Streamlining Knowledge Discovery in Scientific Literature: A Comprehensive End-to-End System for Research Artifact Analysis

Anonymous ACL submission

Abstract

Knowledge Discovery and Research Artifact Analysis (RAA) are crucial for promoting reproducibility and reusability in scientific research. In this work, we introduce a novel end-to-end system to efficiently identify and analyze tangible research artifacts (RAs), specifically datasets and software, within scientific literature. Building on recent advancements, our architecture employs Large Language Models (LLMs) fine-tuned with the Low-Rank Adaptation (LoRA) method to streamline the process of RAA into an instruction-based Question Answering (QA) task. The system comprises five stages: (i) candidate detection using a list of curated keywords and gazetteers, (ii) RA mention identification and validation, (iii) extraction of RA mention metadata, such as names, versions, licenses, and URLs, (iv) classification of RA mentions by usage and provenance, and (v) deduplication of RA mentions to ensure the uniqueness of each identified RA. Through benchmarking on two RA mention datasets, we demonstrated robust performance in RAA and provided a comprehensive qualitative analysis, underscoring the nuances and complexities of ensuring reproducibility and reusability in diverse scientific fields.

1 Introduction

The continuous advancement of scientific knowledge necessitates the development of novel methodologies for identifying and analyzing research artifacts (RAs) within scientific literature. Such tools should streamline the process of Research Artifact Analysis (RAA), strengthening both the reproducibility of experiments and the reusability of data and software. In addition, these tools should maintain a balanced parameter-to-performance ratio, making them more accessible to a broader part of the scientific community, especially to research groups with limited resources.

RAs often fall into two broad categories: tangible and intangible. Tangible RAs include items with a physical or digital presence, such as software and datasets. In contrast, intangible RAs, like methodologies and procedures, represent theoretical frameworks and structured approaches to research. Despite the apparent simplicity of this distinction, it is important to note that the boundaries between these categories are not always clear-cut, highlighting the complexity of RAA. Significant research efforts have thus been devoted to developing robust architectures and models for RAA, addressing the unique characteristics and requirements of each category (Wang et al., 2022; Krüger and Schindler, 2020).

In this work, we focus on identifying tangible named and unnamed RAs, specifically software and datasets. In order to accomplish this objective, we developed an innovative end-to-end system (Figure 1) that utilizes Large Language Models (LLMs) to efficiently identify RAs and extract their associated metadata. More specifically, we expand upon recent findings (Stavropoulos et al., 2023), about the efficacy of fine-tuned LLMs, using the Low-Rank Adaptation (LoRA) (Hu et al., 2021) method, in extracting RA mentions and their metadata. Our objective is to further harness and extend the potential of these models, aiming for comprehensive RAA within the context of scientific publications.

The proposed end-to-end system comprises five steps:

- **Candidate detection:** Through meticulous scanning of the scientific text, potential trigger words for RA mentions are identified.
- **RA Identification & Validation:** Each candidate RA mention is rigorously assessed as a valid RA mention or an incidental reference.
- **Metadata extraction:** For each valid RA, relevant information such as their name, version, license, and URL are extracted.
- **RA classification:** For each valid RA, their usage and provenance by the authors within

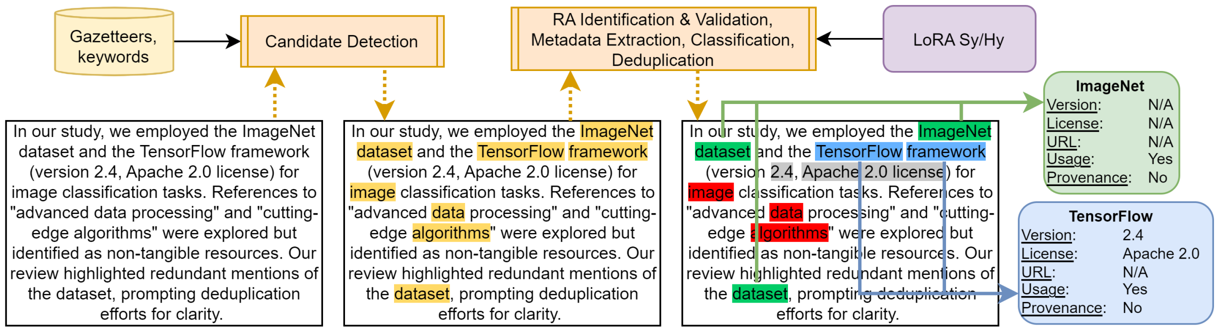

Figure 1: Overview of our end-to-end system for extracting research artifacts (RAs) from scientific literature. Yellow highlights candidate RA mentions, red invalid RA mentions, green dataset mentions, blue software mentions, and gray RA metadata.

the scientific text are classified.

- **RA Deduplication:** RA mentions are consolidated into unique RAs, and their metadata and usage/provenance are aggregated and reassessed.

In the subsequent sections, we detail the LoRA-finetuned models that serve as the foundation of our RAA system (Section 2) and provide an extensive overview of the end-to-end system architecture (Section 3). We then explore the deployment and results of our system on two selected RA mention datasets (Pan et al., 2023; Schindler et al., 2021) (Section 4) and analyze its performance. Finally, we conduct a detailed review of related technologies from existing literature (Section 5).

Our key contributions are:

1. We developed a novel end-to-end RAA system, identifying both named and unnamed RAs, including datasets and software[1]. Our system leverages the LLMs previously fine-tuned and fully documented in the work of (Stavropoulos et al., 2023) on the task of RA mention extraction.
2. We developed a comprehensive deduplication pipeline that consolidates RA mentions into unique RAs, enabling document-level RAA.
3. We evaluated our system against two prominent RA mention datasets, DMDD and SoMeSci. Despite not being trained on these datasets, our system performed comparably to top-performing models in both dataset and software mention identification, metadata extraction and usage/provenance classification.

2 LoRA-finetuned LLMs for RAA

Our proposed architecture employs the LoRA-Sy and LoRA-Hy models, which are Flan-T5 Base (Chung et al., 2022) models fine-tuned, using the LoRA method (Hu et al., 2021), on the Synthetic and Hybrid RA mention datasets, as detailed in (Stavropoulos et al., 2023). These models have been fine-tuned to tackle RAA as an instruction-based Question Answering (QA) task and are integral to the RA identification and validation, metadata extraction, and RA classification phases of our system.

Snippet	We used the SciPy <m>library</m> (version 1.7.0) for scientific computations. SciPy is released under the BSD license and can be accessed at https://www.scipy.org/.
Type	Software
Valid	Yes
Name	SciPy
Version	1.7.0
License	BSD
URL	https://www.scipy.org/
Provenance	No
Usage	Yes

Figure 2: An example of a named RA mention containing all metadata.

Snippet	To train our introduced learnable parameters, we compose a dataset of <m>44K fine-grained masks</m> from several sources.
Type	Dataset
Valid	Yes
Name	N/A
Version	N/A
License	N/A
URL	N/A
Provenance	Yes
Usage	Yes

Figure 3: An example of an unnamed RA mention.

Each snippet in the task contains RA mentions marked with '<m>' and '</m>' tags. The model is prompted to respond to a series of questions to establish the validity of each RA mention and extract metadata[2], strictly confined to the information presented within the snippet (Figures 2-3).

An RA mention is considered valid if it represents a tangible research input or output. Gen-

[1] The code repository will be provided upon acceptance.

[2] Metadata includes: Type, Valid, Name, Version, License, URL, Usage and Provenance.

eral references to RAs are considered invalid. The model eliminates incorrectly identified RA mentions during the candidate detection phase using the validity questions. Following the convention used by the Synthetic and Hybrid datasets in (Stavropoulos et al., 2023), we define a dataset as a systematically organized collection of data, and software as concrete applications, programs, algorithmic frameworks, and implemented model architectures.

Furthermore, the LoRA fine-tuned LLMs are trained to handle situations where the RA mention within the '<m>' and '</m>' tags refers to multiple RAs (e.g., 'datasets'). The models generate the respective validity, metadata, and usage/provenance for each RA delineated using the '|' symbol. This functionality ensures full coverage when multiple RAs are closely referenced within the same context.

3 System Architecture

In this section, we outline our end-to-end system architecture for the extraction of RAs in scientific text. Our approach consists of five phases: (i) candidate detection, (ii) RA identification and validation, (iii) metadata extraction, (iv) RA classification, and (v) RA deduplication. The system pipeline is illustrated in detail in Appendix A.

Our system processes the full text of publications in a structured format (sections, paragraphs, sentences) using GROBID[3]. It can also handle any unstructured text, treating it as a single section and paragraph. Before deduplication, the output is a list of identified RA mentions with metadata, usage, and provenance information. After deduplication, it is a list of RA mention clusters representing unique RAs, with usage and provenance reevaluated based on all mentions within each cluster.

3.1 Candidate Detection

In the candidate detection phase, our system identifies keywords and key phrases that act as triggers for datasets and software in scientific texts. Initially, we manually crafted a seed list of these triggers, meticulously selected from scientific literature. This list was then expanded using Word2Vec embeddings (Mikolov et al., 2013) to identify near-synonyms and underwent thorough manual curation to ensure relevance and precision.

We also incorporated gazetteers from the PapersWithCode (PwC) dataset[4], aiding in the identification of 'candidate RA mentions' within the text. Gazetteers within the PwC dataset can share names, which might lead to their identification as both datasets and software, potentially triggering multiple RA candidate mentions.

Our system uses regular expressions to scan scientific texts for matches of keywords, key phrases, and gazetteers. When a match is found, the system records the exact location of the RA candidate mention within the text, including the section, paragraph, sentence, and offset, preserving the mention in its proper context.

Additionally, the candidate detection stage includes a mechanism that allows the incorporation of gazetteers from external sources beyond the PwC dataset. This mechanism includes additional RA names and triggers collected from the Synthetic and Hybrid datasets. During inference, the system uses this mechanism by issuing a 'special' question (Stavropoulos et al., 2023) for each snippet, generating a list of named RAs to incorporate as additional gazetteers. This approach enhances the model's ability to identify named RAs in new, unseen scientific texts, especially when triggers identified by key phrases or the PwC gazetteers are absent (Appendix D, Figure 8).

To enhance detection efficiency, we integrated a Paragraph Relevance Checker into the candidate detection phase. This submodule uses the LoRA fine-tuned LLM with the 'special' question described above to identify and list all RA mentions in a paragraph. If RA mentions are detected, the paragraph is marked for further examination. This method allows the system to check entire paragraphs with a single question, reducing the need to process multiple candidate RA mentions and improving overall performance by filtering out generic references and invalid RA mentions.

3.2 RA Identification & Validation

After candidate detection, the system proceeds to the RA identification and validation phase. This process uses the LoRA fine-tuned LLM model to determine whether the candidate RA mentions are valid RAs or merely incidental references or descriptive terms within the text. Validation is performed using a 'validity question', as detailed in (Stavropoulos et al., 2023).

[3]Tool that converts publication PDFs to TEI XML format. It can be found at github.com/kermitt2/grobid.

[4]github.com/paperswithcode/paperswithcode-data

Our system employs a classification protocol (Appendix B) that evaluates a set of possible answers to a given question and computes the likelihood of each being generated by the LLM. This method assesses the validity of a candidate RA mention by assigning a score to two definitive responses: 'Yes' or 'No'.

The probability value of the 'Yes' response represents the validity score of the RA mention. This score is compared to a predefined threshold, defaulting to 0.5, allowing control over precision and recall. Only candidate RA mentions surpassing this threshold are considered valid and proceed to the next phase. This ensures that subsequent phases handle only relevant and high-quality RA mentions.

3.3 Metadata Extraction

After validating the RA mentions, the system proceeds to the metadata extraction phase. Here, the LoRA fine-tuned LLM identifies and extracts essential metadata associated with each validated RA mention, such as name, version, license, and URL. These metadata provide a comprehensive understanding of each RA mention and are used in the deduplication phase.

Metadata extraction employs 'metadata questions' directed at the LoRA fine-tuned LLM. As described in (Stavropoulos et al., 2023), these models are trained to use only the text from the snippet for their responses, minimizing hallucinations. Our system discards any metadata not found within the snippet to maintain accuracy.

3.4 RA Classification

In the RA classification phase, the system categorizes identified RA mentions based on their usage and provenance as determined by the authors. This phase distinguishes whether the RA was actively used in the research, created by the authors, or merely cited. It is crucial that this classification is based solely on the snippet of the RA mention, without relying on external information.

To achieve this, the system uses 'classification questions' (Stavropoulos et al., 2023) designed to determine the usage and provenance of the RA mention. Using our classification protocol, the system evaluates the model's confidence regarding the RA mention's usage and provenance, with responses limited to 'Yes' or 'No'. Similar to the identification and validation phases, score thresholds for usage and provenance are set, with default values at 0.5, allowing control over precision and recall.

3.5 RA Deduplication

In the final phase, RA deduplication, the system ensures the distinctiveness of each identified RA by aggregating mentions that refer to the same RA through their metadata, including names and trigger phrases. This phase further refines the evaluation of the usage and provenance of each RA by considering the specific context of each RA mention within the scientific text. The result is a unique list of metadata-enriched RAs.

The process begins by consolidating RA mentions based on their names. The system then considers the trigger words for each cluster, generating a list of alternative names for each name cluster (e.g., [Yandex, Yandex dataset]). Name clusters starting with the same word substring are grouped together, ensuring distinct clusters for similar names (e.g., [Yandex, Yandex data] and [Yandex testing, Yandex testing dataset] are combined, while [ImageNet, ImageNet dataset] remains distinct from [ImageNet1K, ImageNet1K data]). Clusters with the same word tokens in different orders are also merged (e.g., [Human Pose MPII] and [MPII Human Pose]). This process prevents redundant clusters due to similar naming patterns.

To address the complexity of identifying similar RAs, the system uses the SciCo Longformer model (Cattan et al., 2021), specialized in hierarchical cross-document coreference resolution (H-CDCR). This model handles the diversity in scientific language, accurately clustering related RA mentions.

The system leverages the SciCo model's similarity scores to cluster unnamed mentions with named clusters within the same paragraph, as this locality often indicates a strong relationship between mentions. Unnamed mentions not similar enough to any named clusters are independently clustered within the same paragraph. Subsequently, unnamed clusters are matched to named clusters across paragraphs, ensuring all relationships are identified.

Finally, the system merges the clusters that refer to RA mentions with identical citation marks (Appendix D, Figure 9). This step is crucial in a scientific context, as it often indicates references to the same RA within the same publication.

Post-deduplication, the system aggregates metadata such as licenses, versions, and URLs to create a comprehensive overview for each RA. The evaluation of usage and provenance is refined based

on the RA's cluster of mentions. If any mention within the cluster indicates usage, the entire RA is marked as 'used'. Further analysis of the location of RA mentions within the scientific text is conducted. If the initial mention of an RA outside introductory sections, such as 'Background' or 'Related Work', suggests authorial provenance, it is presumed the RA was created by the authors. This approach filters out potential false positives, accurately identifying RAs created by the authors.

It is important to note that the clustering techniques leveraging the SciCo model, as well as the metadata and usage/provenance reevaluations, are not included in the evaluation of our system detailed in section 4. These components are primarily utilized for in-depth analysis of full scientific texts and require further experimentation and evaluation.

4 Experimental results and analysis

4.1 Evaluation Datasets

To evaluate our system, we used two prominent RA mention datasets: DMDD (Pan et al., 2023) and SoMeSci (Schindler et al., 2021). These datasets frame RAA as a Named Entity Recognition (NER) task, in contrast to our instruction-based Question Answering (QA) approach. Below is an overview of each dataset:

- **DMDD:** DMDD includes full-text articles from various scientific disciplines, sourced from S2ORC (Lo et al., 2020) and PwC. These texts are divided into sections and individual sentences. DMDD's primary goal was to create a large-scale dataset, hence only specific named dataset mentions were programmatically annotated using PwC, omitting mentions beyond PwC's scope. However, the evaluation subset, DMDD-E[5], incorporates exhaustive human curation.

- **SoMeSci:** SoMeSci gathers scientific publications from the PubMed Central (PMC) Open Access (OA) subset. These texts are categorized into four subsets, each annotated for software mentions and associated metadata. Mention labels cover both software and mention types. Software is categorized into 'Application', 'Plugin', 'Operating System', and 'Programming Environment', with added tags for 'Abbreviation' and 'AlternativeName'. Mention types include 'Mention', 'Usage', 'Creation', and 'Deposition', aligning with our 'Usage' and 'Provenance' definitions. Specifically, our system interprets 'Mention' as neither usage nor provenance, 'Usage' as usage but not provenance, and both 'Creation' and 'Deposition' as signifying both usage and provenance. Additional metadata, such as URLs, licenses, extensions, versions, developers, and citations, are linked to the mentioned software.

Due to limited computational resources, we conducted evaluations on subsets of the DMDD-E and SoMeSci test sets, containing sentences with at least one RA mention. These subsets are referred to as DMDD-E+ and SoMeSci_test+, respectively.

We selected these datasets because they align with our definitions of datasets and software. The clarity and rigor of the research conducted by the respective scientific teams allowed for a fair and comprehensive comparison with our results. While many studies focus on RAA, particularly software and dataset mentions, direct comparison is challenging. As noted by (Heddes et al., 2021), despite the widespread use of NER, each study employs different approaches, datasets, and unique modifications, adding complexity to the task. We provide an in-depth exploration of RA mention datasets and related models in Section 5.

4.2 Evaluation method

The DMDD and SoMeSci datasets comprise sentence-level gold annotations formatted as an NER task. Our system, in contrast, identifies candidate mentions through triggers and uses an instruction-based QA pipeline to fill templates for each RA. To ensure a fair comparison, we adopted the following methodology:

- We extracted the unique gold RA mentions per sentence from the BIO schema for both datasets.[6]

- We ran our system architecture, including the first step of the deduplication pipeline, to find all RA mentions clusters in each sentence.[7]

[5]Access the DMDD and its DMDD-E subset at kaggle.com/datasets/panhuitong/dmdd-corpus.

[6]Our system predicts both named and unnamed RA mentions using triggers. Converting to BIO notation using pattern matching might introduce biases into our system's results.

[7]In the current setup, annotations are at the sentence level, negating the need for the full deduplication pipeline. Without paragraph context for the SciCo model, aggregating RA metadata and reassessing their usage and provenance is unnecessary.

- We excluded all unnamed RA mentions from our system results, as both datasets focus on named RAs. While SoMeSci has some unnamed software mentions (software coreference mentions), their count was too small for a valid comparison. Even though our system identifies unnamed software mentions, including them would be misleading since they would be considered incorrect for the dataset, potentially biasing the results.

- In cases where the LoRA-Hy/Sy models produce multiple RA mentions from a single trigger (e.g., 'datasets' referring to multiple datasets), we used the '|' character to separate them into individual RA mentions.

- We incorporated the list of alternative names and abbreviations provided by authors of the DMDD and SoMeSci datasets to reduce both false positives and false negatives by recognizing synonyms. For instance, when identifying a cluster as [SNLI, SNLI data], and knowing 'SNLI' stands for 'Stanford Natural Language Inference', we expanded the cluster to include [SNLI, SNLI data, Stanford Natural Language Inference, Stanford Natural Language Inference data].

Our experiments encompassed two evaluation strategies: Exact Match and Partial Match. Exact Match ensures that at least one prediction within an RA mention cluster aligns with all word tokens of a gold RA mention, regardless of their order. This is a stricter measure, serving as an upper boundary when compared with BIO-tag-based results. On the other hand, Partial Match determines whether any prediction within an RA mention cluster is a substring of, or contains all word tokens from, a gold RA mention, and vice versa. This makes the Partial Match closer to BIO-tag-based results, serving as a potential lower boundary. If a predicted RA mention cluster matches multiple unique gold RA mentions, each matched gold RA mention is considered a true positive. To avoid potential bias and ensure that the count of gold targets remains unchanged, the unique gold RA mentions are not subjected to the deduplication pipeline's grouping steps.

In terms of our metrics, we computed both Macro and Micro PRF scores. The Macro PRF scores were computed by calculating the Precision, Recall, and F1-score (PRF) for each instance and then averaging them. In contrast, the Micro PRF scores were calculated by aggregating the true positives, false positives, and false negatives across all instances to produce overall PRF scores. While the Micro scores provide insights into the overall system performance, they can become overly sensitive when the system accumulates multiple false positives in a particularly noisy or problematic instance, leading to an exaggerated decline in overall performance. Conversely, Macro scores are less impacted by individual problematic instances, as any incorrect prediction within an instance leads to a score of 0.0. Thus, the Macro metric is less susceptible to instance-level errors and more indicative of the system's performance.[8]

Additionally, we incorporated an Accuracy metric to quantify the number of accurately identified gold RA mentions. Given our system's design, which responds only upon correct identification of a gold RA mention, smaller models like the Flan-T5 Base might produce disproportionately high PRF metrics for metadata and usage/provenance. This is often due to their evaluation against a smaller number of gold RA mentions, particularly those with well-defined metadata. To provide a holistic perspective, Table 2 presents two variations of PRF scores for metadata and usage/provenance. The first version measures the metrics based on the metadata of correctly identified RA mentions. In contrast, the second version refines these scores by multiplying them with the Accuracy measure, providing a more nuanced evaluation that incorporates the impact of identification errors on each model's performance.

4.3 Evaluation Results & Qualitative Analysis

Table 2 showcases the performance of the LoRA-Sy and LoRA-Hy models compared to the Flan-T5 Base and XL models on the DMDD-E+ and SoMeSci_test+ datasets. These results are also compared to the top-performing models provided by the creators of the two datasets.

For metadata extraction and RA classification, the scores have two variations. Scores outside parentheses reflect performance based solely on correctly identified RA mentions, while scores inside parentheses are adjusted using the Accuracy score, as explained in Subsection 4.2.

The PRF scores in Table 2 reveal that the LoRA fine-tuned models outperform the Flan-T5 Base

[8]Table 2 presents the results using Macro scores. Results using Micro scores are available in Appendix C.

Model	Scoring Method	Metric	DMDD-E+ mention	SoMeSci_test+ mention	url	version	license	usage	provenance
Flan-T5 Base	**Exact match**	Accuracy	0.214	0.116	-	-	-	-	-
		Precision	0.223	0.157	0.404 (0.047)	0.421 (0.049)	0.825 (0.096)	0.000 (0.000)	0.737 (0.085)
		Recall	0.213	0.139	0.404 (0.047)	0.421 (0.049)	0.825 (0.096)	0.000 (0.000)	0.560 (0.065)
		F1	0.214	0.144	0.404 (0.047)	0.421 (0.049)	0.825 (0.096)	0.000 (0.000)	0.636 (0.074)
	Partial match	Accuracy	0.240	0.177	-	-	-	-	-
		Precision	0.243	0.220	0.427 (0.076)	0.549 (0.097)	0.817 (0.145)	1.000 (0.177)	0.519 (0.092)
		Recall	0.236	0.196	0.427 (0.076)	0.549 (0.097)	0.817 (0.145)	0.060 (0.011)	0.560 (0.099)
		F1	0.235	0.203	0.427 (0.076)	0.549 (0.097)	0.817 (0.145)	0.114 (0.020)	0.538 (0.095)
Flan-T5 XL	**Exact match**	Accuracy	0.607	0.269	-	-	-	-	-
		Precision	0.678	0.367	0.873 (0.235)	0.896 (0.241)	0.910 (0.245)	0.958 (0.258)	0.919 (0.247)
		Recall	0.670	0.324	0.873 (0.235)	0.896 (0.241)	0.910 (0.245)	0.919 (0.247)	0.654 (0.176)
		F1	0.662	0.336	0.873 (0.235)	0.896 (0.241)	0.910 (0.245)	**0.938** (0.252)	**0.764** (0.206)
	Partial match	Accuracy	0.683	0.400	-	-	-	-	-
		Precision	0.724	0.496	0.890 (0.356)	0.868 (0.347)	0.935 (0.374)	0.972 (0.389)	0.900 (0.360)
		Recall	0.731	0.446	0.892 (0.357)	0.874 (0.350)	0.935 (0.374)	0.926 (0.370)	0.643 (0.257)
		F1	0.716	0.460	0.891 (0.356)	0.870 (0.348)	0.935 (0.374)	**0.948** (0.379)	**0.750** (0.300)
LoRA-Sy	**Exact match**	Accuracy	0.791	0.592	-	-	-	-	-
		Precision	0.761	0.658	0.952 (0.564)	0.923 (0.546)	0.943 (0.558)	0.902 (0.534)	0.898 (0.532)
		Recall	0.847	0.651	0.960 (0.568)	0.927 (0.549)	0.943 (0.558)	0.812 (0.481)	0.557 (0.330)
		F1	0.781	0.638	0.954 (0.565)	0.925 (0.548)	**0.943** (0.558)	0.854 (0.506)	0.688 (0.407)
	Partial match	Accuracy	0.836	0.777	-	-	-	-	-
		Precision	0.791	0.821	0.958 (0.744)	0.890 (0.692)	0.964 (0.749)	0.921 (0.716)	0.887 (0.689)
		Recall	0.887	0.829	0.965 (0.750)	0.893 (0.694)	0.964 (0.749)	0.837 (0.650)	0.553 (0.430)
		F1	0.816	0.803	**0.960** (0.746)	0.890 (0.692)	**0.964** (0.749)	0.877 (0.681)	0.681 (0.529)
LoRA-Hy	**Exact match**	Accuracy	0.812	0.606	-	-	-	-	-
		Precision	0.756	0.660	0.952 (0.577)	0.949 (0.575)	0.932 (0.565)	0.930 (0.564)	0.929 (0.563)
		Recall	0.864	0.677	0.968 (0.587)	0.955 (0.579)	0.932 (0.565)	0.794 (0.481)	0.627 (0.380)
		F1	**0.785**	**0.649**	**0.958** (**0.581**)	**0.951** (**0.576**)	0.932 (**0.565**)	0.857 (**0.519**)	0.748 (**0.453**)
	Partial match	Accuracy	0.859	0.801	-	-	-	-	-
		Precision	0.786	0.816	0.951 (0.762)	0.889 (0.712)	0.957 (0.767)	0.941 (0.754)	0.898 (0.719)
		Recall	0.904	0.847	0.966 (0.774)	0.898 (0.719)	0.958 (0.767)	0.842 (0.674)	0.631 (0.505)
		F1	**0.819**	**0.807**	0.956 (**0.766**)	**0.891** (**0.714**)	0.957 (**0.767**)	0.889 (**0.712**)	0.741 (**0.594**)
Base-cased SciBERT (Pan et al., 2023)	**BIO tags**	Precision	0.639 ± 0.002	-	-	-	-	-	-
		Recall	0.919 ± 0.002	-	-	-	-	-	-
		F1	0.754 ± 0.002	-	-	-	-	-	-
SoMeNLP (Schindler et al., 2021)	**BIO tags**	Precision	-	0.820	0.963	0.937	0.786	0.865	0.787
		Recall	-	0.804	0.981	0.932	0.786	0.877	0.815
		F1	-	0.803	0.972	0.934	0.786	0.871	0.800

Table 1: Experimental results on the DMDD-E+ and SoMeSci_test+ datasets.

and XL models. Regarding the SoMeSci dataset, it is evident that the Base and XL models struggle in mention extraction. An example of this is illustrated in Figure 4, where only the fine-tuned models successfully identified the software mention. While the XL model shows promising results in metadata extraction and RA mention classification, its performance is poor when inspecting the Accuracy-adjusted PRF scores.

Initially, the XL model appears comparable to the fine-tuned models, seeming superior to the Base model. However, a qualitative assessment of the results shows that the XL model correctly identifies fewer and less complex RA mentions compared to the fine-tuned models. For the DMDD dataset, the Flan-T5 XL model demonstrates a better understanding of the RAA task, yet it still does not perform as well as the fine-tuned models (Appendix D, Figures 10-11).

Sentence	To address these limitations, we present 4Cin, a <m>method</m> to generate 3D models and derive virtual Hi-C (vHi-C) heat maps of genomic loci based on 4C-seq or any kind of 4C-seq-like data, such as those derived from NG Capture-C.
Gold RA mentions	4Cin
Predicted RA clusters	[**'4Cin'**, '4Cin method']

Figure 4: An example of a successful software cluster prediction using the LoRA-Hy model. No software was identified using the Flan-T5 Base and XL models.

The evaluation reveals that the LoRA-Hy model excels in RAA, especially in metadata extraction and classification. The Partial Match metric is particularly effective in capturing model performance, allowing flexibility by including adjacent words in RA mentions. This helps avoid penalizing correct predictions that are slightly broader in scope, thus reducing false negatives (Appendix D, Figure 12). The performance gap between LoRA-Sy and LoRA-Hy in Table 2 highlights the potential for improvement with more comprehensive and diverse RA datasets.

By employing the LoRA-Sy and LoRA-Hy models for the evaluation of the SoMeSci dataset, our system operates in a zero-shot setting due to the dataset's different domain. Remarkably, the system demonstrated solid performance, particularly in software mentions when assessed using Partial Matching (Appendix D, Figures 13-14).

Direct comparison between our system and SoMeNLP is challenging due to significant differences in model metrics. SoMeNLP excels at analyzing biomedical publications for program-specific software, whereas our system identifies a broader range of software categories, including machine learning models, algorithms, and architectures. This indicates that our system's capabilities are not fully represented when evaluated solely with the SoMeSci dataset.

Examining the SoMeSci dataset, we observed that software type mentions (e.g., usage) are categorized at a sentence level, which limits document-level Research Artifact Analysis (RAA). Document-level analysis requires synthesizing sentence-level mentions into cohesive document-

level annotations. Our system's deduplication pipeline is designed to address this challenge, consolidating both named and unnamed software mentions into singular, document-wide annotations that capture all their metadata. However, fully realizing the potential of this capability requires further experimentation.

5 Related Work

Our study provided an in-depth examination of RA mention datasets, focusing on DMDD and SoMeSci, which target dataset and software mentions. However, the research landscape includes numerous other RA mention datasets proposed in recent literature. Some datasets focus on abstract RAs, capturing elements like materials, methods, metrics, and tasks (Augenstein et al., 2017; Luan et al., 2018; Jain et al., 2020; Färber et al., 2021; Zhao et al., 2019), while others emphasize on tangible RAs, specifically datasets (Heddes et al., 2021; Lafia et al., 2021) and software (Gupta and Manning, 2011; Istrate et al., 2022).

RAA within the specified datasets predominantly uses NER methods. Leading techniques in this domain include Recurrent Neural Networks (RNNs) (Hopfield, 1982) and BERT-based models (Devlin et al., 2019). Specifically, the extraction of dataset and software mentions has been primarily conducted using Long Short-Term Memory (LSTM) models (Hochreiter and Schmidhuber, 1997) paired with conditional random fields (CRFs) (Lafferty et al., 2001; Prasad et al., 2019; Schindler et al., 2020; Hou et al., 2022; Luan, 2018; Zeng and Acuna, 2020). Additionally, transformer-based models (Vaswani et al., 2017) such as BERT and SciBERT (Beltagy et al., 2019; Schindler et al., 2021; Färber et al., 2021) have been applied to these tasks. Notably, some works even combine both techniques (Pan et al., 2023; Heddes et al., 2021; Wadden et al., 2019).

Several models with complex architectures have been developed that perform NER by incorporating additional features for a more thorough analysis. Among them, the SoMeNLP model (Schindler et al., 2021) stands out for integrating a relation extraction component and a hierarchical clustering-based disambiguation mechanism (Schindler et al., 2021, 2020). This approach enables the extraction of enriched metadata from both text and external knowledge bases. Similarly, the Softcite service (Lopez et al., 2021) leverages a GROBID module to identify software mentions and extract associated metadata.

Training models on RA datasets introduces inherent biases, primarily stemming from their scope. While datasets such as DMDD (Pan et al., 2023) and CZ Software Mentions (Istrate et al., 2022) offer extensive coverage, their emphasis on particular scientific domains can cause biases. For instance, models like SoMeNLP show a strong preference for Life Sciences. Various models trained on RA datasets cover a broad range of scientific disciplines, from Biomedical fields (Duck et al., 2013; Schindler et al., 2021) to Economic Science (Du et al., 2021) and Computer Science (Heddes et al., 2021; Luan et al., 2018). Additionally, only a few studies, notably the Softcite and SoMeSci datasets (Du et al., 2021; Schindler et al., 2021), have tackled the complex task of metadata extraction and linking.

6 Discussion and Conclusions

In this work, we introduced a novel end-to-end system utilizing fine-tuned LLMs to effectively extract RAs from scientific literature. By employing two fine-tuned Flan-T5 models, we demonstrated the potential of even smaller models to perform RAA. This advancement is particularly significant for research teams with limited resources, as it can facilitate the reproducibility and reusability of RAs. Moreover, our system has the potential to revolutionize editorial reviews by detecting unnamed RA mentions, highlighting critical information gaps.

The performance of our system varied across scientific domains due to the field-specific nature of the keywords and gazetteers used in candidate detection and RA mention identification. This variation underscores the need for tailored model adjustments for each scientific field.

The performance of the LoRA-Hy and LoRA-Sy models, both based on the Flan-T5 Base model, is influenced by their training data quality. The inclusion of real-world mentions in the Hybrid dataset used for fine-tuning the LoRA-Hy model resulted in superior performance compared to the LoRA-Sy model, trained only on synthetic data. This highlights the importance of real-world data in improving model effectiveness.

Looking ahead, enhancing our system involves two key areas: expanding the datasets with diverse real-world RA mentions to improve model performance, and exploring newer, state-of-the-art LLMs. These steps promise to refine our tools and deepen our understanding of LLMs in RAA.

Limitations

Despite the promising results demonstrated by our novel end-to-end system for Research Artifact Analysis (RAA), several limitations must be acknowledged. These limitations highlight areas for further improvement and refinement.

Our system's performance is influenced by the discipline-specific nature of keywords and gazetteers used in candidate detection and RA mention identification. While effective within certain domains, its generalizability across all scientific disciplines is limited, requiring further adaptation for broader applicability. Additionally, the quality and diversity of the training data significantly impact performance. The LoRA-Sy and LoRA-Hy models are trained on datasets specific to certain disciplines, which may not fully represent all possible RA mentions, leading to biases.

Furthermore, we currently categorize RAs solely as software or datasets, which limits the system's comprehensiveness. Expanding our categorization to include a broader range of RA types, which vary by discipline, would enhance the system's applicability across diverse scientific fields.

Errors in candidate detection and RA validation can propagate through the pipeline, affecting the accuracy of identified RA mentions and extracted metadata. Focusing on the early stages of the pipeline and introducing stricter thresholds could help mitigate error propagation and improve overall performance. Our evaluation metrics, while comprehensive, might not capture all complexities of RAA tasks, as research artifacts can be found under many alternative names. Developing more standardized benchmarks, comprehensive metrics, or even performing human evaluations would provide a clearer picture of system capabilities and areas needing improvement. Comparing our system's results with top-performing models is challenging due to differences in task formats, further complicating the evaluation process.

Our system, based on Large Language Models (LLMs) and the instruction-based Question Answering (QA) task, requires significant computational resources. Larger models, such as Flan-T5 XL, which show improved accuracy, necessitate substantial resources for both training and deployment. Additionally, the system's design requires multiple prompts to the LLM, which is time-consuming. Balancing model size, efficiency, and performance remains a challenge. Investing in more real-world instances from diverse scientific disciplines and optimizing smaller LLM models might be a more practical approach for enhancing performance without excessive resource demands.

Addressing these limitations involves creating discipline-specific gazetteers, annotating and curating real-world examples for more effective training, refining candidate detection with stricter thresholds to prevent error propagation, balancing computational efficiency with performance, and developing more standardized evaluation benchmarks. Additionally, exploring state-of-the-art LLMs that offer an optimal balance of size and performance could further enhance the system's capabilities.

Acknowledgments

We thank our reviewers for their constructive feedback. We also appreciate the insights from colleagues and collaborators that enriched this research. Specific acknowledgments and funding details will be provided upon acceptance.

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *Preprint*, arXiv:1903.10676.

Arie Cattan, Sophie Johnson, Daniel S Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. 2021. Scico: Hierarchical cross-document coreference for scientific concepts. In *3rd Conference on Automated Knowledge Base Construction*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Caifan Du, Johanna Cohoon, Patrice Lopez, and James Howison. 2021. Softcite dataset: A dataset of software mentions in biomedical and economic research publications. *Journal of the Association for Information Science and Technology*, 72(7):870–884.

Geraint Duck, Goran Nenadic, Andy Brass, David L Robertson, and Robert Stevens. 2013. Bionerds: Exploring bioinformatics' database and software use through literature mining. *BMC Bioinformatics*, 14(1).

Michael Färber, Alexander Albers, and Felix Schüber. 2021. Identifying used methods and datasets in scientific publications. In *Proceedings of the Workshop on Scientific Document Understanding: co-located with 35th AAAI Conference on Artificial Inteligence (AAAI 2021) ; Remote, February 9, 2021. Ed.: A. P. B. Veyseh*, volume 2831 of *CEUR Workshop Proceedings*. RWTH Aachen.

S. Gupta and Christopher D. Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *International Joint Conference on Natural Language Processing*.

Jenny Heddes, Pim Meerdink, Miguel Pieters, and maarten marx. 2021. The automatic detection of dataset names in scientific articles. *Data*, 6:84.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

J J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

Linlin Hou, Ji Zhang, Ou Wu, Ting Yu, Zhen Wang, Zhao Li, Jianliang Gao, Yingchun Ye, and Rujing Yao. 2022. Method and dataset entity mining in scientific literature: A cnn + bilstm model with self-attention. *Knowledge-Based Systems*, 235:107621.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Ana-Maria Istrate, Donghui Li, Dario Taraborelli, Michaela Torkar, Boris Veytsman, and Ivana Williams. 2022. A large dataset of software mentions in the biomedical literature. *Preprint*, arXiv:2209.00693.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Frank Krüger and David Schindler. 2020. A literature review on methods for the extraction of usage statements of software and data. *Computing in Science & Engineering*, 22:26–38.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sara Lafia, Jeongwoo Ko, Elizabeth Moss, Jinseok Kim, Andrea K. Thomer, Libby, and Hemphill. 2021. Detecting informal data references in academic literature.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Patrice Lopez, Caifan Du, Johanna Cohoon, Karthik Ram, and James Howison. 2021. Mining software entities in scientific literature: Document-level ner for an extremely imbalance and large-scale task. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 3986–3995, New York, NY, USA. Association for Computing Machinery.

Yi Luan. 2018. Information extraction from scientific literature for method recommendation. *ArXiv*, abs/1901.00401.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

Huitong Pan, Qi Zhang, Eduard Constantin Dragut, Cornelia Caragea, and Longin Jan Latecki. 2023. Dmdd: A large-scale dataset for dataset mentions detection. *ArXiv*, abs/2305.11779.

Animesh Prasad, Chenglei Si, and Min-Yen Kan. 2019. Dataset mention extraction and classification. In *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications*, pages 31–36, Minneapolis, Minnesota. Association for Computational Linguistics.

Justin Reppert, Ben Rachbach, Charlie George, Luke Stebbing, Jungwon Byun, Maggie Appleton, and Andreas Stuhlmüller. 2023. Iterated decomposition: Improving science qa by supervising reasoning processes.

David Schindler, Felix Bensmann, Stefan Dietze, and Frank Krüger. 2021. Somesci- a 5 star open data gold standard knowledge graph of software mentions in scientific articles. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4574–4583, New York, NY, USA. Association for Computing Machinery.

David Schindler, Benjamin Zapilko, and Frank Krüger. 2020. Investigating software usage in the social sciences: A knowledge graph approach. *The Semantic Web*, 12123:271 – 286.

Petros Stavropoulos, Ioannis Lyris, Natalia Manola, Ioanna Grypari, and Haris Papageorgiou. 2023. Empowering knowledge discovery from scientific literature: A novel approach to research artifact analysis.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Yuzhuo Wang, Chengzhi Zhang, and Kai Li. 2022. A review on method entities in the academic literature: extraction, evaluation, and application. *Scientometrics*, 127:2479 – 2520.

Tong Zeng and Daniel Ernesto Acuna. 2020. Finding datasets in publications: the syracuse university approach.

He Zhao, Zhunchen Luo, Chong Feng, Anqing Zheng, and Xiaopeng Liu. 2019. A context-based framework for modeling the role and function of on-line resource citations in scientific literature. 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 5206–5215, Hong Kong, China. Association for Computational Linguistics.

A System Architecture Overview

In this appendix, we provide detailed figures illustrating the system pipeline. The figures show the system's architecture in more detail, using color coding to distinguish different elements. The green dotted boxes indicate sections, the blue dotted boxes indicate paragraphs, the black boxes indicate sentences, and the black rounded boxes indicate RA mentions. The red boxes indicate irrelevant paragraphs or sentences with no RA mentions.

Figure 5 illustrates the Candidate Detection using the system's gazetteers, keywords, and key phrases, as well as the Paragraph Relevance Check for all paragraphs of a publication, which we receive in a structured layout format (sections, paragraphs, and sentences). It also shows the RA validation process to determine which candidate RA mentions are valid and which are merely generic references.

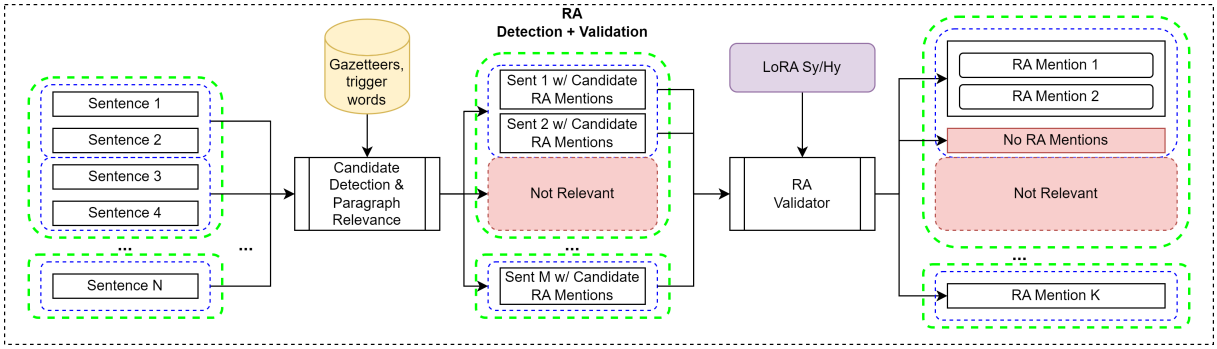

Figure 5: Candidate Detection, Paragraph Relevance and RA validation

Figure 6 illustrates the RA Metadata Extraction and RA Usage and Provenance Classification for the valid RA mentions. This phase involves extracting key metadata such as name, version, license, and URL for each RA mention. Additionally, it classifies each RA mention based on its usage (e.g., whether the RA is used in the study) and provenance (e.g., whether the RA was created by the authors).

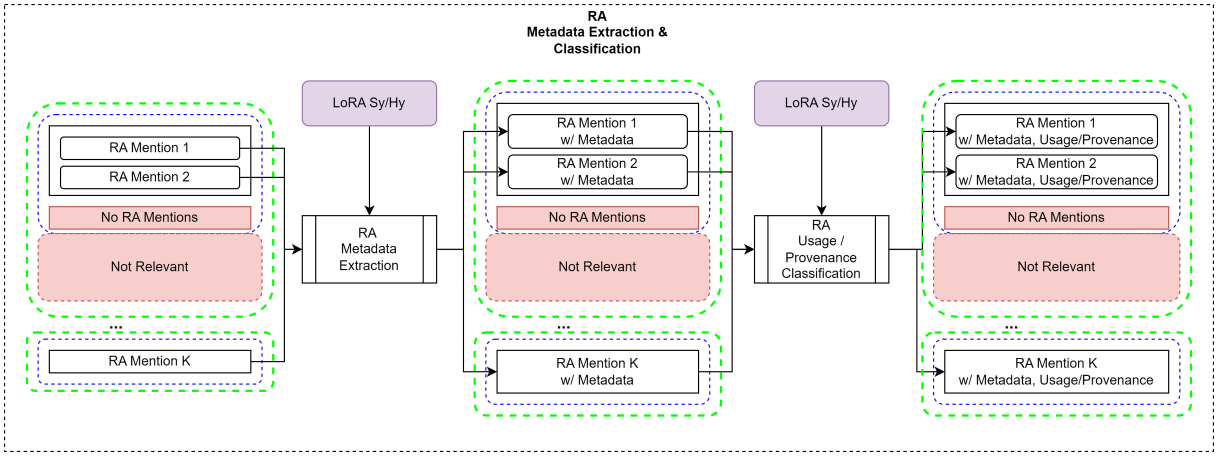

Figure 6: RA Metadata Extraction and RA Usage and Provenance Classification.

Figure 7 illustrates the clustering and deduplication of the RA mentions (both named and unnamed) into unique RAs. This process involves clustering similar RA mentions using their metadata and the SciCo model to aggregate all relevant metadata. It ensures each RA is distinctly represented and includes the reevaluation of their usage and provenance based on their clusters.

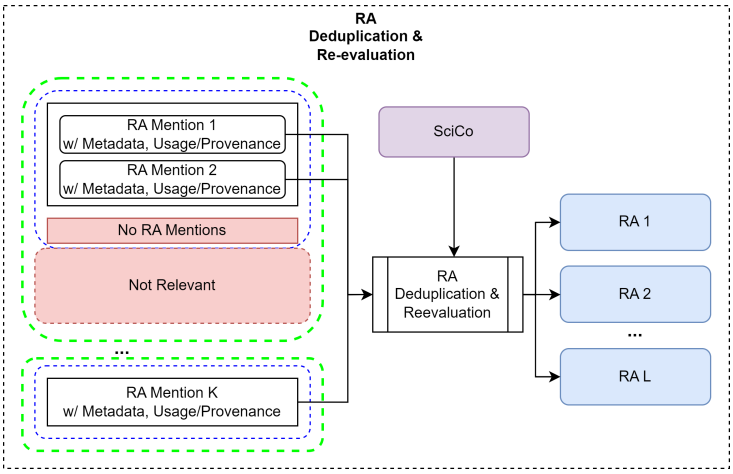

Figure 7: Clustering and deduplication of RA mentions into unique RAs, with usage and provenance reevaluation.

B Classification Protocol for LLMs

Our end-to-end system employs a unique classification protocol for Large Language Models (LLMs). This protocol leverages the probability scores of text generated by LLMs, converting unrestricted outputs into a controlled set of answers or choices. This methodology is critical in reducing the tendency of LLMs to generate 'hallucinations' or inaccurate information.

The protocol builds upon the method outlined in (Reppert et al., 2023). By incorporating this method, we calculate scores based on the probabilities of predefined answers generated by the LLM. This scoring mechanism is particularly advantageous during the RA validation and RA classification phases, converting these tasks into conventional classification tasks and facilitating the use of output thresholds.

An essential component of this protocol is the calculation of relative probabilities for a set of choices based on LLM predictions. The process begins with tokenizing each choice and calculating an absolute score for each tokenized choice. The score for a choice c is computed by summing the probabilities of tokens in the LLM's predictions that match the tokens in the tokenized choice.

Mathematically, this is expressed as:

$$abs\_probs[c] = \sum_{t \in c} \sum_{i=0}^{n} predictions[i][t]$$

where abs_probs is a dictionary storing the absolute probabilities of the choices, n is the length of the predictions, and $predictions[i][t]$ denotes the probability of token t at position i in the predictions.

After computing the absolute probabilities for all choices, the algorithm normalizes these scores to derive relative probabilities. This normalization is performed by dividing each absolute score by the sum of all absolute scores, Z, which serves as a normalization factor. The relative probability for choice c is calculated as:

$$rel\_probs[c] = \frac{abs\_probs[c]}{Z}$$

where

$$Z = \sum_{c} abs\_probs[c]$$

A known limitation of this approach is its bias towards longer choices, as they contain more tokens and thus accumulate a higher absolute score. This can lead to an overestimation of the probability for longer choices. Nevertheless, experimental observations indicate that the initial tokens in the choices have a more significant impact on the absolute score. This mitigates the bias towards longer choices to some extent, as the leading tokens contribute more substantially to the score. However, in scenarios where two choices begin with the same tokens, the ability to differentiate based on subsequent tokens is preserved, though with reduced impact.

C Additional Experimental Results

			DMDD-E+	SoMeSci_test+					
Model	**Scoring Method**	**Metric**	**mention**	**mention**	**url**	**version**	**license**	**usage**	**provenance**
Flan-T5 Base	**Exact Match**	Accuracy	0.214	0.116	-	-	-	-	-
		Precision	0.836	0.682	0.091 (0.011)	0.059 (0.007)	0.000 (0.000)	0.000 (0.000)	0.737 (0.085)
		Recall	0.223	0.120	0.333 (0.039)	0.333 (0.039)	0.000 (0.000)	0.000 (0.000)	0.560 (0.065)
		F1	0.352	0.205	0.143 (0.017)	0.100 (0.012)	0.000 (0.000)	0.000 (0.000)	0.636 (0.074)
	Partial Match	Accuracy	0.240	0.177	-	-	-	-	-
		Precision	0.902	0.957	0.140 (0.025)	0.349 (0.062)	0.136 (0.024)	1.000 (0.177)	0.519 (0.092)
		Recall	0.250	0.179	0.583 (0.103)	0.846 (0.150)	0.750 (0.133)	0.060 (0.011)	0.560 (0.099)
		F1	0.392	0.301	0.226 (0.040)	0.494 (0.088)	0.231 (0.041)	0.114 (0.020)	0.538 (0.095)
Flan-T5 XL	**Exact Match**	Accuracy	0.607	0.269	-	-	-	-	-
		Precision	0.825	0.726	0.469 (0.126)	0.517 (0.139)	0.077 (0.021)	0.958 (0.258)	0.919 (0.247)
		Recall	0.634	0.277	0.789 (0.212)	1.000 (0.269)	0.250 (0.067)	0.919 (0.247)	0.654 (0.176)
		F1	0.717	0.401	0.588 (0.158)	0.682 (0.183)	0.118 (0.032)	0.938 (0.252)	0.764 (0.206)
	Partial Match	Accuracy	0.683	0.400	-	-	-	-	-
		Precision	0.884	0.985	0.524 (0.210)	0.648 (0.259)	0.190 (0.076)	0.972 (0.389)	0.900 (0.360)
		Recall	0.713	0.398	0.880 (0.352)	0.920 (0.368)	1.000 (0.400)	0.926 (0.370)	0.643 (0.257)
		F1	0.790	0.567	0.657 (0.263)	0.760 (0.304)	0.320 (0.128)	0.948 (0.379)	0.750 (0.300)
LoRA-Sy	**Exact Match**	Accuracy	0.791	0.592	-	-	-	-	-
		Precision	0.734	0.657	0.689 (0.408)	0.660 (0.391)	0.000 (0.000)	0.902 (0.534)	0.898 (0.532)
		Recall	0.826	0.608	0.721 (0.427)	0.814 (0.482)	0.000 (0.000)	0.812 (0.480)	0.557 (0.330)
		F1	0.777	0.632	0.705 (0.417)	0.729 (0.432)	0.000 (0.000)	0.854 (0.506)	0.688 (0.407)
	Partial Match	Accuracy	0.836	0.777	-	-	-	-	-
		Precision	0.765	0.821	0.722 (0.561)	0.763 (0.593)	0.250 (0.194)	0.921 (0.716)	0.887 (0.689)
		Recall	0.873	0.771	0.736 (0.572)	0.733 (0.569)	0.857 (0.666)	0.837 (0.650)	0.553 (0.430)
		F1	0.816	0.795	0.729 (0.566)	0.747 (0.581)	0.387 (0.301)	0.877 (0.682)	0.681 (0.529)
LoRA-Hy	**Exact Match**	Accuracy	0.812	0.606	-	-	-	-	-
		Precision	0.718	0.631	0.660 (0.400)	0.708 (0.429)	0.000 (0.000)	0.930 (0.563)	0.929 (0.563)
		Recall	0.848	0.631	0.778 (0.471)	0.810 (0.491)	0.000 (0.000)	0.794 (0.481)	0.627 (0.380)
		F1	0.778	0.631	0.714 (0.433)	0.756 (0.458)	0.000 (0.000)	0.857 (0.519)	0.748 (0.453)
	Partial Match	Accuracy	0.859	0.801	-	-	-	-	-
		Precision	0.750	0.782	0.656 (0.525)	0.793 (0.635)	0.227 (0.182)	0.941 (0.754)	0.898 (0.720)
		Recall	0.897	0.791	0.769 (0.616)	0.644 (0.515)	1.000 (0.801)	0.842 (0.675)	0.631 (0.505)
		F1	0.817	0.786	0.708 (0.567)	0.710 (0.569)	0.370 (0.297)	0.889 (0.712)	0.741 (0.594)

Table 2: Experimental results on the DMDD-E+ and SoMeSci_test+ datasets, using Micro instead of Macro metrics.

D Additional Examples

Snippet	We used AmoebaNet-A as a teacher with 83.9% of ImageNet validation top-1 accuracy.
Candidate triggers	`[dataset]: ImageNet`
External gazetteers	`[software]: AmoebaNet-A`
Candidate RA Mention	We used <m>AmoebaNet-A</m> as a teacher with 83.9% of ImageNet validation top-1 accuracy.
Type	software
Name	AmoebaNet-A
License	N/A
Version	N/A
URL	N/A
Usage	Yes
Provenance	No

Figure 8: An example of a successful software identification in DMDD using the 'special' question of LoRA-Hy. No keyword or keyphrase from the PwC gazetteer was present, yet the correct instance was found.

Snippet 1	To evaluate the perception modules, we make use of three datasets with different sensory modalities: The Berlin Emotional Speech Database (EmoDB) [17] corpus is used to train and evaluate the auditory channel, the Face Expression Recognition Plus dataset (FER+) [9] corpus is used for the visual channel and The One-Minute Gradual-Emotional Behavior dataset (OMG-Emotion dataset) [7] is used for the cross-channel evaluation and the emotional concept clustering.
Snippet 2	To solve this, the FER+ dataset [9] was proposed.
RA names	Face Expression Recognition Plus dataset | FER+ dataset | FER+ corpus
Citation mark	[9]

Figure 9: Example of deduplication where two snippets with the same citation mark are consolidated. The snippets refer to the same dataset, the FER+ dataset, with one using the complete name and the other using a shorthand. Despite these differences, they are correctly identified as references to the same dataset.

Snippet	Table 5 shows that DivCNN performs better than best baselines on NEWSROOM, REDDIT and reaches incredible ROUGE scores more than 60 (but no baseline is reported in the dataset paper so the result is not comparable).
Gold RAs (datasets)	NEWSROOM, REDDIT
Flan-T5 XL Predictions	-
LoRA-Hy Predictions	NEWSROOM, REDDIT

Figure 10: Comparison of gold targets and predictions from Flan-T5 XL and LoRA-Hy models for a DMDD-E+ test set instance. The example shows that the Flan-T5 XL model fails to predict any datasets, while the LoRA-Hy model successfully predicts NEWSROOM and REDDIT. This indicates that the XL model struggles in the absence of obvious trigger phrases.

Snippet	We select four widely used pedestrian datasets, namely Virtual [21] , INRIA [22] , Daimler [23] and KITTI [24] , to evaluate the RF-DA methods.
Gold RAs (datasets)	Virtual, INRIA, Daimler, KITTI
Flan-T5 XL Predictions	KITTI
LoRA-Hy Predictions	Virtual, INRIA, Daimler, KITTI

Figure 11: Comparison of gold targets and predictions from Flan-T5 XL and LoRA-Hy models for a DMDD-E+ test set instance. This example demonstrates that the Flan-T5 XL model only predicts KITTI, whereas the LoRA-Hy model correctly identifies all four datasets: Virtual, INRIA, Daimler, and KITTI. This indicates that the XL model struggles in scenarios involving multiple datasets.

Snippet	Picture luminance was calculated with Adobe Photoshop CS2 (Adobe Systems Inc., USA) in 0-255 gray scale.
Gold RAs (software)	Photoshop
LoRA-Hy Prediction	Adobe Photoshop CS2
Exact Match	False
Partial Match	True

Figure 12: Example from SoMeSci_test+ that the Partial Match is more effective in capturing model performance, as it allows adjacent words in RA mentions, thus avoiding penalizing correct predictions that are slightly broader in scope, reducing false negatives.

Snippet	Real-time PCR gene-specific primers for s100a8, s100a9, and -actin were designed using Oligo Calc (Kibbe, 2007) as follows: s100a8, 5'-ACCATGCCCTCTACAAGAATGACT-3'; 5'-ACTCCTTGTGGCTGTCTTTGTG-3'; s100a9, 5'-AACCAGGACAATCAGCTGAGCTTT-3'; 5'-AGGCCATTGAGTAAGCCATTCCC-3'; -actin, 5'-ACCACAGCTGAGAGGGAAATCGT-3'; 5'-AACCGCTCGTTGCCAATAGTGA-3'.
Gold RAs (software)	Oligo Calc
Flan-T5 XL Prediction	-
LoRA-Hy Prediction	Oligo Calc

Figure 13: Example from SoMeSci_test+ in the biomedical domain where our system performed well, accurately identifying the reference to the software.

Snippet	To estimate the cumulative incidence of T2D within strata defined by quartiles of the genetic risk score (cutoffs derived from the distribution in the sub-cohort) and modifiable risk factors, we used the Stata bsample command to recreate the full cohort by resampling with replacement from the sub-cohort, according to the distributions of the stratum variables within the sub-cohort.
Gold RAs (software)	Stata
Flan-T5 XL Prediction	Stata
LoRA-Hy Prediction	Stata bsample

Figure 14: Example from SoMeSci_test+ in the biomedical domain where our system performed well, accurately identifying the reference to Stata using partial matching.