing for digital libraries (BIRNDL)*, volume 7, pages 850–865, 2017.
5. T. Jech. The ranking of incomplete tournaments: a mathematician's guide to popular sports. *The American Mathematical Monthly*, 90(4):246–266, 1983. URL http://www.jstor.org/stable/2975756.
6. D. Jung, W. Kim, H. Song, J.-i. Hwang, B. Lee, B. Kim, and J. Seo. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 6706–6717, 2017. ISBN 978-1-4503-4655-9.
7. P. Mitra, C. L. Giles, J. Z. Wang, and X. Lu. Automatic categorization of figures in scientific documents. In *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, pages 129–138. IEEE, 2006.
8. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Manuscript, Stanford University, 1998.
9. C. Redmond. A natural generalization of the win-loss rating system. *Mathematics Magazine*, 76(2):119–126, 2003. URL http://www.jstor.org/stable/3219304.
10. S. Sarawagi and S. Chakrabarti. Open-domain quantity queries on Web tables: Annotation, response, and consensus models. In *SIGKDD Conference*, 2014.
11. M. Singh, R. Sarkar, A. Vyas, P. Goyal, A. Mukherjee, and S. Chakrabarti. Automated early leaderboard generation from comparative tables. *arXiv*, 1802.04538, 2018. URL https://arxiv.org/abs/1802.04538. In ECIR 2019.
12. W. Xing and A. Ghorbani. Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 305–314. IEEE, 2004.

Automated Early Leaderboard Generation From Comparative Tables (Supplementary Material)

Mayank Singh[¶], Rajdeep Sarkar[†], Atharva Vyas[†], Pawan Goyal[†], Animesh Mukherjee[†], and Soumen Chakrabarti[‡]

IIT Gandhinagar[¶], IIT Kharagpur[†], and IIT Bombay[‡]



Simulator	Distance	Turnover	Divergence
GT	–	2.70	–
PA	4.98	0.97	0.77
CP ($p_{\text{copy}} = .5$)	1.97	1.53	0.18
WYY ($\lambda = .11$)	1.67	2.59	0.13

Fig. 1: Comparative charts and tables embedded together in a single figure.

Paper	GS	SS	SIG		UNW
			MAX	AVG	
Reinforced Mnemonic Reader for Machine Reading Comprehension	×	×	×	×	×
Structural Embedding of Syntactic Trees for Machine Comprehension	×	×	×	×	×
ReasoNet: Learning to Stop Reading in Machine Comprehension	×	×	×	✓	✓
Bidirectional Attention Flow for Machine Comprehension	×	×	×	×	×
Multi-Perspective Context Matching for Machine Comprehension	×	×	×	×	×
Exploring Question Understanding and Adaptation in Neural-Network-Based ...	×	×	✓	✓	✓
Dynamic Coattention Networks For Question Answering	×	✓	×	×	×
Ruminating Reader: Reasoning with Gated Multi-Hop Attention	×	×	×	×	×
Reading Wikipedia to Answer Open-Domain Question	×	×	×	×	×
Making Neural QA as Simple as Possible but not Simpler	×	×	×	×	×
Learning Recurrent Span Representations for Extractive Question Answering	×	×	✓	✓	✓
Machine Comprehension Using Match-LSTM and Answer Pointer	×	×	×	×	×
Words or Characters? Fine-grained Gating for Reading Comprehension	×	×	×	×	×
End-to-End Answer Chunk Extraction and Ranking for Reading Comprehension	×	×	×	×	×

Table 1: Ability of GS, SS, and our system to recall prominent leaderboard papers for the SQuAD.

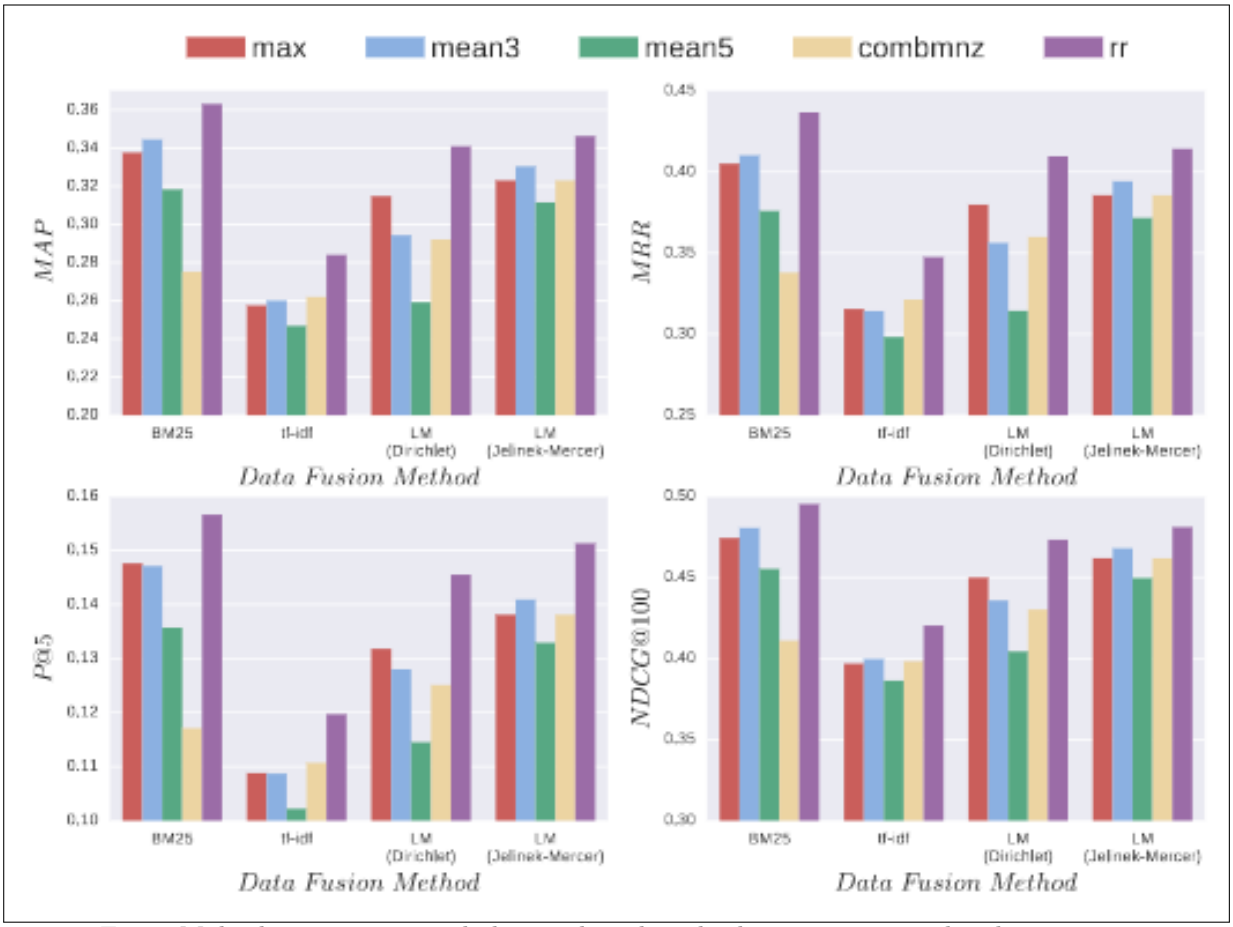


Fig. 2: Multiple comparative subplots with multi-color bars representing baseline papers.

Paper	GS	SS	SIG		UNW
			MAX	AVG	
Rethinking Atrous Convolution for Semantic Image Segmentation	×	×	✓	✓	✓
Wider or Deeper: Revisiting the ResNet Model for Visual Recognition	×	×	×	×	×
RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation	✓	×	×	×	×
Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes	×	×	✓	✓	✓
Multi-level Contextual RNNs with Attention Model for Scene Labeling	×	×	×	×	×
DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution ...	×	✓	✓	✓	✓
Efficient piecewise training of deep structured models for semantic segmentation	✓	✓	✓	✓	✓
SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation	×	✓	✓	✓	✓

Table 2: Ability of GS, SS, and our system to recall prominent leaderboard papers for the CityScape.

Tracker	accuracy	# failures	overlap	speed (fps)
MDNet [9]	0.5620	46	0.3575	1
EBT [41]	0.4481	49	0.3042	5
DeepSRDCF [6]	0.5350	60	0.3033	< 1 *
SiamFC-3s (ours)	0.5335	84	0.2889	**86**
SiamFC (ours)	0.5240	87	0.2743	58
SRDCF [42]	0.5260	71	0.2743	5
sPST [43]	0.5230	85	0.2668	2
LDP [12]	0.4688	78	0.2625	4 *
SC-EBT [44]	0.5171	103	0.2412	–
NSAMF [45]	0.5027	87	0.2376	5 *
StruckMK [3]	0.4442	90	0.2341	2
S3Tracker [46]	0.5031	100	0.2292	14 *
RAJSSC [12]	0.5301	105	0.2262	2 *
SumShift [46]	0.4888	97	0.2233	17 *
DAT [47]	0.4705	113	0.2195	15
SO-DLT [7]	0.5233	108	0.2190	5

Fig. 3: Sample performance numbers in a table with citations. Each row corresponds to a competing algorithm or system, which is associated with a paper cited (green highlighted link) from that row. Each column represents a performance metric.

Paper	GS	SS	SIG		UNW
			MAX	AVG	
DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations	×	✓	×	×	×
A Deep Spatial Contextual Long-term Recurrent Convolutional Network for Saliency Detection	✓	✓	×	×	×
Predicting Human Eye Fixations via an LSTM-based Saliency Attentive Model	×	✓	✓	✓	✓
SalGAN: Visual Saliency Prediction with Generative Adversarial Networks	×	✓	×	×	×
A Deep Multi-Level Network for Saliency Prediction	×	×	×	×	×
Deep Visual Attention Prediction	✓	✓	✓	✓	✓
Shallow and Deep Convolutional Networks for Saliency Prediction	×	×	×	×	×
DeepFeat: A Bottom Up and Top Down Saliency Model Based on Deep Features of ...	✓	✓	✓	✓	✓
Visual saliency detection: a Kalman filter based approach	✓	✓	✓	✓	✓
End-to-end Convolutional Network for Saliency Prediction	×	✓	✓	✓	✓
WEPSAM: Weakly Pre-Learnt Saliency Mode	✓	✓	✓	✓	✓
Visual Language Modeling on CNN Image Representations	✓	✓	×	×	×
Visual saliency estimation by integrating features using multiple kernel learning	✓	✓	×	×	×
Saliency Detection by Forward and Backward Cues in Deep-CNNs	✓	✓	✓	✓	✓

Table 3: Ability of GS, SS, and our system to recall prominent leaderboard papers for the MIT Saliency (MIT-300).