Kabongo, Salomon; D'Souza, Jennifer; Auer, Sören



# ORKG-Leaderboards: a systematic workflow for mining leaderboards as a knowledge graph

International Journal on Digital Libraries

International Journal on Digital Libraries (2024) 25:41–54
https://doi.org/10.1007/s00799-023-00366-1

# ORKG-Leaderboards: a systematic workflow for mining leaderboards as a knowledge graph

**Salomon Kabongo**[1] · **Jennifer D'Souza**[2] · **Sören Auer**[1,2]

Received: 3 August 2022 / Revised: 10 May 2023 / Accepted: 19 May 2023 / Published online: 15 June 2023


**Abstract**

The purpose of this work is to describe the ORKG-Leaderboard software designed to extract *leaderboards* defined as *task–dataset–metric* tuples automatically from large collections of empirical research papers in artificial intelligence (AI). The software can support both the main workflows of scholarly publishing, viz. as LaTeX files or as PDF files. Furthermore, the system is integrated with the open research knowledge graph (ORKG) platform, which fosters the machine-actionable publishing of scholarly findings. Thus, the systemsss output, when integrated within the ORKG's supported Semantic Web infrastructure of representing machine-actionable 'resources' on the Web, enables: (1) broadly, the integration of empirical results of researchers across the world, thus enabling transparency in empirical research with the potential to also being complete contingent on the underlying data source(s) of publications; and (2) specifically, enables researchers to track the progress in AI with an overview of the state-of-the-art across the most common AI tasks and their corresponding datasets via dynamic ORKG frontend views leveraging tables and visualization charts over the machine-actionable data. Our best model achieves performances above 90% F1 on the *leaderboard* extraction task, thus proving ORKG-Leaderboards a practically viable tool for real-world usage. Going forward, in a sense, ORKG-Leaderboards transforms the *leaderboard* extraction task to an automated digitalization task, which has been, for a long time in the community, a crowdsourced endeavor.

**Keywords** Table mining · Information extraction · Scholarly text mining · Neural machine learning · Semantic networks · Knowledge graphs

## 1 Introduction

Shared tasks—a long-standing practice in the natural language processing (NLP) community—are competitions to which researchers or teams of researchers submit systems that address a specific *Task*, evaluated based on a predefined *Metric* [1]. Seen as "drivers of progress" for empirical research, they attract diverse participating groups from both academia and industry, as well as are harnessed as testbeds for new emerging shared tasks on under-researched and under-resourced topics [2]. Examples of long-standing shared tasks include the Conference and Laboratories of the Evaluation Forum (CLEF)[1] organized at the Conference on natural language learning (CoNLL),[2] the International Workshop on Semantic Evaluation (SEMEVAL),[3] or the biomedical domain-specific BioNLP Shared Task Series [3] and the Critical Assessment of Information Extraction in Biology (BioCreative).[4] Being inherently competitive, shared tasks offer as a main outcome *Leaderboards* that publish participating system rankings.

Inspired by shared tasks, the *Leaderboards* construct of progress trackers is simultaneously taken up for the recording of results in the field of empirical artificial intelligence (AI) at large. Here, the information is made available via the

✉ Salomon Kabongo
kabenamualu@l3s.de

Jennifer D'Souza
jennifer.dsouza@tib.eu

Sören Auer
soeren.auer@tib.eu

1 L3S Research Center, Leibniz University of Hannover, Hannover, Lower-saxony, Germany

2 TIB, Leibniz Information Centre for Science and Technology, Hannover, Lower-saxony, Germany

[1] http://www.clef-initiative.eu/.

[2] https://www.signll.org/conll.

[3] https://semeval.github.io/.

[4] https://biocreative.bioinformatics.udel.edu/tasks/.

traditional scholarly publishing flow as PDFs and preprints, unlike in Shared Tasks where the community is relegated to a list of researchers wherein tracking the dataset creators and individual systems applied is less cumbersome as they can be found within the list of researchers that sign up to organize or participate in the task. On the other hand, general publishing avenues bespeak of a deluge of peer-reviewed scholarly publications [4] and PDF preprints ahead (or even instead) of peer-reviewed publications [5]. This high-volume publication trend problem is only compounded by the diversity in empirical AI research where *Leaderboards* can potentially be searched and tracked on research problems in various fields such as computer vision, time series analysis, games, software engineering, graphs, medicine, speech, audio processing, adversarial learning, etc. Thus, the problem of obtaining completed *Leaderboard* representations of empirical research seems a tedious if not completely insurmountable task.

Regardless of the setup, i.e., from shared tasks or empirical AI research, another problem in the current methodology is the information representation of *Leaderboards* which is often via Github repositories, shared task websites, or researchers' personal websites. Some well-known websites that exist to this end are: PapersWithCode (PwC) [6],[5] NLP-Progress [7], AI-metrics [8], SQUaD explorer [9], Reddit SOTA [10]. The problem with leveraging websites for storing *Leaderboards* is the resulting rich data's lack of machine actionability and integrability. In other words, unstructured, non-machine-actionable information from scholarly articles is converted to semi-structured information on the websites which still unfortunately remain non-machine-actionable. In the broader context of scholarly knowledge, the FAIR guiding principles for scientific data management and stewardship [11] identify general guidelines for making data and metadata machine-actionable by making them maximally Findable, accessible, interoperable, and reusable for machines and humans alike. Semantic Web technologies such as the W3C recommendations resource description framework (RDF) and web ontology language (OWL) are the most widely accepted choice for implementing the FAIR guiding principles [12]. In this context, the open research knowledge graph (ORKG) [13] https://orkg.org/ as a next-generation library for digitalized scholarly knowledge publishing presents a framework fitted with the necessary Semantic Web technologies to enable the encoding of *Leaderboards* as FAIR, machine-actionable data. Adopting semantic standards to represent *Leaderboards* not just *task–dataset–metric* but also related information such as code links, pre-trained models, and so on can be made machine-actionable and consequently queryable. This would directly address the lack of transparency and integration of various results' problems identified in current methods of recording empirical research [1, 2, 14].

This work, taking note of the two main problems around *Leaderboard* construction, i.e., *information capture* and *information representation*, proposes solutions to address them directly. First, regarding information capture, we recognize due to the overwhelming volume of data, now more than ever, that it is of paramount importance to empower scientists with automated methods to generate the *Leaderboards* oversight. The community could greatly benefit from an automatic system that can generate a *Leaderboard* as a *task–dataset–metric* tuple over large collections of scholarly publications both covering empirical AI, at large and encapsulating shared tasks, specifically. Thus, we empirically tackle the *Leaderboard* knowledge mining machine learning (ML) task via a detailed set of evaluations involving large datasets for the two main publishing workflows, i.e., as LaTeX source and PDF, with several ML models. For this purpose, we extend the experimental settings from our prior work [15] by adding support for information extraction from LaTeX code source and compared empirical evaluations on longer input sequences (beyond 512 tokens) for both XLNet and BigBird [16]. Our ultimate goal with this study is to help the digital library (DL) stakeholders to select the optimal tool to implement knowledge-based scientific information flows w.r.t. *Leaderboard*s. To this end, we evaluate four state-of-art transformer models, viz. BERT, SciBERT, XLNet, and BigBird, each of which has its own unique strengths. Second, regarding information representation, ORKG-Leaderboards workflow, is integrated in the knowledge graph-based DL infrastructure of the ORKG [13]. Thus, the resulting data will be made machine-actionable and served via the dynamic ORKG Frontend views[6] and further queryable via structured queries over the larger scholarly KG using SPARQL.[7]

In summary, the contributions of our work are:

1. we construct a large empirical corpus containing over 4000 scholarly articles and 1548 *leaderboards* TDM triples for the development of text mining systems;
2. we empirically evaluate three different transformer models and leverage the best model, i.e., ORKG-Leaderboards $_{XLNet}$, for the ORKG benchmarks curation platform;
3. produced a pipeline that works both with the raw PDF and the LaTeX code source of a research publication.
4. we extended our previous work [15] by empirically investigating our approach with longer input beyond the traditional 512 sequence length limit by BERT-based models, and added support for both mainstreams of research publication PDFs and LaTeX code source.

[5] https://paperswithcode.com.

[6] https://orkg.org/benchmarks.

[7] https://orkg.org/triplestore or https://orkg.org/sparql/.

5. in a comprehensive empirical evaluation of ORKG-Leaderboards for both LaTeX and PDFs based pipelines, we obtain around 93% micro and 92% macro F1 scores which outperform existing systems by over 20 points.

To the best of our knowledge, the ORKG-Leaderboards system obtains state-of-the-art results for the *Leaderboard* extraction defined as *task–dataset–metric* triples extraction from empirical AI research articles handling both LaTeX and PDF formats. Thus, ORKG-Leaderboards can be readily leveraged within KG-based DLs and be used to comprehensively construct *Leaderboards* with more concepts beyond the TDM triples. To facilitate further research, our data[8] and code[9] are made publicly available.

## 2 Definitions

This section defines the central concepts in the *task–dataset–metric* extraction schema of ORKG-Leaderboards. Furthermore, the semantic concepts used in the information representation for the data in the ORKG are defined.

**Task.**

It is a natural language mention phrase of the theme of the investigation in a scholarly article. Alternatively referred to as research problem [17] or focus [18]. An article can address one or more tasks. *Task* mentions being often found in the article Title, Abstract, Introduction, or Results tables and discussion, e.g., question answering, image classification, drug discovery, etc.

**Dataset.**

A mention phrase of the dataset encapsulates a particular *Task* used in the machine learning experiments reported in the respective empirical scholarly articles. An article can report experiments on one or more datasets. *Dataset* mentions are found in similar places in the article as *Task* mentions, e.g., HIV dataset,[10] MNIST [19], Freebase 15K [20], etc.

**Metric.**

Phrasal mentions of the standard of measurement[11] used to evaluate and track the performance of machine learning models optimizing a *Dataset* objective based on a *Task*. An article can report performance evaluations on one or more metrics. *Metrics* are generally found in Results tables and discussion sections in scholarly articles, e.g., BLEU (bilingual evaluation understudy) [21] used to evaluate "machine translation" tasks, $F$-measure [22] used widely in "classification" tasks, MRR (mean reciprocal rank) [23] used to evaluate the correct ordering of a list of possible responses in "information retrieval" or "question answering" tasks, etc.

**Benchmark.**

ORKG *Benchmark*s (https://orkg.org/benchmarks) organize the state-of-the-art empirical research within ORKG *research fields*[12] and are powered in part by automated information extraction supported by the ORKG-Leaderboards software within a human-in-the-loop curation model. A benchmark per research field is fully described in terms of the following elements: research problem or *Task*, *Dataset*, *Metric*, *Model*, and *Code*, e.g., a specific instance of an ORKG benchmark[13] on the "Language Modelling" *Task*, evaluated on the "WikiText-2" *Dataset*, evaluated by "Validation perplexity" *Metric* with a listing of various reported Models with respective Model scores.

**Leaderboard.**

Is a dynamically computed trend-line chart on respective ORKG benchmark pages leveraging their underlying machine-actionable data from the knowledge graph. Thus, *Leaderboard*s depict the performance trend-line of models developed over time based on specific evaluation *Metric*s.

## 3 Related work

There is a wealth of research in the NLP community on specifying a collection of extraction targets as a unified information-encapsulating unit from scholarly publications. The two main related lines of work that are at the forefront are: (1) extracting instructional scientific content that captures the experimental process [24–28]; and (2) extracting terminology as named entity recognition objectives [18, 29–32] to generally obtain a concise representation of the scholarly article which also includes the *Leaderboard* information unit [33–35].

Starting with the capture of the experimental process, [24] proposed an AI-based clustering method for the automatic semantification of bioassays based on the specification of the BAO ontology.[14] In [26], they annotate wet laboratory protocols, covering a large spectrum of experimental biology w.r.t. laboratory procedures and their attributes including materials, instruments, and devices used to perform specific actions as a prespecified machine-readable format as opposed to the ad hoc documentation norm. Within scholarly articles,

[8] https://doi.org/10.5281/zenodo.7419877.

[9] https://github.com/Kabongosalomon/task-dataset-metric-nli-extraction/tree/latex.

[10] https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data.

[11] https://www.merriam-webster.com/dictionary/metric.

[12] https://orkg.org/fields.

[13] https://orkg.org/benchmark/R121022/problem/R120872.

[14] https://github.com/BioAssayOntology/BAO.

such instructions are typically published in the Materials and Method section in Biology and Chemistry fields. Similarly, in [25, 27], to facilitate machine learning models for automatic extraction of materials syntheses reactions and procedures from text, they present datasets of synthesis procedures annotated with semantic structure by domain experts in materials science. The types of information captured include synthesis operations (i.e., predicates), and the materials, conditions, apparatus, and other entities participating in each synthesis step.

In terms of extracting terminology to obtain a concise representation of the article, an early dataset called the FTD corpus [18] defined *focus*, *technique*, and *domain* entity types which were leveraged to examine the influence between research communities. Another dataset, the ACL RD-TEC corpus [29] identified seven conceptual classes for terms in the full-text of scholarly publications in computational linguistics, viz. *Technology and Method*, *Tool and Library*, *Language Resource*, *Language Resource Product*, *Models*, *Measures and Measurements*, and *Other* to generate terminology lists. Similarly, terminology mining is the task of scientific keyphrase extraction. Extracting keyphrases is an important task in publishing platforms as they help recommend articles to readers, highlight missing citations to authors, identify potential reviewers for submissions, and analyze research trends over time. Scientific keyphrases, in particular, of type *Processes*, *Tasks*, and *Materials* were the focus of the SemEval17 corpus annotations [30] which included full-text articles in Computer Science, Material Sciences, and Physics. The SciERC corpus [31] provided a resource of annotated abstracts in artificial intelligence which annotations for six concepts, viz. *Task*, *Method*, *Metric*, *Material*, *Other-Scientific Term*, and *Generic* to facilitate the downstream task of generating a searchable KG of these entities. On the other hand, the STEM-ECR corpus [32] notable for its multidisciplinarity included 10 different STEM domains annotated with four generic concept types, viz. *Process*, *Method*, *Material*, and *Data* that mapped across all domains, and further with terms grounded in the real world via Wikipedia/Wiktionary links. Finally, several works have recently emerged targeting the task of Leaderboard extraction, with the TDM-IE pioneering work [33] also addressing the much harder *Score* element as an extraction target. Later works attempted the document-level information extraction task by defining explicit relations *evaluatedOn* between *Task* and *Dataset* elements and *evaluatedBy* between *Task* and *Metric* [34, 35]. In contrast, in our prior ORKG-TDM system [15] and in this present extended ORKG-Leaderboards experimental report, we attempt the *task–dataset–metric* tuple extraction objective assuming implicitly encoded relations. This simplifies the pipelined entity and relation extraction objectives as a single tuple inference task operating over the entire document. Nevertheless, [34, 35] also defined coreference relations between similar term mentions, which can be leveraged complementarily in our work to enrich the respective *task–dataset–metric* mentions.

## 4 The ORKG-Leaderboards task dataset

### 4.1 Task definition

The *Leaderboard* extraction task addressed in ORKG-Leaderboards can be formalized as follows. Let $p$ be a paper in the collection $P$. Each $p$ is annotated with at least one triple $(t_i, d_j, m_k)$ where $t_i$ is the $i$th *Task* defined, $d_j$ the $j$th *Dataset* that encapsulates *Task* $t_i$, and $m_k$ is the $k$th evaluation *Metric* used to evaluate a system performance on a *Task*'s *Dataset*. While each paper has a varying number of *task–dataset–metric* triples, they occur at an average of roughly 4 triples per paper.

In the supervised inference task, the input data instance corresponds to the pair: a paper $p$ represented as the DocTAET context feature $p_{DocTAET}$ and its *task–dataset–metric* triple $(t, d, m)$. The inference data instance, then is $(c; [(t, d, m), p_{DocTAET}])$ where $c \in \{true, false\}$ is the inference label. Thus, specifically, our *Leaderboard* extraction problem is formulated as a natural language inference task between the DocTAET context feature $p_{DocTAET}$ and the $(t, d, m)$ triple annotation. $(t, d, m)$ is $true$ if it is among the paper's *task–dataset–metric* triples, where they are implicitly assumed to be related, otherwise $false$. The $false$ instances are artificially created by a random selection of inapplicable $(t, d, m)$ annotations from other papers. Cumulatively, *Leaderboard* construction is a multi-label, multi-class inference problem.

#### 4.1.1 DocTAET context feature

The DocTAET context feature representation [33] selects only the parts of a paper where the *task–dataset–metric* mentions are most likely to be found. While the *Leaderboard* extraction task is applicable on the full scholarly paper content, feeding a machine learning model with the full article is disadvantageous since the model will be fed with a large chunk of text which would be mostly noise as it is redundant to the extraction task. Consequently, an inference model fed with large amounts of noise as contextual input cannot generalize well. Instead, the DocTAET feature was designed to heuristically select only those parts of an article that are more likely to contain *task–dataset–metric* mentions as true contextual information signals. Specifically, as informative contextual input to the machine learning model, DocTAET captures sentences from four specific places in the article that are most likely to contain *task–dataset–metric* mentions, viz. the **Doc**ument **T**itle, **A**bstract, first few lines of

the **E**xperimental setup section and **T**able content and captions.

## 4.2 Task dataset

To facilitate supervised system development for the extraction of *Leaderboards* from scholarly articles, we built an empirical corpus that encapsulates the task. *Leaderboard* extraction is essentially an inference task over the document. To alleviate the otherwise time-consuming and expensive corpus annotation task involving expert annotators, we leverage distant supervision from the available crowdsourced metadata in the PwC (https://paperswithcode.com/) KB. In the remainder of this section, we explain our corpus creation and annotation process.

### 4.2.1 Scholarly papers and metadata from the PwC knowledge base

We created a new corpus as a collection of scholarly papers with their *task–dataset–metric* triple annotations for evaluating the *Leaderboards* extraction task inspired by the original IBM science result extractor [33] corpus. The collection of scholarly articles for defining our *Leaderboard* extraction objective is obtained from the publicly available crowdsourced leaderboards PwC. It predominantly represents articles in the natural language processing and computer vision domains, among other AI domains such as Robotics, Graphs, Reasoning, etc. Thus, the corpus is representative for empirical AI research. The original downloaded collection (timestamp 2021-05-10 at 12:30:21)[15] was pre-processed to be ready for analysis. While we use the same method here as the science result extractor, our corpus is different in terms of both labels and size, i.e., number of papers, as many more *Leaderboards* have been crowdsourced and added to PwC since the original work. Furthermore, as an extension to our previous work [15] on this theme, based on the two main scholarly publishing workflows, i.e., as LaTeX or PDF, correspondingly two variants of our corpus are created and their models, respectively, developed.

Recently, publishers are increasingly encouraging paper authors to provide the supporting LaTeX files accompanying the corresponding PDF article. The advantage of having the LaTeX source files is that they contain the original article in plain-text format and thus result in cleaner data in downstream analysis tasks. Our prior ORKG-TDM [15] model was fine-tuned only on the parsed plain-text output of PDF articles wherein the plain text was scraped from the PDF which results in partial information loss. Thus, in this work, we modify our previous workflow deciding to tune one model on LaTeX source files as input data, given the increasing impetus of authors also submitting the LaTeX source code; and a second model following our previous work on plain text scraped from PDF articles.

1. ***LaTeX pre-processed corpus.*** To obtain the LaTeX sources, we queried arXiv based on the paper titles from the 5361 articles of our original corpus leveraged to developed ORKG-TDM [15]. Resultingly, LaTeX sources for roughly 79% of the papers from the training and test datasets in our original work were obtained. Thus, the training set size was reduced from 3753 papers in the original work to 2951 papers in this work with corresponding LaTeX sources. Similarly, the test set size was reduced from 1608 papers in the original work to 1258 papers in this work for which LaTeX sources could be obtained. Thus, the total size of our corpus reduced from 5361 papers to 4209 papers. Once the LaTeX sources were, respectively, gathered for the training and test sets, the data had to undergo one additional step of preprocessing. With the help of pandoc,[16] latex format files were converted into the XML TEI[17] markup format files. This is the required input for the heuristics-based script that produces the DocTAET feature. Thus, the resulting XML files were then fed as input to the DocTAET feature extraction script. The pipeline to reproduce this process is released in our code repository.[18]
2. ***PDF pre-processed corpus.*** For the 4209 papers with LaTeX sources, we created an equivalent corpus but this time using the PDF files. This is the second experimental corpus variant of this work. To convert PDF to plain text, following along the lines of our previous work [15], the GROBID parser [36] was applied. The resulting files in XML TEI markup format were then fed into the DocTAET feature extraction script similar to the LaTeX document processing workflow.

### 4.2.2 *Task–dataset–metric* annotations

Since the two corpus variants used in the empirical investigations in this work are a subset of the corpus in our earlier work [15], the 4209 papers in our present corpus, regardless of the variant, i.e., LaTeX or PDF, retained their originally obtained *task–dataset–metric* labels via distant labeling supervision on the PwC knowledge base (KB).

[15] Our corpus was downloaded from the PwC GitHub repository https://github.com/paperswithcode/paperswithcode-data and was constructed by combining the information in the files *All papers with abstracts* and *Evaluation tables* which included article urls and TDM crowdsourced annotation metadata.

[16] https://pandoc.org/.

[17] https://tei-c.org/.

[18] https://github.com/Kabongosalomon/task-dataset-metric-nli-extraction/tree/main/data.

### 4.3 Task dataset statistics

Our overall corpus statistics are shown in Table 1. The column "Ours-Prior" reports the dataset statistics of our prior work [15] for comparison purposes. The column "Ours-Present" reports the dataset statistics of the subset corpus used in the empirical investigations reported in this paper. The corpus size is the same for both the LaTeX and PDF corpus variants. In all, our corpus contains 4208 papers split as 2946 as training data and 1262 papers as test data. There were 1724 unique TDM triples overall. Note that since the test labels were a subset of the training labels, the unique labels overall can be considered as those in the training data. Table 1 also shows the distinct *Tasks*, *Datasets*, *Metrics* in the last three rows. Our corpus contains 262 *Tasks* defined on 853 *Datasets* and evaluated by 528 *Metrics*. This is significantly larger than the original corpus which had 18 *Tasks* defined on 44 *Datasets* and evaluated by 31 *Metrics*.

#### 4.3.1 DocTAET context feature statistics

Figure 1 shows in detail the variance of the DocTAET Context Feature over three datasets proposed for *Leaderboard* extraction as *task–dataset–metric* triples: (1) Fig. 1a for the dataset from the pioneering science result extractor system [33]; (2) Fig. 1b for the dataset from our prior ORKG-TDM work [15]; (3) Fig. 1c, d for the dataset in our present paper from the Grobid and LaTeX workflows, respectively (column "Ours-Present" in Table 1)).

Both the prior datasets, i.e., the original science result extractor dataset [33] and the ORKG-TDM dataset [15], followed the Grobid processing workflow and reported roughly the same average length of the DocTAET feature. This reflects the consistency preserved in the method of computing the DocTAET feature of between 300 to 400 tokens. Note the ORKG-TDM corpus was significantly larger than the original science result extractor corpus; hence, their DocTAET feature length statistics do not match exactly.

In our present paper, as reported earlier, we use a subset of papers from the ORKG-TDM dataset for which the corresponding LaTeX sources could be obtained to ensure similar experimental settings between the Grobid and LaTeX processing workflows. This is why the DocTAET feature length statistics between the ORKG-TDM dataset (Fig. 1b) and our present dataset in the Grobid processing workflow (Fig. 1c) do not match exactly. Still, we see that they are roughly in similar ranges. Finally, of particular interest is observing the DocTAET feature length statistics that could be obtained from the LaTeX processing workflow introduced in this work (Fig. 1d). Since from the LaTeX processing workflow cleaner plain-text output could be obtained, the corresponding DocTAET feature lengths in many of the papers were longer than all the rest of the datasets considered, which operated in the Grobid processing workflow over PDFs.

## 5 The ORKG-Leaderboards system

This section depicts the overall end-to-end ORKG-Leaderboards, including details on the deep learning models used in our natural language inference (NLI) task formulation.

### 5.1 Workflow

The overall ORKG-Leaderboards workflow as depicted in Fig. 2 includes the following steps:

1. A user provides the article input as either the main ".tex" file or a PDF file.
2. If the input is provided as a ".tex" file, the pandoc script is applied to convert the LaTeX to the corresponding XML TEI marked-up format.
3. Alternatively, if the input is provided as a PDF file, the Grobid parser is applied to obtain the corresponding scraped plain text in the XML xxxx marked-up format.
4. Once the XML xxx marked-up files are obtained, the DocTAET feature extraction script is applied to obtain the paper context representations.
5. Furthermore, if in the training phase, the collection of papers in the training set is assigned their respective *true task–dataset–metric* labels and a random set of "False" *task–dataset–metric* labels.
6. Otherwise, if in the test phase, the query paper is assigned all the *task–dataset–metric* inference targets as candidate labels.
7. Finally, on the one hand, for the training phase, for each of the input file formats, i.e., ".tex" or PDF, an optimal inference model is trained by testing four transformer model variants, viz. BERT, SciBERT, XLNet, and BigBird.
8. On the hand, for the test phase, depending on the input file format, i.e., ".tex" or PDF, the corresponding trained optimal model is applied to the query instance.
9. Finally, from the test phase, the predicted *task–dataset–metric* tuples output are integrated in the ORKG.

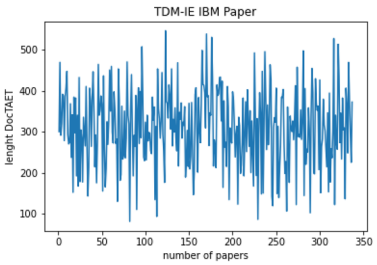
### 5.2 Leaderboards natural language inference (NLI)

To support *Leaderboard* inference [33], we employ deep transfer learning modeling architectures that rely on a recently popularized neural architecture—the transformer [37]. Transformers are arguably the most important architecture for natural language processing (NLP) today since they have shown and continue to show impressive results in several NLP tasks [38]. Owing to the self-attention mechanism in these models, they can be fine-tuned on many down-

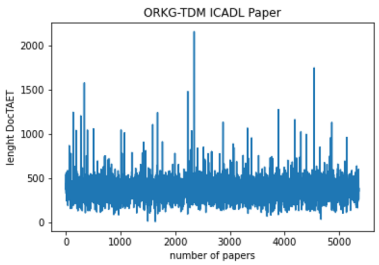
**Table 1** Ours-prior [15] versus ours-present versus the original science result extractor [33] corpora statistics

| | Ours-prior | | Ours-present | | Original | |
|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test |
| Papers | 3753 | 1608 | 2946 | 1262 | 170 | 167 |
| "Unknown" annotations | 922 | 380 | 2359 | 992 | 46 | 45 |
| Total TDM triples | 11,724 | 5060 | 9614 | 4096 | 327 | 294 |
| Avg. number of TDM triples per paper | 4.1 | 4.1 | 4.3 | 4.2 | 2.64 | 2.41 |
| Distinct TDM triples | 1806 | 1548 | 1668 | 1377 | 78 | 78 |
| Distinct *Tasks* | 288 | 252 | 262 | 228 | 18 | 18 |
| Distinct *Datasets* | 908 | 798 | 853 | 714 | 44 | 44 |
| Distinct *Metrics* | 550 | 469 | 528 | 434 | 31 | 31 |

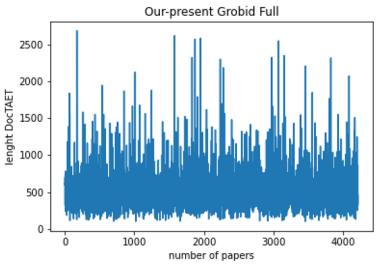
The "unknown" labels were assigned to papers with no TDM triples after the label filtering stage



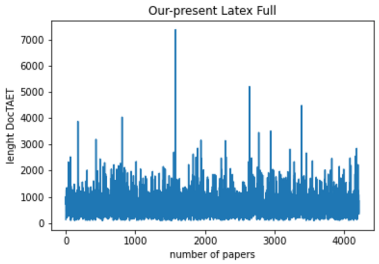

(a) DocTAET feature length in the original science result extractor corpus [33] had a **max**, **min**, and **mean** length of **546**, **81** and **309.45**, respectively




(b) DocTAET feature length in the dataset in our prior work [15] had a **max**, **min**, and **mean** length of **2161**, **5** and **378.88**, respectively




(c) DocTAET feature length in the dataset from the Grobid workflow in our present paper has a **max**, **min**, and **mean** length of **2686**, **101** and **513.37**, respectively




(d) DocTAET feature length in the dataset from the LaTeX workflow in our present paper has a **max**, **min**, and **mean** length of **7374**, **100** and **685.25**, respectively

**Fig. 1** DocTAET feature length of papers in the original science result extractor dataset [33] Fig. 1a, the dataset used in our prior ORKG-TDM experiments [15] Fig. 1a, the dataset from the Grobid workflow in our present work Fig. 1c, and the dataset from the LaTeX workflow in our present work Fig. 1d

stream tasks. These models have thus crucially popularized the transfer learning paradigm in NLP. We investigate three transformer-based model variants for *leaderboard* extraction in a natural language inference configuration.

Natural language inference (NLI), generally, is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise" [39]. For *leaderboard* extraction, the slightly adapted NLI task is to determine that the (*task*, *dataset*, *metric*) "hypothesis" is true (entailed) or false (not entailed) for a paper given the "premise" as the DocTAET context feature representation of the paper.

Currently, there exist several transformer-based models. In our experiments, we investigated four core models: three variants of BERT, i.e., the vanilla BERT [38], scientific BERT (SciBERT) [40], and BigBird [16]. We also tried a different type of transformer model than BERT called XLNet [41], which employs Transformer XL as the backbone model. Next, we briefly describe the four variants we use.

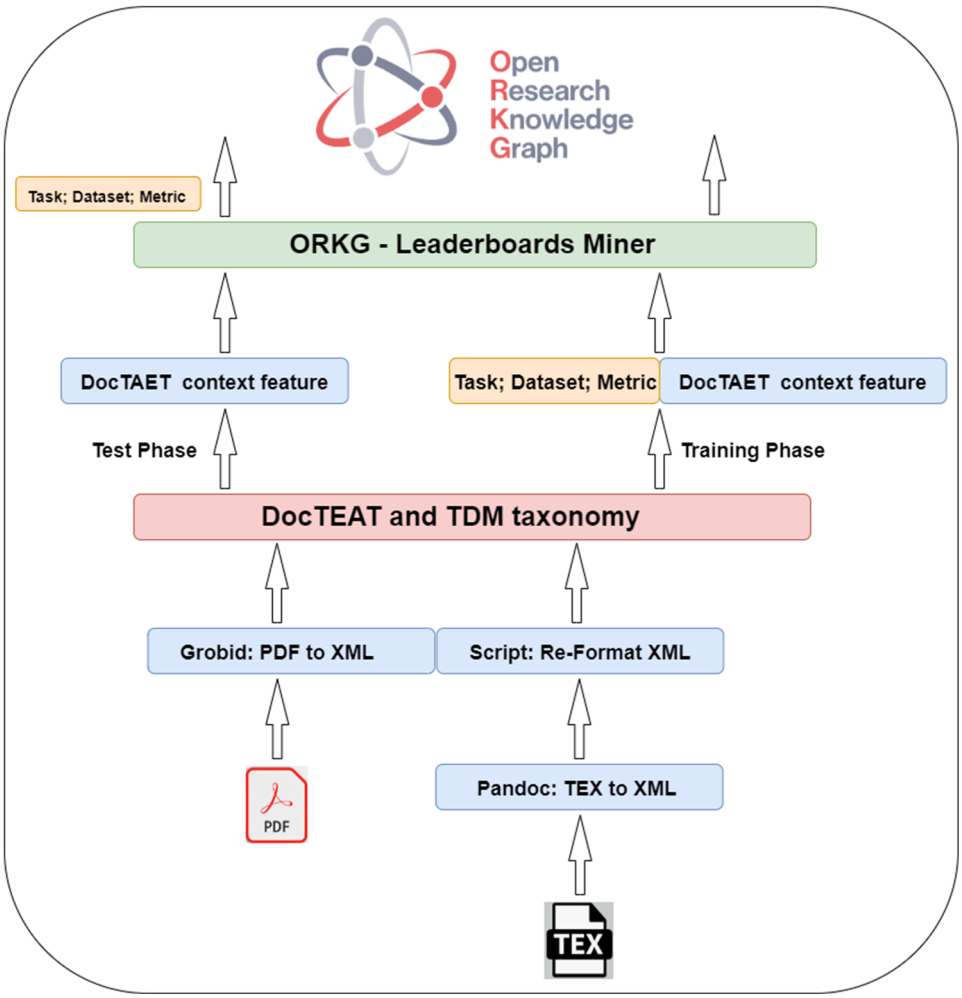



**Fig. 2** The ORKG-Leaderboards end-to-end system workflow in the context of the open research knowledge graph (ORKG) digital library https://orkg.org/

**BERT models**

BERT (i.e., Bidirectional Encoder Representations from Transformers) is a bidirectional autoencoder (AE) language model. As a pre-trained language representation built on the deep neural technology of transformers, it provides NLP practitioners with high-quality language features from text data simply out of the box and thus improves performance on many NLP tasks. These models return contextualized word embeddings that can be directly employed as features for downstream tasks [42].

The first BERT model we employ is $\text{BERT}_{base}$ (12 layers, 12 attention heads, and 110 million parameters), which was pre-trained on billions of words from the BooksCorpus (800 M words) and the English Wikipedia (2500 M words).

The second BERT model we employ is the pre-trained scientific BERT called SciBERT [40]. SciBERT was pre-trained on a large corpus of scientific text. In particular, the pre-training corpus is a random sample of 1.14M papers from Semantic Scholar[19] consisting of full texts of 18% of the papers from the computer science domain and 82% from the broad biomedical field. We used their uncased variants for both $\text{BERT}_{base}$ and SciBERT.

**XLNet**

XLNet is an autoregressive (AR) language model [41] that enables learning bidirectional contexts using permutation language modeling. This is unlike BERT's masked language modeling strategy. Thus in PLM, all tokens are predicted but in random order, whereas in MLM, only the masked (15%) tokens are predicted. This is also in contrast to the traditional language models, where all tokens are predicted in sequential order instead of randomly. Random order prediction helps the model to learn bidirectional relationships and, therefore, better handle dependencies and relations between words. In addition, it uses Transformer XL [43] as the base architecture, which models long contexts, unlike the BERT models with contexts limited to 512 tokens. Since only cased models are available for XLNet, we used the cased $\text{XLNet}_{base}$ (12 layers, 12 attention heads, and 110 million parameters).

[19] https://semanticscholar.org.

**BigBird**
BigBird is a sparse-attention-based transformer that extends Transformer based models, such as BERT, to much longer sequences. Moreover, BigBird comes along with a theoretical understanding of the capabilities of a complete transformer that the sparse model can handle [16]. BigBird takes inspiration from graph sparsification methods by relaxing the need for the attention to fully attend to all the input tokens. Formally the model first builds a set of $g$ global tokens attending on all parts of the sequence, then all tokens attend to a set of $w$ local neighboring tokens, and finally, all tokens attend to a set of $r$ random tokens. The empirical configuration explained in the last paragraph leads to a high-performing attention mechanism scaling to much longer sequence lengths ($8\times$) [16].

## 6 ORKG-Leaderboards system experiments

### 6.1 Experimental setup

**Parameter tuning**
We use the Hugging Transformer libraries[20] with their BERT variants and XLNet implementations. In addition to the standard fine-tuned setup for NLI, the transformer models were trained with a learning rate of $1e^{-5}$ for 14 epochs; and used the $AdamW$ optimizer with a weight decay of 0 for *bias*, *gamma*, *beta* and 0.01 for the others. Our models' hyperparameters details can be found in our code repository online at.[21]

In addition, we introduced a task-specific parameter that was crucial in obtaining optimal task performance from the models. It was the number of $false$ triples per paper. This parameter controls the discriminatory ability of the model. The original science result extractor system [33] considered $|n| - |t|$ *false* instances for each paper, where $|n|$ was the distinct set of triples overall and $|t|$ was the number of $true$ *leaderboard* triples per paper. This approach would not generalize to our larger corpus with over 1724 distinct triples. In other words, considering that each paper had on average 4 *true* triples, it would have a larger set of *false* triples which would strongly bias the classifier learning toward only *false* inferences. Thus, we tuned this parameter in a range of values in the set {10, 50, 100} which at each experiment run was fixed for all papers.

Finally, we imposed an artificial trimming of the DocTAET feature to account for BERT and SciBERT's maximum token length of 512. For this, the token lengths of the experimental setup and table info were initially truncated to roughly 150 tokens, after which the DocTAET feature is trimmed at the right to 512 tokens. Whereas, XLNet and BigBird are specifically designed to handle longer contexts of undefined lengths. Nevertheless, to optimize for training speed, we incorporated a context length of 2000 tokens.

**Evaluation**
Similar to our prior work [15], all experiments are performed via twofold cross-validation. Within the twofold experimental settings, we report macro- and micro-averaged precision, recall, and F1 scores for our *Leaderboard* extraction task on the test dataset. The macro scores capture the averaged class-level task evaluations, whereas the micro scores represent fine-grained instance-level task evaluations.

Further, the macro and micro evaluation metrics for the overall task have two evaluation settings: (1) considers papers with *task–dataset–metric* and papers with "unknown" in the metric computations; and (2) only papers with *task–dataset–metric* are considered while the papers with "unknown" are excluded. In general, we focus on the model performances in the first evaluation setting as it directly emulates the real-world application setting that includes papers that do not report empirical research and therefore for which the *Leaderboard* model does not apply. In the second setting, however, the reader still can gain insights into the model performances when given only papers with *Leaderboards*.

### 6.2 Experimental results

In this section, we discuss new experimental findings shown in Tables 2, 3, 4, and 5 with respect to four research questions elicited as **RQ1**, **RQ2**, **RQ3**, and **RQ4**, respectively.

**RQ1: Which is the best model in the real-world setting when considering a dataset of both kinds of papers: those with *Leaderboard*s and those without *Leaderboard*s therefore labeled as "Unknown"?**

For these results, we refer the reader to the first four results' rows in both Tables 2 and 3, respectively. Note, Table 2 reports results from the Grobid processing workflow and Table 3 reports results from the LaTeX processing workflow. In both cases, it can be observed that ORKG-Leaderboards$_{XLNet}$ is the best transformer model for the *Leaderboard* inference task in terms of micro-F1. In the case of the Grobid processing workflow, the best micro-F1 from this model is 94.8%. Whereas in the case of LaTeX processing workflow, the best micro-F1 from ORKG-Leaderboards$_{XLNet}$ is 93.0%. Note in selecting the best model we prefer the micro evaluations since they reflect the fine-grained discriminative ability of the models at the instance level. The macro scores are seen simply as supplementary measures in this regard to observing the performance of the models at the class level.

[20] https://github.com/huggingface/transformers.

[21] https://github.com/Kabongosalomon/task-dataset-metric-nli-extraction/blob/main/train_tdm.py.

**Table 2** $BERT_{512}$, $SciBERT_{512}$, $XLNet_{2000}$, and $BigBird_{2000}$ results, trained on the subset of the dataset released by [15] from the Grobid workflow

| | Ma-P[1] | Ma-R | Ma-F1 | Mi-P[2] | Mi-R | Mi-F1 |
|---|---|---|---|---|---|---|
| *Average evaluation across twofold* | | | | | | |
| ORKG-Leaderboards$_{BERT}$ | **93.2** | 95.7 | 93.5 | 95.4 | 93.9 | 94.7 |
| ORKG-Leaderboards$_{SciBERT}$ | 92.6 | 94.3 | 92.2 | 95.4 | 91.1 | 93.2 |
| ORKG-Leaderboards$_{XLNet}$ | 93.1 | **96.4** | **93.7** | 95.1 | **94.6** | **94.8** |
| ORKG-Leaderboards$_{BigBird}$ | **93.2** | 94.9 | 93.0 | **95.7** | 92.4 | 94.0 |
| *Average evaluation across twofold (without "unknown" annotation)* | | | | | | |
| ORKG-Leaderboards$_{BERT}$ | 91.3 | 94.4 | 91.8 | 94.8 | **93.9** | **94.3** |
| ORKG-Leaderboards$_{SciBERT}$ | 90.5 | 92.5 | 90.3 | 94.8 | 90.6 | 92.7 |
| ORKG-Leaderboards$_{XLNet}$ | 91.3 | **95.0** | **92.0** | 94.3 | 93.5 | 93.9 |
| ORKG-Leaderboards$_{BigBird}$ | **91.5** | 93.3 | 91.3 | **95.2** | 92.2 | 93.6 |

The numbers in bold correspond to the best model's performance
[1]Macro precision
[2]Micro precision

**Table 3** $BERT_{512}$, $SciBERT_{512}$, $XLNet_{2000}$ and $BigBird_{2000}$ results, based on DocTEAT from LaTeX code source

| | Ma-P[1] | Ma-R | Ma-F1 | Mi-P[2] | Mi-R | Mi-F1 |
|---|---|---|---|---|---|---|
| *Average evaluation across twofold* | | | | | | |
| ORKG-Leaderboards$_{BERT}$ | **93.5** | 94.2 | **92.8** | **96.0** | 90.0 | 92.9 |
| ORKG-Leaderboards$_{SciBERT}$ | 91.7 | 93.9 | 91.6 | 94.6 | 88.6 | 91.5 |
| ORKG-Leaderboards$_{XLNet}$ | 91.9 | **94.4** | 92.0 | 94.9 | **91.2** | **93.0** |
| ORKG-Leaderboards$_{BigBird}$ | 90.7 | 91.6 | 89.7 | 94.6 | 87.2 | 90.7 |
| *Average evaluation across twofold (without "unknown" annotation)* | | | | | | |
| ORKG-Leaderboards$_{BERT}$ | **91.2** | 92.3 | **90.6** | **95.4** | 88.0 | 91.5 |
| ORKG-Leaderboards$_{SciBERT}$ | 89.4 | 91.7 | 89.2 | 93.7 | 86.0 | 89.7 |
| ORKG-Leaderboards$_{XLNet}$ | 89.5 | **92.4** | 89.8 | 94.2 | **89.4** | **91.7** |
| ORKG-Leaderboards$_{BigBird}$ | 87.5 | 88.7 | 86.6 | 93.6 | 85.3 | 89.3 |

The numbers in bold correspond to the best model's performance
[1]Macro precision
[2]Micro precision

**RQ2: How do the models in two processing workflows, i.e., Grobid producing plain text with some noise and the clean plain text from LaTeX, compare, both in general and specifically for the best ORKG-Leaderboards$_{XLNet}$ model?**

The model trained on the plain text obtained from LaTeX contrary to our intuition shows a lower performance compared to the one trained on the noisy Grobid produced plain text. One possible cause maybe related to the context length as the LaTeX produced dataset has an average length of 685.25 compared to 512.37 for the Grobid produced data, as shown in Fig. 1c, d. In this case, we hypothesize that for the LaTeX processing workflow to be implemented with the most effective model, experiments with a much larger dataset are warranted. There may be one of two outcomes: (1) the model from the LaTeX workflow still performs worse than the model from the Grobid workflow in which case we can conclude that longer contexts regardless of whether they are from a clean source or noisy source are difficult to generalize from, or (2) the model from the LaTeX workflow indeed begins to outperform the model from the Grobid workflow in which case we can safely conclude that for the transformer models to generalize on longer contexts a much larger training dataset is needed. We relegate these further detailed experiments to future work.

**RQ3: Which insights can be gleaned from the BERT and SciBERT models operating on shorter context lengths of 512 tokens versus the more advanced models, viz. XLNet and BigBird, operating on longer context lengths of 2000 tokens?**

We observed that BERT and SciBERT models show lower performance compared to the XLNet transformer model operating on 2000 tokens. This we hypothesized as expected behavior since the longer contextual information can capture richer signals for the model to learn from, which is highly likely to be lost when imposing the 512 tokens limit. Contrary to this intuition, however, the BigBird model with the longer context is not able to outperform BERT and SciBERT.

**Table 4** Performance of our best model, i.e., ORKG-Leaderboards$_{XLNet}$, for *Task*, *Dataset*, and *Metric* concept extraction of the *leaderboard* for the grobid workflow

| Entity | Macro | | | Micro | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| TDM | 93.1 | 96.4 | 93.7 | 95.1 | 94.6 | 94.8 |
| Task | 94.3 | 97.2 | 95.0 | 96.8 | 95.9 | 96.4 |
| Dataset | 93.8 | 96.7 | 94.4 | 96.2 | 95.4 | 95.8 |
| Metric | 93.7 | 96.9 | 94.4 | 96.0 | 95.3 | 95.6 |

**Table 5** Performance of our best model, i.e., ORKG-Leaderboards$_{XLNet}$, for *Task*, *Dataset*, and *Metric* concept extraction of the *leaderboard* for the latex workflow

| Entity | Macro | | | Micro | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| TDM | 91.9 | 94.4 | 92.0 | 94.9 | 91.2 | 93.0 |
| Task | 94.3 | 97.2 | 95.0 | 96.8 | 95.9 | 96.4 |
| Dataset | 93.8 | 96.7 | 94.4 | 96.2 | 95.4 | 95.8 |
| Metric | 93.7 | 96.9 | 94.4 | 96.0 | 95.3 | 95.6 |

We suspect the specific attention mechanism in the BigBird model [16] needs further examination over a much larger dataset to conclude that it is ineffective for *task–dataset–metric* extraction task compared to other transformer-based models.

#### RQ4: Which of the three *Leaderboard task–dataset–metric* concepts are easy or challenging to extract?

As a fine-grained examination of our best model, i.e., ORKG-Leaderboards$_{XLNet}$, we examined its performance for extracting each of three concepts $(task, dataset, metric)$ separately. These results are shown in Tables 3 and 4. From the results, we observe that *Task* is the easiest concept to extract, followed by *Metric*, and then *Dataset*. We ascribe the low performance for extracting the *Dataset* concept due to the variability in its naming seen across papers even when referring to the same real-world entity. For example, the real-world dataset entity "CIFAR-10" is labeled as "CIFAR-10, 4000 Labels" in some papers and "CIFAR-10, 250 Labels" in others. This phenomenon is less prevalent for *Task* and the *Metric* concepts. For example, the *Task* "Question Answering" is rarely referenced differently across papers addressing the task. Similarly, for *Metric*, "accuracy" as an example, has very few variations.

## 7 Integrating ORKG-Leaderboards in the open research knowledge graph

In this era of the publications deluge worldwide [4, 5, 44], researchers are faced with a critical dilemma: *How to stay on track with the past and the current rapid-evolving research progress?* With this work, our main aim is to propose a solution to this problem. And with the ORKG-Leaderboards software, we have concretely made advances toward our aim in the domain of empirical AI research. Furthermore, with the software integrated into the next-generation digitalized publishing platform, viz. https://orkg.org/, the machine-actionable *task–dataset–metric* data represented as a knowledge graph with the help of the Semantic Web's RDF language makes the information skimmable for the scientific community. This is achieved via the dynamic Frontend views of the ORKG benchmarks feature https://orkg.org/benchmarks. This is illustrated via Fig. 3. On the left side of Fig. 3 is shown the traditional PDF-based paper format. Highlighted within the view are the Task, Dataset, and Metric phrases. As evident, the phrases are mentioned in several places in the paper. Thus in this traditional model of publishing via non-machine-actionable PDFs, a researcher interested in this critical information would need to scan the full paper content. They are then faced with the intense cognitive burden of repeating such a task over a large collection of articles. On the right side of Fig. 3 is presented a dynamic ORKG Frontend view of the same information, however over machine-actionable RDF semantically represented information of the Task, Dataset, and Metric elements. To generate such a view, the ORKG-Leaderboard software would simply be applied on a large collection of articles either in LaTeX or PDF format, and the resulting *task–dataset–metric* tuples uploaded in the ORKG. Note, however, ORKG-Leaderboard does not attempt extraction of the *Score* element. We observed from some preliminary experiments that the *Score* element poses a particularly hard extraction target. This is owing to the fact that the underlying contextual data supporting *Score* extraction is especially noisy–clean table data extraction from PDFs are a challenging problem in the research community that would need to be addressed first to develop promising *Score* extractors. Nevertheless, in the context of this missing data in the ORKG Benchmarks pages, its human-in-the-loop curation model is relied on. In such a setting, respective article authors with their *task–dataset–metric* model information being automatically extracted to the KG can simply edit their corresponding model scores in the graph. Thus as concretely shown on the right screen of Fig. 3, empirical results are made skimmable and easy to browse for researchers interested in gaining an overview of empirical research progress via a ranked list of papers proposing models and a performance progress trend chart computed over time.

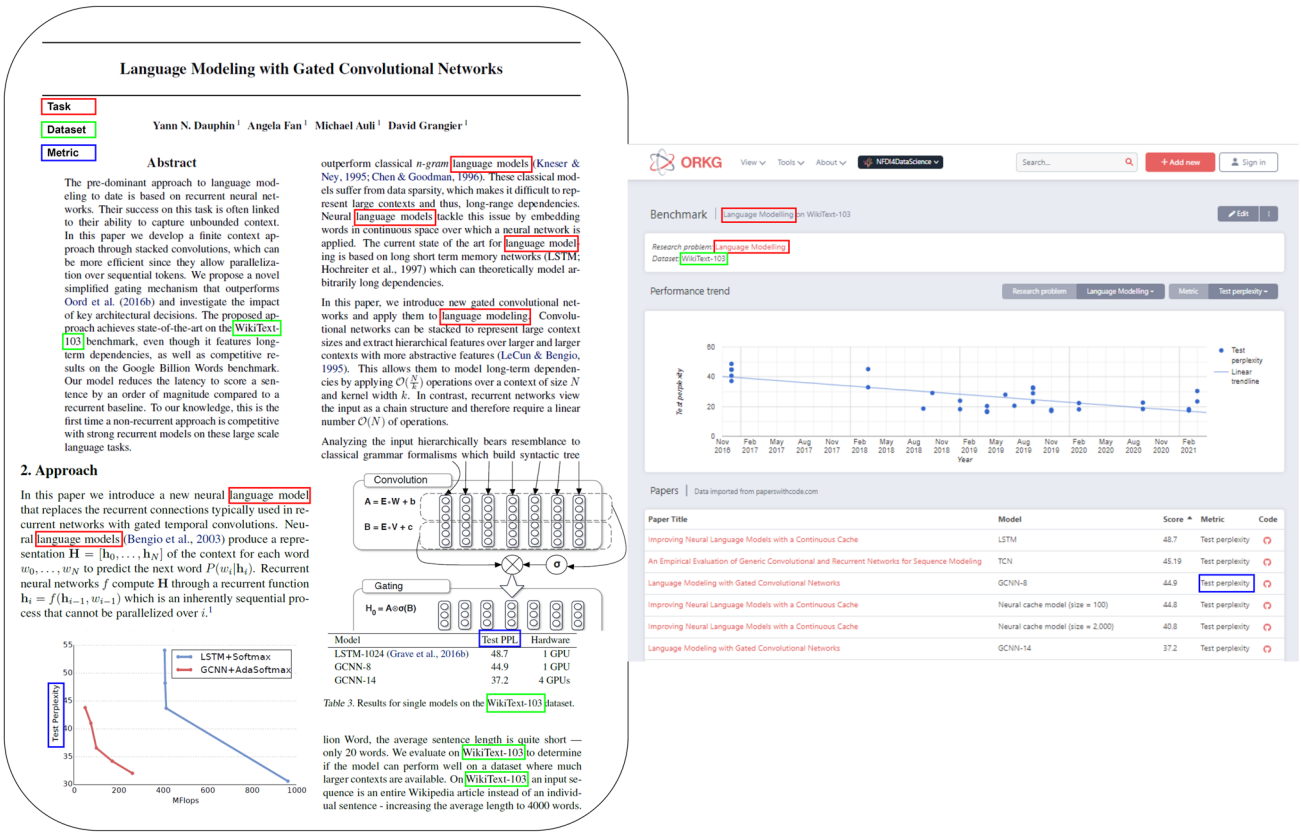



**Fig. 3** A contrastive view of *task–dataset–metric* information in the traditional PDF format of publishing as non-machine-actionable data (on the left) versus as machine-actionable data with *task–dataset–metric* annotations obtained from ORKG-Leaderboards and integrated in the next-generation scholarly knowledge platform as the ORKG benchmarks view (on the right)

Although the experiments of our study targeted empirical AI research, we are confident, that the approach is transferable to similar scholarly knowledge extraction tasks in other domains. For example in Chemistry or Material Sciences, experimentally observed properties of substances or materials under certain conditions could be obtained from various papers.

## 8 Conclusion and future work

In this work, we experimented with the empirical construction of Leaderboards, using four recent transformer-based models (BERT, SciBERT, XLNet, BigBird) that have achieved state-of-the-art performance in several tasks and domains in the literature. Leveraging the two main streams of information acquisition used in scholarly communication, i.e., (Pdf, LaTeX), our work published two models to accurately extract task dataset and metric entities from an empirical AI research publication. Therefore as a next step, we will extend the current triples (task, dataset, metric) model with additional concepts that are suitable candidates for a Leaderboard such as score or code URLs, etc. We also envision the task–dataset–metric extraction approach to be transferable to other domains (such as materials science, engineering simulations, etc.). Our ultimate target is to create a comprehensive structured knowledge graph tracking scientific progress in various scientific domains, which can be leveraged for novel machine-assistance measures in scholarly communication, such as question answering, faceted exploration, and contribution correlation tracing.

**Acknowledgements** This work was co-funded by the Federal Ministry of Education and Research (BMBF) of Germany for the project LeibnizKILabor (grant no. 01DD20003), BMBF project SCINEXT (GA ID: 01lS22070), NFDI4DataScience (grant no. 460234259) and by the European Research Council for the project ScienceGRAPH (Grant agreement ID: 819536).

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## References


1. Parra Escartín, C., Reijers, W., Lynn, T., Moorkens, J., Way, A., Liu, C.-H.: Ethical considerations in NLP shared tasks. In: Proceedings of the First ACL Workshop on Ethics in Natural Language Processing, pp. 66–73. Association for Computational Linguistics, Valencia, Spain (2017). https://doi.org/10.18653/v1/W17-1608
2. Nissim, M., Abzianidze, L., Evang, K., van der Goot, R., Haagsma, H., Plank, B., Wieling, M.: Last words: sharing is caring: the future of shared tasks. Comput. Linguist. **43**(4), 897–904 (2017)
3. Kim, J.-D., Pyysalo, S.: In: Dubitzky, W., Wolkenhauer, O., Cho, K.-H., Yokota, H. (eds.) BioNLP Shared Task, pp. 138–141. Springer, New York (2013). https://doi.org/10.1007/978-1-4419-9863-7_138
4. Jinha, A.E.: Article 50 million: an estimate of the number of scholarly articles in existence. Learn. Publ. **23**(3), 258–263 (2010)
5. Chiarelli, A., Johnson, R., Richens, E., Pinfield, S.: Accelerating scholarly communication: the transformative role of preprints (2019)
6. paperswithcode.com. https://paperswithcode.com/. Accessed 26 Apr 2021
7. NLP-progress. http://nlpprogress.com/. Accessed 26 Apr 2021
8. AI metrics. https://www.eff.org/ai/metrics. Accessed 26 Apr 2021
9. SQuAD Explorer. https://rajpurkar.github.io/SQuAD-explorer/. Accessed 26 Apr 2021
10. Reddit Sota. https://github.com/RedditSota/state-of-the-art-result-for-machine-learning-problems. Accessed 26 Apr 2021
11. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. Sci. Data **3**, 1–9 (2016)
12. Jacobsen, A., de Miranda Azevedo, R., Juty, N., Batista, D., Coles, S., Cornet, R., Courtot, M., Crosas, M., Dumontier, M., Evelo, C.T., et al.: FAIR Principles: Interpretations and Implementation Considerations. MIT Press, Cambridge (2019)
13. Auer, S., Oelen, A., Haris, M., Stocker, M., D'Souza, J., Farfar, K.E., Vogt, L., Prinz, M., Wiens, V., Jaradeh, M.Y.: Improving access to scientific literature with knowledge graphs. Bibliothek Forschung und Praxis **44**(3), 516–529 (2020)
14. Escartín, C.P., Lynn, T., Moorkens, J., Dunne, J.: Towards transparency in NLP shared tasks. arXiv preprint arXiv:2105.05020 (2021)
15. Kabongo, S., D'Souza, J., Auer, S.: Automated mining of leaderboards for empirical ai research. In: International Conference on Asian Digital Libraries, pp. 453–470 . Springer (2021)
16. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L.: Big bird: transformers for longer sequences. Adv. Neural. Inf. Process. Syst. **33**, 17283–17297 (2020)
17. D'Souza, J., Auer, S.: Computer science named entity recognition in the open research knowledge graph. In: From Born-Physical to Born-Virtual: Augmenting Intelligence in Digital Libraries: 24th International Conference on Asian Digital Libraries, ICADL 2022, Hanoi, Vietnam, November 30–December 2, 2022, Proceedings, pp. 35–45 . Springer (2022)
18. Gupta, S., Manning, C.: Analyzing the dynamics of research by extracting key aspects of scientific papers. In: Proceedings of 5th International Joint Conference on Natural Language Processing, pp. 1–9. Asian Federation of Natural Language Processing, Chiang Mai, Thailand (2011). https://aclanthology.org/I11-1001
19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
20. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th InternationalConference on Neural Information Processing Systems, vol. 2. NIPS 13, pp. 2787–2795. Curran Associates Inc., Red Hook, NY, USA (2013)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
22. Sasaki, Y.: The truth of the f-measure. Teach. Tutor. Mater. **1**(5), 1–5 (2007)
23. Voorhees, E.M.: The trec-8 question answering track report. In: Trec, vol. 99, pp. 77–82 (1999)
24. Anteghini, M., D'Souza, J., dos Santos, V.A., Auer, S.: Easy semantification of bioassays. In: International Conference of the Italian Association for Artificial Intelligence, pp. 198–212 . Springer (2022)
25. Kononova, O., Huo, H., He, T., Rong, Z., Botari, T., Sun, W., Tshitoyan, V., Ceder, G.: Text-mined dataset of inorganic materials synthesis recipes. Sci. Data **6**(1), 1–11 (2019)
26. Kulkarni, C., Xu, W., Ritter, A., Machiraju, R.: An annotated corpus for machine reading of instructions in wet lab protocols. In: NAACL: HLT, Volume 2 (Short Papers), New Orleans, Louisiana, pp. 97–106 (2018). https://doi.org/10.18653/v1/N18-2016
27. Mysore, S., Jensen, Z., Kim, E., Huang, K., Chang, H.-S., Strubell, E., Flanigan, J., McCallum, A., Olivetti, E.: The materials science procedural text corpus: annotating materials synthesis procedures with shallow semantic structures. In: Proceedings of the 13th Linguistic Annotation Workshop, pp. 56–64 (2019)
28. Kuniyoshi, F., Makino, K., Ozawa, J., Miwa, M.: Annotating and extracting synthesis process of all-solid-state batteries from scientific literature. In: LREC, pp. 1941–1950 (2020)
29. Handschuh, S., QasemiZadeh, B.: The acl rd-tec: a dataset for benchmarking terminology extraction and classification in computational linguistics. In: COLING 2014: 4th International Workshop on Computational Terminology (2014)
30. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. In: SemEval@ACL (2017)
31. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: EMNLP (2018)
32. D'Souza, J., Hoppe, A., Brack, A., Jaradeh, M.Y., Auer, S., Ewerth, R.: The stem-ECR dataset: Grounding scientific entity references in stem scholarly content to authoritative encyclopedic and lexicographic sources. In: LREC, Marseille, France, pp. 2192–2203 (2020)
33. Hou, Y., Jochim, C., Gleize, M., Bonin, F., Ganguly, D.: 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, pp. 5203–5213. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1513
34. Jain, S., van Zuylen, M., Hajishirzi, H., Beltagy, I.: Scirex: A challenge dataset for document-level information extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7506–7516 (2020)

35. Mondal, I., Hou, Y., Jochim, C.: End-to-end construction of nlp knowledge graph. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1885–1895 (2021)
36. GROBID. GitHub (2008–2022)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
38. Kenton, J.D.M.-W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
39. Natural Language Inference. https://paperswithcode.com/task/natural-language-inference. Accessed 22 Apr 2021
40. Beltagy, I., Lo, K., Cohan, A.: 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), pp. 3615–3620. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1371
41. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
42. Jiang, M., D'Souza, J., Auer, S., Downie, J.S.: Improving scholarly knowledge representation: Evaluating bert-based models for scientific relation classification. In: International Conference on Asian Digital Libraries, pp. 3–19 . Springer (2020)
43. Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2978–2988 (2019)
44. Ware, M., Mabe, M.: The STM report: An overview of scientific and scholarly journal publishing (2015)