TELIN: Table Entity LINker for Extracting Leaderboards from Machine Learning Publications

Sean T. Yang[*]
University of Washington
Seattle, WA
tyyang38@uw.edu

Curtis Wigington
Adobe Research
College Park, MD
wigingto@adobe.com

Christopher Tensmeyer
Adobe Research
College Park, MD
tensmeye@adobe.com

Abstract

Tracking state-of-the-art (SOTA) results in machine learning studies is challenging due to high publication volume. Existing methods for creating leaderboards in scientific documents require significant human supervision or rely on scarcely available LaTeX source files. We propose Table Entity LINker (TELIN), a framework which extracts (task, model, dataset, metric) quadruples from collections of scientific publications in PDF format. TELIN identifies scientific named entities, constructs a knowledge base, and leverages human feedback to iteratively refine automatic extractions. TELIN identifies and prioritizes uncertain and impactful entities for human review to create a cascade effect for leaderboard completion. We show that TELIN is competitive with the SOTA but requires much less human annotation.

1 Introduction

Advances in the field of Machine Learning (ML) are typically evidenced by producing better empirical results on benchmark datasets. With over 334k AI papers published in 2021 (Zhang et al., 2022), automated approaches to extract and categorize empirical results would help practitioners track progress in the field.

Leaderboard extraction is challenging because there is no universal lexicon, taxonomy, or structure for reporting empirical results in ML publications. New benchmark datasets and tasks are frequently introduced, and established datasets are updated or repurposed for new tasks or metrics. For example, a publication with a table containing numerical results on "ImageNet" could refer to any particular LSVRC challenge year (2010-2017), task (e.g., classification, object detection, localization), number of classes, dataset version, evaluation metric, etc. These necessary details could be specified in table header cells, table captions, paragraphs referencing the table, or elsewhere in the paper. Additionally, ML publications are often only available in PDF format which infrequently explicitly encodes the underlying document paragraph and table structures.

Prior work on scientific leaderboard construction suffer from the following weaknesses:
(1) Unimodal E.g., tables (Singh et al., 2019), citations (Viswanathan et al., 2021), and knowledge bases (Chen et al., 2020). Leaderboard construction can benefit from processing publications holistically rather than as a single data mode.
(2) Requires LaTeX source files (Singh et al., 2019; Kardas et al., 2020). While extracting document structure is easier from LaTeX files than PDF, many publications are only publicly available in PDF.
(3) Closed Taxonomy (Kardas et al., 2020; Hou et al., 2019). Assuming that the names of all datasets, tasks and metrics are known apriori is unrealistic given the rapid pace of the field.
(4) High Manual Effort. State-of-the-art methods (Kardas et al., 2020; Hou et al., 2019) use supervised models that require large and manually-curated training datasets.
(5) Crowd Sourced. E.g., paperswithcode.com generally has precise leaderboard entries, but lack systematic examination of the literature to ensure leaderboard recall.

This work proposes Table Entity LINker (TELIN) as a multi-modal framework that extracts leaderboards from PDF collections of ML publications. TELIN produces (Task, Dataset, Metric, Score) quadruples associated with each paper, which can be grouped and sorted to produce a leaderboard for each (Task, Dataset, Metric) triplet. First, TELIN extracts textual content and tables from all input PDFs and utilizes an off-the-shelf scientific Named Entity Recognition (NER) model, SpERT (Eberts and Ulges, 2020), to identify scientific Named Entities (NE) in the text. Then,

[*]The work was done while the author interned at Adobe Research.

Proceedings of the 1st Workshop on Information Extraction from Scientific Publications, pages 20–25

TELIN matches NEs to table heading cell text to infer the meaning of table cell values and extract quadruples.As additional publications are parsed and more NEs are recognized, TELIN iteratively propagates these labels to previously seen tables and text. TELIN also allows human feedback to label new NEs in table header text. To facilitate this, TELIN intelligently selects tables for human labeling based on their potential for label propagation.

Our evaluation on the PWCLeaderboards dataset (Kardas et al., 2020) shows that TELIN uses significantly less human supervision on PDF inputs to achieve comparable accuracy with the state-of-the-art leaderboard extraction system, Axcell (Kardas et al., 2020), which requires LaTeX source file inputs. While their accuracy is similar, we conclude that TELIN is likely a more practical tool for leaderboard extraction since it requires less human annotation and can be applied to any publication avaiable in PDF.

2 Methodology

Figure 1 illustrates the pipeline of TELIN, whose objective is to extract empirical result quadruples (Task, Dataset, Metric, Score) from a PDF collection of ML publications. We designed TELIN based on the following observations: **(1)** Many scores are presented in tables, but not all tables display scores. **(2)** In most tables, column header text (and separately row text) contain NEs of only a single NE type - e.g., row headers only have model names while col headers contain only metrics. **(3)** NER on individual table cell texts is difficult since the cell text is often only a few words and the NER model is trained on full sentences. However, table cell NEs are lexically the same as or similar to NEs in the main document text, so NEs recognized by a pretrained model in the main text can help identify NEs within cell text. We now explain each step of the pipeline in detail.

(a) Document Decomposition TELIN first converts an unstructured PDF into a structured document using a YOLO-based object detection model (Redmon and Farhadi, 2018) to identify paragraphs, section headings, captions, and table regions. The rows, columns, heading blocks, and cells are then extracted from each table region using the SPLERGE model (**?**) The PDF text can then be associated with the identified regions to form a structured document. While there are errors in this extraction process, we found that the majority of leaderboard errors are not a result of the extraction process.

(b) Scientific NER on Text NER models typically require heavy supervision, so TELIN applies a pre-trained SpERT (Span-based Entity and Relation Transformer) model (Eberts and Ulges, 2020) to the entire main text of each PDF to identify NEs. SpERT is a BERT-based model for NER that is pre-trained on the SCiERC dataset (Luan et al., 2018) of 500 abstracts from 12 AI conference and workshop proceedings. SpERT classifies scientific entities into 5 categories: Task, Method, Evaluation Metric, Material (dataset), and General, which align well with our quadruple schema of (Task, Metric, Dataset, Score).

Since SpERT is trained on full sentences, it performs poorly on short non-sentence text such as table header cell text. Therefore, TELIN takes the NEs from the main text and compares them with table cell text and propagates NE labels for closely matching text.

(c) Strings Matching After identifying NEs from the main text, we perform string matching between these NEs and the text of each non-numeric table cell. One challenge is that acronyms are often used to shorten method, dataset, and metric names. Another challenge is that exact string matches are not guaranteed. To overcome these challenges, TELIN uses a combination of fuzzy search and short text representations to measure string similarity:

$$char\_s(a,b) = \max(t\_dist(a,b), dist(a,b)) \quad (1)$$

$$score = \frac{char\_s + sim(\mathbf{A},\mathbf{B})}{2} \quad (2)$$

where a, b are the two compared strings, $\mathbf{A}, \mathbf{B}$ are their respective Sentence-Bert (Reimers and Gurevych, 2019) feature vectors, $sim()$ is cosine similarity, $dist()$ is the length-normalized Levenshtein string distance, and $t\_dist()$ computes the difference between the tokenized strings. [1] The implementation of computing character level similarity ratio is able to draw comparison between acronyms. The cosine similarity between

[1] We use the WuzzyFuzzy `https://github.com/seatgeek/thefuzz` library. Specifically, we use `fuzz.ratio()` for $dist()$ and `fuzz.token_set_ratio` for $t\_dist()$. Higher number means more similar between strings.

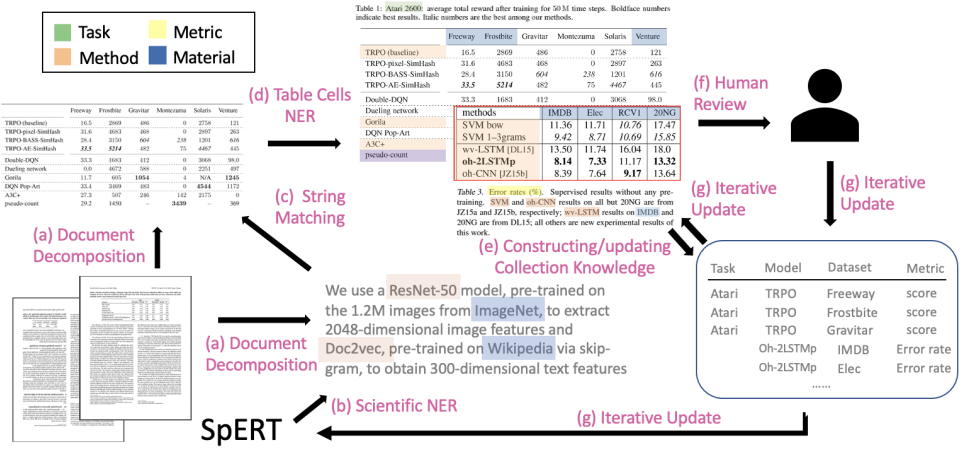

Figure 1: The TELIN framework consumes a collection of Machine Learning publications in PDF and extracts reported results as (Task, Dataset, Metric, Value) quadruples.

the sentence representations indicates how close the strings are semantically.

(d) Table Cells NER SpERT predictions can be inaccurate, and the same or similar strings can be predicted as different entity types. To disambiguate the entity type of a string, we soft-label the string based on majority vote of all predictions for that string across the entire collection text. These labels are then assigned to matching table cell strings. Next, we assign labels to rows and columns of table header cells based on our observation that the type of all cells within a header row/column is often the same. We do this based on cell majority vote and propagate this label to all unlabeled cells. For example, a header row/col with five cells would be labeled when three cells have the same entity type. Then, the 2 remaining unlabled cells would be labeled with this majority type. Finally, the leaderboards are identified when at least three out of the four entities (Task, Dataset, Metric, Model) appear in a table and caption.

(e) Constructing Collection Knowledge We construct a knowledge base from the identified leaderboards and use this as shared knowledge to discover more entities in the documents. The whole collection goes through a few iterations of updates before the human review.

(f) Human Review TELIN integrates a guided human review mechanism to significantly improve the overall entity prediction and quadruple extraction. We compute an influence score E_v for each entity and populate the table with the highest influence score for human annotations. The design philosophy is to prioritize uncertain entities and impactful entities: (1) Uncertain entities have high entropy distributions for predicted entity type distributions from SpERT. (2) Impactful entities are those that can cause a cascade effect for leaderboard completion. A cascade occurs if labeling a string with an entity type and propagating that label to all occurrences of that string throughout the collection would cause a majority labeling of a table header row/col and therefore trigger the propagation of the label to other strings in that table header row/col. Such label propagation may then continue to trigger further cascading of the label.

Note that common entities, such as accuracy (as metric) and COCO (as dataset), do not automatically belong to this category. The proposed design of this task is inspired by identifying influential nodes in a network (Guo et al., 2020; Zhang et al., 2013; Molaei et al., 2020).

First, we compute the uncertainty of a cell by calculate the entropy of the predicted entity type distribution:

$$H_v = \sum_l -p_l \log p_l \tag{3}$$

where p_l is the probability of entity type l for string v. Higher values of H_v indicates higher uncertainty of the entity type.

Then, we compute the uncertainty of the headers.

$$H_h = \sum_{cl \in \Gamma_h} -p_{cl} \log p_{cl} \tag{4}$$

where p_{cl} is the probability of the label l for header h. This step aims to find headers that almost meet the threshold for header labeling.

Next, we construct a heterogeneous network for the purpose of computing the potential of a cell to cause a cascade. Each confirmed entity is a node and edges are formed when two entities appear in the same table header row/col. The "spreading ability" (Guo et al., 2020) of a cell is computed as:

$$H_{uv} = -p_{uv} \log p_{uv} \quad (5)$$

where $p_{uv} = \frac{d_u}{\sum_{l \in \Gamma_v} d_l}$, Γ_v are the immediate neighbors of node v, and d_u is the degree of node u. H_{uv} indicates the spreading ability from node u to node v.

Finally, the influence score of an entity E_v can be acquired by:

$$E_v = H_v + n_{uv} \sum_{u \in \Gamma_v} H_{uv} + \sum_{u \in \Gamma_h} H_h \quad (6)$$

TELIN selects tables including the entities with the highest influence scores for human review. The users are able to confirm or correct the types of the entities on a row/column basis. The user can also label any useful entities in the caption of the table.

(g) Iterative Update The entity type labels from human feedback are treated as ground truth and are used to finetune the SpERT model. The finetuned SpERT model is then used to provide updated NE predictions. This process continues for several iterations until convergence.

3 Experiments and Results

We evaluate TELIN's end-to-end performance on Task, Dataset, Metric, Score (TDMS) quadruple extraction on the PWCLeaderboards (Kardas et al., 2020) task and compare it to the state-of-the-art AxCell model (Kardas et al., 2020). We select AxCell as our main competitor due to its superiority against other existing work (Hou et al., 2019). PWCLeaderboards include 731 papers and 3,445 leaderboards, which include the unique TDMS quadruples in every paper. We follow Kardas et al. for evaluation metrics. We also investigate the performance improvement from the human feedback phase.

End-to-end Performance Table 1 reports the extraction results on PWCLeaderboards dataset. TELIN's performance is comparable to the state-of-the-art results from AxCell with fewer annotations. AxCell includes significant supervision in their pipeline: a table type classification model and a table segmentation model. Both models are trained with 1400 carefully labeled tables. The labeling of these tables require expertise and is time-consuming. The guided human mechanism in TELIN substantially reduces the requirements of human supervision to achieve similar performance as the state-of-the-art.

Table 1: Extraction results on PWCLeaderboards dataset for entire quadruple (TDMS), triple with no score (TDM), and individual entities. The performance of our model is comparable to the state-of-the-art from AxCell with less annotations.

Entity	Micro			Macro		
	P	R	F1	P	R	F1
Axcell (1400 tables)						
TDMS	37.3	23.2	28.7	24.0	21.8	21.1
TDM	67.8	47.8	56.1	47.9	46.4	43.5
Task	70.6	57.3	63.3	60.7	62.6	59.7
Dataset	70.2	48.4	57.3	53.5	52.7	49.9
Metric	68.8	58.5	63.3	58.4	60.4	56.5
Ours (75 tables)						
TDMS	38.3	20.8	26.3	26.6	19.2	21.3
TDM	68.2	45.3	56.5	49.7	43.1	42.5
Task	70.3	53.7	59.2	60.5	57.3	57.1
Dataset	70.9	52.8	59.3	54.7	55.2	53.9
Metric	63.2	57.9	60.2	56.3	55.1	55.4

Analysis of Human Review We further investigate the effect of the feedback by the number of the annotations. Figure 2 shows the impact of the guided human review system. We see improvement in accuracy over the first 50 annotations with convergence after 50 annotations. We observe that the system struggles to identify the 60+ datasets in Atari Games and all the presentation variations of the Accuracy metric without human feedback. The tables with these entities are always among the first for human review.

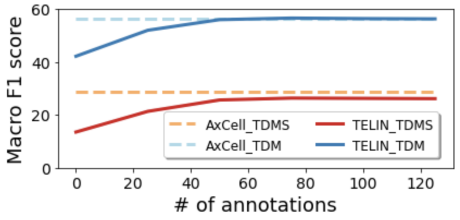

Figure 2: Effect of active learning on the performance. Solid lines are the performance of TELIN on quadruple (Red) and triple (Blue) extraction. Dashed lines are the performance of AxCell as a reference. Human feedback provides performance boost in the first 50 annotations. The performance converges after 50 annotations.

4 Discussion

While TELIN presents promising performance, it still does not exceed the state-of-the-art accuracy in extracting leaderboards from machine learning research papers. Our method relies on the propagation of discoveries from one paper to another. The relatively small data size (731 papers) of PWCLeaderboard dataset limits the capability of TELIN. We will investigate whether introducing more data helps the performance of TELIN. Moreover, unlike existing studies relying on taxonomy of leaderboards known in advance, TELIN operates without any assumptions of taxonomy. We are interested in analyzing the capacity of TELIN for novel taxonomy discovery.

Extracting leaderboards from the scientific papers on the web is an example of integrating artificial intelligence in conceptual modeling (Embley et al., 1998; Olivé, 2007). Conceptual modeling is a vessel for humans to transform the noise in the nature to structured or semi-structured presentations. While automatic machine extraction has been utilized to collect and organize data from a wide variety of sources in conceptual modeling (Embley et al., 1998; Bork, 2022; Nalchigar and Yu, 2018), the role of deep learning and artificial intelligence remains understudied in this field. The design of TELIN is a demonstration of involving artificial intelligence to facilitate conceptual modeling. We hope this effort will invite future studies in this domain.

References

Dominik Bork. 2022. Conceptual modeling and artificial intelligence: Challenges and opportunities for enterprise engineering. In *Enterprise Engineering Working Conference*, pages 3–9. Springer.

Zhiyu Chen, Mohamed Trabelsi, Jeff Heflin, and Brian D Davison. 2020. Towards knowledge acquisition of metadata on ai progress. In *ISWC (Demos/Industry)*, pages 232–237.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020*, pages 2006–2013. IOS Press.

David W Embley, Douglas M Campbell, YS Jiang, Stephen W Liddle, Y-K Ng, DW Quass, and Randy D Smith. 1998. A conceptual-modeling approach to extracting data from the web. In *International Conference on Conceptual Modeling*, pages 78–91. Springer.

Chungu Guo, Liangwei Yang, Xiao Chen, Duanbing Chen, Hui Gao, and Jing Ma. 2020. Influential nodes identification in complex networks via information entropy. *Entropy*, 22(2):242.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5203–5213.

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. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreferencefor scientific knowledge graph construction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*.

Soheila Molaei, Reza Farahbakhsh, Mostafa Salehi, and Noel Crespi. 2020. Identifying influential nodes in heterogeneous networks. *Expert Systems with Applications*, 160:113580.

Soroosh Nalchigar and Eric Yu. 2018. Business-driven data analytics: A conceptual modeling framework. *Data & Knowledge Engineering*, 117:359–372.

Antoni Olivé. 2007. *Conceptual modeling of information systems*. Springer Science & Business Media.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Mayank Singh, Rajdeep Sarkar, Atharva Vyas, Pawan Goyal, Animesh Mukherjee, and Soumen Chakrabarti. 2019. Automated early leaderboard generation from comparative tables. In *European Conference on Information Retrieval*, pages 244–257. Springer.

Vijay Viswanathan, Graham Neubig, and Pengfei Liu. 2021. Citationie: Leveraging the citation graph for scientific information extraction. *arXiv preprint arXiv:2106.01560*.

Daniel Zhang, Nestor Maslej, Andre Barbe, Helen Ngo, Latisha Harry, Ellie Sakhaee, Benjamin Bronkema-Bekker, et al. 2022. The ai index 2022 annual report.

Xiaohang Zhang, Ji Zhu, Qi Wang, and Han Zhao. 2013. Identifying influential nodes in complex networks with community structure. *Knowledge-Based Systems*, 42:74–84.