LREC 2026

Natural Scientific Language Processing (NSLP 2026) @ LREC 2026

Workshop Proceedings

Editors
Georg Rehm, Stefan Dietze, Danilo Dessí,
Diana Maynard, Sonja Schimmler

12 May 2026

ISBN 978-2-493814-61-6

Preface

Scientific research is almost exclusively published in unstructured text formats, which are not readily machine-readable. While technological approaches can help to get this flood of scientific information and new knowledge under control, the development of such technologies is very complex in practice and hinders the creation of infrastructures and systems to track research and assist the scientific community with applications such as dedicated scientific search engines and recommender systems. The 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026) aims to bring together researchers working on the processing, analysis, transformation and making-use-of scientific language including all relevant sub-topics. NSLP 2026 was a full-day workshop co-located with LREC 2026 held in Palma de Mallorca, Spain in May 2026.

In addition to the opportunity to submit papers covering original research that fits the workshop's topics of interest, the event also offered two shared tasks: Scientific Fact-Checking of Social Media Claims (ClimateCheck 2026) and Software Mention Disambiguation and Coreference Resolution (SOMD 2026). This proceedings volume contains two overview articles that describe these two shared tasks in more detail as well as several papers submitted to the two shared tasks.

With NSLP 2026 as the third edition of this workshop series, we were happy about a total of 38 submissions out of which 29 papers were accepted (76%) after a thorough, double-blind peer-review process with three reviews per submission. The NSLP 2026 workshop consisted of paper and poster presentations as well as two invited keynote presentations, given by Yufang Hou, ITU Austria, and Iryna Gurevych, TU Darmstadt, Germany. The abstracts of the two keynote presentations are included in this proceedings volume.

The workshop programme is structured into three sessions and a total of four tracks. Session 1 includes the two *Shared Task Overview Papers*. Session 2, *Scholarly Infrastructure and Data Foundations*, consists of two tracks. Track A, *Knowledge Graphs and Taxonomy*, builds what we can call the research or scientific knowledge graph, while Track B, *Domain-Specific Resources and Metadata*, populates the graph with data. Session 3, *Reliability, Reach and Scale*, concentrates on two tracks: Track C, *Factuality and Agentic Reasoning* asks if the data is true and Track D, *Multilingualism and Model Optimization*, asks how we scale that truth globally.

We would like to thank the LREC 2026 organisers and workshop chairs for accepting our workshop proposal. We would also like to extend our gratitude to our keynote speakers for their insightful talks. Thanks are also due to the members of the Programme Committee for reviewing the paper submissions under rather tight deadlines. Finally, we would like to thank Raia Abu Ahmad and Ekaterina Borisova for setting up and maintaining the workshop website.[1]

This workshop was organised under the umbrella of the project NFDI for Data Science and Artificial Intelligence (NFDI4DS), which is part of the wider German NFDI initative (National Research Data Infrastructure). Without the financial support of this project, the NSLP 2026 workshop would not have been possible.

Georg Rehm, Stefan Dietze, Danilo Dessí, Diana Maynard, Sonja Schimmler

[1] https://nfdi4ds.github.io/nslp2026/

Organizing Committee

- Georg Rehm – Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI) & Humboldt-Universität zu Berlin, Germany
- Stefan Dietze – GESIS Leibniz Institut für Sozialwissenschaften, Cologne & Heinrich-Heine-Universität Düsseldorf, Germany
- Danilo Dessí – University of Sharjah, UAE
- Diana Maynard – University of Sheffield, UK
- Sonja Schimmler – Technical University of Berlin & Fraunhofer FOKUS, Germany

Programme Committee

- Raia Abu Ahmad – Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
- Marcel R. Ackermann – Schloss Dagstuhl, Leibniz Center for Informatics (LZI), DBLP, Germany
- Sören Auer – TIB Leibniz Information Centre for Science and Technology, Germany
- Fabio Barth – Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
- Ekaterina Borisova – Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
- Davide Buscaldi – LIPN, CNRS, University Paris 13, France
- Leyla Jael Castro – ZB MED Information Centre for Life Sciences, Germany
- Mathieu d'Aquin – Université de Lorraine, France
- Catherine Faron – Université Cote d'Azur, France
- Paul Groth – University of Amsterdam, The Netherlands
- Robert Jäschke – Humboldt-Universität zu Berlin, Germany
- Frank Krüger – Wismar University of Applied Sciences, Germany
- Julia Matela – Wismar University of Applied Sciences, Germany
- Philipp Mayr – GESIS Leibniz-Institute for the Social Sciences, Germany
- Julian Moreno-Schneider – Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
- Pedro Ortiz Suarez – Common Crawl Foundation, USA
- Wolfgang Otto – GESIS Leibniz-Institute for the Social Sciences, Germany
- Haris Papageorgiou – R. C. Athena, Greece
- Harald Sack – FIZ Karlsruhe, Germany

- Angelo Salatino – The Open University, UK
- Philipp Schaer – TH Köln (University of Applied Sciences), Germany
- Vera Schmitt – Technische Universität Berlin, Germany
- Veronika Solopova – Technische Universität Berlin, Germany
- Sharmila Upadhyaya – GESIS Leibniz-Institute for the Social Sciences, Germany
- Max Upravitelev – Technische Universität Berlin, Germany
- Ricardo Usbeck – Leuphana University, Germany
- Aida Usmanova – Leuphana University, Germany
- Thanasis Vergoulis – R. C. Athena, Greece

Table of Contents

Workshop Program

09:00-09:10 Welcome and Opening

09:10-09:50 Invited Keynote Speech – Yufang Hou

Session 1: Shared Task Overview Papers

09:50-10:10 *ClimateCheck 2026: Scientific Fact-Checking and Disinformation Narrative Classification of Climate-related Claims*
Raia Abu Ahmad, Max Upravitelev, Aida Usmanova, Veronika Solopova and Georg Rehm

10:10-10:30 *The Software Mention Detection and Coreference Resolution Shared Task 2026*
Sharmila Upadhyaya, Wolfgang Otto, Julia Matela, Frank Krüger and Stefan Dietze

10:30-11:00 Coffee Break and Poster Session

Session 2: Scholarly Infrastructure and Data Foundations
Track A: Knowledge Graphs and Taxonomy

11:00-11:20 *Comparing LLM-Based Knowledge Graph Extraction Approaches on Literary Studies in Spanish: A Case Study on Orbis Tertius*
Federico Cortes

11:20-11:40 *Enhancing Scholarly Knowledge Graphs via Domain-Specific Entity Detection and Linking*
Nicolau Duran-Silva, César A. Parra-Rojas, Pablo Accuosto, Julian Moreno-Schneider and Georg Rehm

11:40-12:00 *UniCite: A Dataset and Unified Hierarchical Taxonomy for Multi-Dimensional Citation Analysis*
Amina Mourky, Elena Leitner, Julian Moreno-Schneider, Raia Abu Ahmad, Ekaterina Borisova and Georg Rehm

Track B: Domain-Specific Resources and Metadata

12:00-12:20 *Beyond Abstracts: A Biomedical MeSH Indexing Corpus Incorporating Summarized Methods Sections*
Sujoy Datta, Robert E. Mercer and Xindi Wang

12:20-12:40 *AstroConcepts: A Large-Scale Multi-Label Classification Corpus for Astrophysics*
Atilla Kaan Alkan, Felix Grezes, Sergi Blanco-Cuaresma, Jennifer Lynn Bartlett, Daniel Chivvis, Anna Kelbert, Kelly Lockhart and Alberto Accomazzi

12:40-12:50 *Normalizing Section Names and Structure of Scientific Articles*
Nicolau Duran-Silva, Julian Moreno-Schneider, César A. Parra-Rojas and Georg Rehm

12:50-13:00 *Generating Research Data Metadata from Their Accompanying README Files*
Kotaro Sekido, Yu Watanabe, Koichiro Ito and Shigeki Matsubara

13:00-14:00 Lunch Break

Session 3: Reliability, Reach and Scale
Track C: Factuality and Agentic Reasoning

14:00-14:20 *Enhancing Factuality and Transparency in Generative Models for Biomedical Question Answering*
Ankita Behura, Siting Liang and Daniel Sonntag

14:20-14:40 *The Linguist's Lie Detector: Linguistic Knowledge in Large Language Models*
Lucía Catalán Gris, Kim Gerdes and John S. Y. Lee

14:40-15:00 *Improving Completeness in Deep Research Agents through Targeted Enrichment*
Jesse Wonnink, Jakub Zavrel and Paul Groth

Track D: Multilingualism and Model Optimization

15:00-15:20 *Transferring Scientific English Pre-Trained Language Models to Multiple Languages Using Cross-Lingual Transfer*
Nikolas Ching-Pu Rauscher, Fabio Barth and Georg Rehm

15:20-15:40 *Evaluating Generative Large Language Models for Portuguese Scientific Information Extraction*
Tomás Pinto, Catarina Silva and Hugo Goncalo Oliveira

15:40-16:00 *Do We Need Bigger Models for Science? Task-Aware Retrieval with Small Language Models*
Florian Kelber, Matthias Jobst, Yuni Susanti and Michael Färber

16:00-16:30 Coffee Break and Poster Session

16:30-17:15 Invited Keynote Speech – Iryna Gurevych

17:15-17:30 Closing and end of workshop

Poster Session

Benchmarking LLMs for ARR Area Assignment: Evidence and Implications for Assignment Strategies
Eileen Bingert, Diego Alves and Stefania Degaetano-Ortlieb

Benchmarking Retrieval-Augmented Generation for Scientific Knowledge QA in European Portuguese
Jose Matos, Catarina Silva and Hugo Goncalo Oliveira

Challenges and Opportunities for NSLP in Scientific Publishing – A Case Study
Thomas Kleinbauer, Michael Didas and Michael Wagner

ClimateSense at ClimateCheck 2026
Thibault Ehrhart, Gregoire Burel and Raphael Troncy

Demystifying Funding: Reconstructing a Unified Dataset of the UK Funding Lifecycle
William Thorne, Rupert Shepherd and Diana Maynard

Do Lexical and Contextual Coreference Resolution Systems Degrade Differently under Mention Noise? An Empirical Study on Scientific Software Mentions
Atilla Kaan Alkan, Felix Grezes, Jennifer Lynn Bartlett, Anna Kelbert, Kelly Lockhart and Alberto Accomazzi

EarlySciRev: A Dataset of Early-Stage Scientific Revisions Extracted from LaTeX Writing Traces
Léane Jourdan, Julien Aubert-Béduchaud, Yannis Chupin, Marah Baccari and Florian Boudin

From Slides to Chatbots: Enhancing Large Language Models with University Course Materials
Tu Anh Dinh, Philipp Nicolas Schumacher and Jan Niehues

Identifying Implicit Research Data References in Paper Citations
Koshi Motegi, Koichiro Ito and Shigeki Matsubara

MioFFAn: An Annotation Software for Formula Formalization with LLM Automation Capabilities
Nicolas Sibuet Ruiz, Horacio Saggion and Riccardo Rossi

Retrieval-Augmented LLMs and Encoder Models for Multi-Label Climate Disinformation Narrative Classification
Neda Foroutan, Alexandra Tsiakalou and Vera Schmitt

Towards Efficient Self-Explainable Climate-Related Claim Verification with Generative Models
Siting Liang, Omar Adjali and Daniel Sonntag

Transformer Encoders with Heuristic-Guided Contrastive Learning for Software Coreference Resolution
Mahmoud Hassan and Dipendra Yadav

XplaiNLP @ ClimateCheck 2026 Task 2: Comparing Hierarchical Approaches for Fine-Grained Climate Disinformation Narrative Classification
Arthur Hilbert, Jing Yang and Vera Schmitt

Invited Keynote

Yufang Hou

ITU Austria, Austria

Synthesizing Scientific Knowledge: From Biomedical Evidence to NLP Claims

Abstract

The exponential growth of scholarly literature poses a substantial challenge for researchers seeking to stay current with the latest findings and synthesize knowledge effectively. In this talk, I will first present our recent studies on supporting domain experts in synthesizing biomedical research findings, guided by established evidence synthesis principles for integrating and interpreting potentially biased and conflicting scientific evidence. Next, I will introduce our work on content meta-analysis in NLP literature, including building leaderboards and tracking the evolution of scientific claims over time. Finally, I will discuss open research challenges in modeling and reasoning over scientific knowledge encoded in scholarly documents.

Bio

Yufang Hou is a professor at IT:U – Interdisciplinary Transformation University Austria. At IT:U, she leads the NLP group with a strong focus on large language model (LLM) governance, computational argumentation, fact-checking, knowledge representation and reasoning, and human-centered NLP applications in science, health, and education. Yufang has served as an organizing committee member and senior program committee member for various NLP conferences, and has co-organized several workshops on topics such as argument mining, efficient NLP, and AI for Science.

Invited Keynote

Iryna Gurevych

Technical University of Darmstadt, Germany

Welcoming AI as a New Colleague: How Should We Evaluate AI for Science?

Abstract

AI is reshaping the work of researchers in real time. With the emergence of tools such as AI Scientist in 2024, a new era of AI-driven scientific discovery has begun. At the same time, high-profile venues are beginning to experiment with AI-assisted peer review. Yet the core question remains unresolved: How should we meaningfully evaluate AI for scientific work? In this talk, I will highlight challenges encountered in several projects focused on AI-assisted scientific communication and discuss how these challenges inform evaluation practices. I will examine methods for assessing AI-generated related-work sections and explore how automated reviewers can be evaluated with respect to research-design reasoning and novelty assessment. Together, these examples illustrate both the promise and the complexity of evaluating AI as it increasingly acts as a collaborator in scientific communication.

Bio

Iryna Gurevych is Professor of Ubiquitous Knowledge Processing in the Department of Computer Science at the Technical University of Darmstadt in Germany. She also is an adjunct professor at MBZUAI in Abu-Dhabi, UAE, and an affiliated professor at INSAIT in Sofia, Bulgaria. She is widely known for fundamental contributions to natural language processing (NLP) and machine learning. Professor Gurevych is a past president of the Association for Computational Linguistics (ACL), the leading professional society in NLP. Her many accolades include being a Fellow of the ACL, an ELLIS Fellow, and the recipient of an ERC Advanced Grant. Most recently, she has received the 2025 Milner award of the British Royal Society for her major contributions to NLP and artificial intelligence that combine deep understanding of human language and cognitive faculty with the latest paradigms in machine learning.

AstroConcepts: A Large-Scale Multi-Label Classification Corpus for Astrophysics

Atilla Kaan Alkan[1], **Felix Grezes**[1], **Sergi Blanco-Cuaresma**[1,2], **Jennifer Lynn Bartlett**[1], **Daniel Chivvis**[1], **Anna Kelbert**[1], **Kelly Lockhart**[1], **Alberto Accomazzi**[1]

[1]Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA
[2]Faculty of Psychology, UniDistance Suisse, Brig, Switzerland
{atilla.alkan, felix.grezes, sblancocuaresma, jennifer.bartlett, daniel.chivvis, anna.kelbert, kelly.lockhart, alberto.accomazzi}@cfa.harvard.edu

Abstract

Scientific multi-label text classification suffers from extreme class imbalance, where specialized terminology exhibits severe power-law distributions that challenge standard classification approaches. Existing scientific corpora lack comprehensive controlled vocabularies, focusing instead on broad categories and limiting systematic study of extreme imbalance. We introduce AstroConcepts, a corpus of English abstracts from 21,702 published astrophysics papers, labeled with 2,367 concepts from the Unified Astronomy Thesaurus. The corpus exhibits severe label imbalance, with 76% of concepts having fewer than 50 training examples. By releasing this resource, we enable systematic study of extreme class imbalance in scientific domains and establish strong baselines across traditional, neural, and vocabulary-constrained LLM methods. Our evaluation reveals three key patterns that provide new insights into scientific text classification. First, vocabulary-constrained LLMs achieve competitive performance relative to domain-adapted models in astrophysics classification, suggesting a potential for parameter-efficient approaches. Second, domain adaptation yields relatively larger improvements for rare, specialized terminology, although absolute performance remains limited across all methods. Third, we propose frequency-stratified evaluation to reveal performance patterns that are hidden by aggregate scores, thereby making robustness assessment central to scientific multi-label evaluation. These results offer actionable insights for scientific NLP and establish benchmarks for research on extreme imbalance.

Keywords: Multi-label Text Classification, Scientific Document Classification, Extreme Label Imbalance

1. Introduction

Scientific multi-label text classification poses significant challenges due to extreme class imbalance, with specialized terminology exhibiting severe power-law distributions that challenge standard classification methods (Liu et al., 2023). While imbalanced distributions are common in scientific domains, existing corpora (Giles et al., 1998; McCallum et al., 2000; Yang et al., 2018) typically provide limited coverage of controlled vocabulary, focus on broad disciplinary categories, or operate at scales that hinder systematic methodological investigation of extreme imbalance scenarios. This limitation is particularly problematic for comprehensive scientific text classification, where controlled vocabularies organize thousands of specialized concepts with naturally occurring power-law distributions. Existing datasets that provide controlled-vocabulary coverage either operate at prohibitive computational scales (Toney and Dunham, 2022) or focus on limited subsets of domain terminology, thereby preventing systematic investigation of how different approaches address the fundamental challenge of learning from severely imbalanced specialized terminologies.

The astrophysics literature exemplifies these challenges while providing an ideal testbed for systematic investigation. The Unified Astronomy Thesaurus (UAT) (Accomazzi et al., 2014) organizes 2,367 concepts across 11 hierarchical levels, creating a comprehensive controlled vocabulary that exhibits natural power-law distributions characteristic of scientific domains.

To provide the community with essential resources for investigating extreme multi-label classification in scientific domains, we introduce AstroConcepts, a corpus of 21,702 astrophysics papers labeled with the complete UAT vocabulary (2,367 concepts). This resource enables systematic investigation of three fundamental research questions:

- **RQ1** How do traditional methods, supervised neural models, and vocabulary-constrained LLMs compare for handling extreme label imbalance in scientific classification?
- **RQ2** Does domain-specific pretraining provide uniform benefits across frequency bins, or do improvements concentrate in specific regions of the label distribution?
- **RQ3** Can vocabulary-constrained LLMs achieve competitive performance with

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 1–12
12 May 2026.

domain-adapted models in astrophysics classification?

Our main contributions are:

1. AstroConcepts[1] provides the community with the first tractable-scale corpus, enabling systematic investigation of extreme multi-label classification with comprehensive controlled-vocabulary coverage.
2. Systematic comparison across traditional, neural, and vocabulary-constrained LLM approaches reveals competitive performance for parameter-efficient methods, opening new research directions for scientific NLP.
3. Introduction of frequency-stratified evaluation framework with robustness metrics that reveal performance patterns invisible in aggregate scores, providing essential tools for extreme multi-label assessment.
4. Demonstration that domain adaptation improvements concentrate on rare specialized terminology, with implications for training strategies in scientific applications.
5. Comprehensive baseline establishment that provides essential benchmarks for scientific multi-label classification while revealing fundamental properties of extreme imbalance in specialized domains.

The remainder of this paper is structured as follows. Section 2 reviews related work in scientific multi-label classification and positions AstroConcepts within the existing landscape of scientific corpora. Section 3 describes the corpus construction methodology, annotation procedures, and key characteristics of the resulting dataset. Section 4 details our experimental setup, including baseline methods, evaluation metrics, and a frequency-stratified analysis framework. Section 5 presents results across all methods, revealing key patterns in overall performance and frequency-specific behaviors. Section 6 discusses the broader implications of our findings for scientific NLP, methodological contributions, and limitations of the current work. Section 7 concludes with a summary of key insights and directions for future research.

2. Scientific Multi-label Classification Benchmarks

Multi-label text classification spans diverse domains and scales. Early work established evaluation protocols with news categorization (Lewis, 1987), while legal document classification introduced a hierarchical structure through EUR-Lex (Loza Mencía and Fürnkranz, 2010), though annotations include complete paths rather than the flat terminal concepts typical in practice. Extreme multi-label benchmarks (Amazon-670K, Wiki-500K) scale to hundreds of thousands of labels but prioritize computational efficiency over domain-specific controlled vocabularies.

Scientific classification began with computer science papers: Giles et al. (1998) used six broad categories, McCallum et al. (2000) expanded to 70 categories with a 3-level hierarchy, and later work incorporated controlled vocabularies (Santos and Rodrigues, 2009; Kowsari et al., 2017; Yang et al., 2018), although these remained limited in scope. Recent efforts leverage Microsoft Academic Graph: Cohan et al. (2020) created embeddings across 19 fields, Sadat and Caragea (2022) scaled to 186K papers with 6-level hierarchy, and Toney and Dunham (2022) used 180–220M papers, though computational demands limit systematic experimentation. While MAG provides broad coverage, it spans multiple disciplines rather than offering deep domain specialization.

Table 1 puts AstroConcepts in context of existing resources. General benchmarks lack controlled vocabularies, whereas scientific corpora either use ad hoc categories, span multiple disciplines without deep specialization, or reach a prohibitive scale. AstroConcepts uniquely combines tractable scale (21K), deep hierarchy (11 levels), and domain-specific controlled vocabulary (UAT), enabling comprehensive evaluation of hierarchy-aware methods and few-shot learning in a realistic scientific classification scenario.

3. The AstroConcepts Corpus

AstroConcepts exhibits characteristics that make scientific multi-label classification challenging: a hierarchical controlled vocabulary (UAT) with flat expert annotations, severe label imbalance, and domain-specific terminology that requires specialized language understanding. This section describes the data collection process, the UAT taxonomy structure, the annotation methodology, and the comprehensive corpus statistics.

3.1. Source Data

We collected 21,702 abstracts from published English-language papers indexed by the NASA-funded Science Explorer (SciX; Bartlett et al. (2025)), an expansion of the Astrophysics Data System (ADS; Accomazzi et al. (2015)) to cover all NASA science disciplines. Building on the ADS legacy, SciX is the primary bibliographic database

[1] https://huggingface.co/datasets/adsabs/SciX_UAT_keywords

Corpus	Docs	Labels	Hierarchy	Supervision	Vocabulary	Domain
General Domain						
Loza Mencía and Fürnkranz (2010)	19K	7.2K	2-level	Hierarchical	EuroVoc	Legal
Lewis (1987)	21K	90	None	Flat	Ad-hoc	News
RCV1	800K	103	4-level	Flat	Ad-hoc	News
Scientific Domain						
Giles et al. (1998)	3K	6	None	Flat	Ad-hoc	CS
Santos and Rodrigues (2009)	15K	92	2-level	Mixed	ACM	CS
AstroConcepts	**21K**	**2.3K**	**11-level**	**Flat**	**UAT**	**Astrophys**
Cohan et al. (2020)	25K	19	1-level	Flat	MAG	Multi-sci
Kowsari et al. (2017)	47K	134	2-level	Hierarchical	WoS	CS+Med
McCallum et al. (2000)	53K	70	3-level	Hierarchical	Ad-hoc	CS
Yang et al. (2018)	55.8K	54	2-level	Flat	Ad-hoc	CS
Sadat and Caragea (2022)	186K	1.2K	6-level	Mixed	MAG	Multi-sci
Toney and Dunham (2022)	180M	313	2-level	Flat	MAG	Multi-sci

Table 1: Multi-label classification benchmarks sorted by corpus size. AstroConcepts provides tractable scale (21K documents) with deep hierarchical structure (11 levels) and domain-specific controlled vocabulary (UAT), enabling comprehensive experimentation on astrophysics literature classification.

for astronomy and astrophysics. It indexes papers from approximately 8,000 refereed journals, including the *Astrophysical Journal* (*ApJ*). Starting in 2018, journal editors began requiring or encouraging UAT concept assignment during submission to promote standardized concept usage in astronomy, resulting in a growing corpus of controlled-vocabulary annotations.

We selected papers meeting the following criteria: (1) published in journals where authors assign UAT concepts during submission, ensuring controlled standard concept annotation, (2) at least one UAT concept assigned, and (3) English-language abstracts. The temporal scope covers 2018-2023, reflecting the data available during corpus construction. Future work could extend coverage to more recent publications. This process yielded 21,702 documents.

3.2. The Unified Astronomy Thesaurus

The UAT (Accomazzi et al., 2014) is a community-maintained controlled vocabulary for astronomical literature, owned and openly licensed by the *American Astronomical Society*. It follows the Simple Knowledge Organization System (SKOS; Miles and Bechhofer (2009)) standard and incorporates terms from earlier astronomy thesauri. Astronomy librarians and domain experts have contributed to its development and maintenance. Released in 2017 and subsequently adopted by major journals and SciX, the UAT provides standardized terminology for indexing and retrieval of astronomy content.

The UAT version 5.1.0[2] used in this work contains 2,367 astronomical concepts organized in an eleven-level hierarchical structure forming a directed acyclic graph (DAG). Concepts range from broad top-level categories such as cosmology or observational astronomy, to highly specific concepts such as the Kreutz group. Each concept includes a unique identifier, a canonical designation, alternative names (synonyms, acronyms), and an optional textual definition (scope note), and explicit relationships to parents, children, and related concepts.

The UAT follows a polyhierarchical structure in which concepts can have multiple parents, creating a DAG topology in which nodes (concepts) can be reached through multiple paths without forming cycles. For example, *Stellar atmosphere* appears under both *Stars* and *Spectroscopy*, as studies of stellar atmospheres involve both stellar physics and spectroscopic methods. This DAG structure reflects the multidisciplinary nature of astronomical research, where concepts naturally belong to multiple semantic categories simultaneously.

3.3. Concept Assignment Process

Labels in AstroConcepts come from the standard publishing process where authors assign UAT concepts to their manuscripts during submission. Authors typically select 4 concepts per paper (see Table 2) using the journal's submission system. They choose specific concepts without marking hierarchical paths. For example, selecting *Exoplanet atmospheres* does not require selecting its broader categories, such as *Exoplanet astronomy*. This creates an interesting challenge: systems must predict specific concepts from a hierarchical vocabulary using only flat annotations. The approach has clear advantages: we collect high-quality labels from domain experts who know their work best, and the standardized UAT vocabulary ensures consistency.

[2] https://github.com/astrothesaurus/UAT/tree/v.5.0.0

However, authors naturally focus on what they consider most important rather than providing comprehensive coverage. This means some relevant concepts might be missing, creating a realistic yet challenging evaluation scenario that mirrors real-world conditions in which systems operate with an incomplete set of annotations.

3.4. Overall Statistics

Table 2 presents overall corpus statistics. AstroConcepts contains 21,702 abstracts with 93,547 total label assignments, averaging 4.31 labels per abstract. The assigned label space comprises 1,864 unique UAT concepts (92% of the UAT vocabulary), indicating that the selected literature abstracts span the conceptual space defined by UAT.

Statistic	Value
Documents	
Total abstracts	21,702
Labels	
Total assignments	93,547
Assigned unique labels	1,864
Per abstract (mean/median)	4.31 / 4
Per abstract (range)	1–12
Text	
Length in words (mean/median)	211.2 / 223
Length (range)	18–462
Vocabulary size	163,326

Table 2: Overall statistics for AstroConcepts.

Abstracts average 211 words (median: 223), typical for scientific abstracts and compatible with standard transformer context windows (512 tokens). The distribution is approximately normal with slight positive skew toward longer abstracts; 94% fit within 512 tokens. We retain abstracts up to 462 words to capture the full range of scientific writing styles without truncation artifacts.

The distribution of labels per abstract (mean: 4.31, median: 4, range: 1–12) reflects the multifaceted nature of astrophysics research. Most abstracts (70%) receive 2–5 labels, with single-label papers typically representing narrowly focused studies and papers with 6+ labels (15%) covering interdisciplinary or methodologically diverse work. This moderate label density exceeds general multi-label benchmarks such as Reuters-21578 (Lewis, 1987), which averages 1.2 labels per document (Huang et al., 2021), underscoring the conceptual complexity inherent in scientific literature.

3.5. Concept Frequency Distribution

A critical characteristic of AstroConcepts is severe label imbalance, typical of real-world multi-label scenarios. Figure 1 shows the label frequency distribution on a log-log scale, revealing a power-law pattern. The most frequent label (*Galaxy evolution*) appears in 1,106 abstracts (5.1%), while the median label appears in only 12 abstracts. We fit a power law $f(r) \propto r^{-\alpha}$ where r is rank and $f(r)$ is frequency, obtaining $\alpha = 1.50$ with $R^2 = 0.825$.

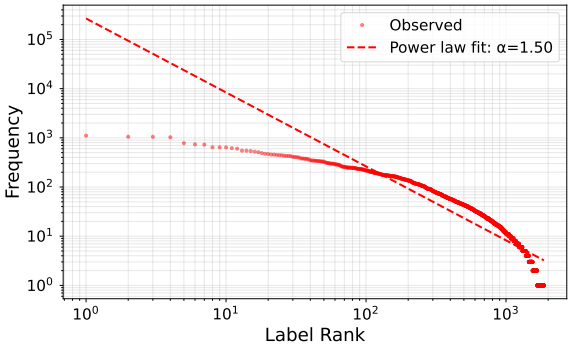


Figure 1: Label frequency distribution (log-log scale) and fitted power-law function with exponent $\alpha = 1.50$ ($R^2 = 0.825$). The long tail contains 76% of labels with fewer than 50 occurrences each, creating a severe class imbalance characteristic of scientific multi-label classification.

The exponent $\alpha = 1.50$ indicates a moderately steep long-tail distribution. This suggests that AstroConcepts presents a substantial but not extreme class imbalance compared to other scientific classification scenarios. The moderate slope means mid-frequency (torso) labels retain more training examples than in steeper distributions; by rank 100, labels still average 58 examples, whereas in datasets with $\alpha = 2.0$, rank-100 labels would have only 25 examples. Nevertheless, a severe imbalance remains: 76% of labels occur fewer than 50 times.

We partition labels into three frequency bins (see Table 3). Head labels (frequency > 500): 17 labels (0.9% of vocabulary) covering 12,288 assignments (13.1% of total). These represent core concepts studied extensively across astrophysics subfields. Torso labels ($50 \leq$ frequency ≤ 500): 429 labels (23.0%) covering 62,808 assignments (67.1%). These represent moderately common research topics with sufficient training examples for standard supervised learning. Tail labels (frequency < 50): 1,418 labels (76.1%) covering 18,451 assignments (19.7%). These represent specialized phenomena, emerging research areas, and niche methodologies with limited training examples, creating the zero-shot challenge we investigate in Section 4.

This distribution creates the multi-level challenge characteristic of extreme multi-label classification: head labels are easily learned from abundant examples (hundreds per label), torso labels require careful modeling to generalize from moderate data

Bin	# Labels	%	# Assign.	%
Head (>500)	17	0.9	12,288	13.1
Torso (50–500)	429	23.0	62,808	67.1
Tail (<50)	1,418	76.1	18,451	19.7
Total	**1,864**	**100.0**	**93,547**	**100.0**

Table 3: Label frequency distribution across bins. The long tail contains 76% of labels but only 20% of assignments, creating a severe class imbalance characteristic of scientific multi-label classification.

(50–500 examples), and tail labels present a few-shot scenario (fewer than 50 examples, median: 12) where standard supervised methods struggle.

The most frequent labels span major astrophysics subfields: extragalactic astronomy (*Galaxy evolution*, *Active galactic nuclei*), stellar physics (*Star formation*, *Neutron stars*), planetary science (*Exoplanets*, *Exoplanet atmospheres*), and observational methods (*Spectroscopy*, *Astronomy data analysis*). No single subfield dominates the top-20 labels, confirming that AstroConcepts captures the breadth of modern astrophysics research rather than concentrating on narrow phenomena. This diversity is important for multi-label classification research: the corpus contains both well-represented core concepts and sparse specialized topics, enabling evaluation across the full spectrum of label frequencies.

3.6. Concept Specificity Patterns

Table 4 reveals that authors prefer moderately specific concepts (peaking at level 4). Both very general concepts (levels 1-2) and highly specialized terminology (levels 6+) are underrepresented relative to mid-level concepts. This concentration around moderate specificity creates additional challenges for classification systems beyond frequency imbalance. Models must handle vocabularies in which training examples cluster around mid-level concepts, with limited examples at both ends of the taxonomic spectrum, requiring systems to predict across varying specificity levels with highly imbalanced training signals.

3.7. Concept Co-occurrence Patterns

We analyzed how concepts co-occur in our corpus. We computed pointwise mutual information (PMI) for concept pairs, where $\text{PMI}(\ell_i, \ell_j) = \log \frac{P(\ell_i,\ell_j)}{P(\ell_i)P(\ell_j)}$ measures whether two concepts appear together more frequently than expected by chance. Positive PMI indicates stronger-than-expected co-occurrence, while negative PMI suggests mutual exclusivity. Focusing on pairs with positive PMI and sufficient co-occurrence ($\geq$ 10 abstracts), we found that authors rarely select hierarchically related concepts together. Concepts from unrelated subtrees show 2.3× higher average PMI than parent-child pairs, indicating that authors typically choose concepts spanning multiple taxonomic branches rather than selecting both general and specific terms from the same hierarchy. This cross-branch annotation behavior represents a challenge for classification systems, which must learn to predict conceptually diverse label combinations spanning the entire taxonomic structure.

Depth	Count	%
1	106	4.5
2	106	4.5
3	405	17.1
4	**680**	**28.7**
5	554	23.4
6	341	14.4
7	110	4.6
8	36	1.5
9	13	0.5
10	11	0.5
11	5	0.2

Table 4: Distribution of UAT concept depths in AstroConcepts showing annotation preference patterns across taxonomic levels.

4. Experiments

4.1. Experimental Setup

Task Formulation We formulate astrophysics concept classification as a multi-label task where, given an abstract x concatenated with its title, the goal is to predict a subset $Y \subseteq \mathcal{L}$ of relevant UAT concepts from the complete label space $\mathcal{L}$ of 2,367 labels. We use title+abstract concatenation as input text to provide models with maximum available semantic information for concept prediction.

Data Partitioning We split the corpus into train (18,677 abstracts, 85%) and test (3,025 abstracts, 15%) sets using label-aware stratification. Labels appearing $\geq$15 times are stratified to achieve approximately 85/15 distribution per label, while labels with $<$15 occurrences are placed entirely in the training set to maximize training signal.

Evaluation Metrics Following standard practice in multi-label classification (Zhang and Zhou, 2014; Tsoumakas and Katakis, 2010), we evaluate using Macro-F1, which averages per-label F1 scores and is essential for imbalanced settings (Wu et al., 2020). For ranking evaluation, we use Precision/Recall at k ($P@k$, $R@k$) with $k \in \{1, 3, 5\}$, chosen to align with the average number of assigned concepts per paper (4.31, see Table 2).

4.2. Baseline Approaches

We establish baselines across multiple paradigms to understand effective modeling approaches for astrophysics concept classification, organized by increasing complexity: non-parametric methods, supervised neural models, and vocabulary-constrained LLMs.

4.2.1. Non-Parametric Methods

Rule-based Matching We implement lexical matching based on the assumption that if a UAT concept is explicitly mentioned in the text, it should be assigned as a label. The method searches for exact string matches of UAT concept names within the title and abstract. For each concept, we check for its canonical designation and optionally include synonyms and abbreviations from the UAT taxonomy (e.g., searching for both *active galactic nuclei* and *AGN*). We evaluate two variants: Rule-based$_{\text{w/o var}}$ uses only canonical names, while Rule-based$_{\text{w/ var}}$ includes all alternative forms provided in the UAT.

k-Nearest Neighbors This approach assumes abstracts with high contextual similarity should share similar concepts. We encode titles and abstracts using three embedding models: astroBERT (Grezes et al., 2024) (adapted specifically for astrophysics texts), INDUS (Bhattacharjee et al., 2024) (a scientific language model covering astrophysics, earth science, and general physics), and `Qwen3-Embedding-8B`[3] (chosen for strong sentence similarity performance). For each abstract from the test set, we retrieve k nearest training neighbors using cosine similarity, then predict the most frequent labels among them. We perform a grid search over $k \in \{5, 10, 20, 50\}$ and all embedding models, with detailed results provided in the appendix. Table 5 reports only the best-performing configuration (embedding model and k value) for readability.

4.2.2. Supervised Neural Models

To investigate the effects of domain adaptation, we fine-tune three transformer models representing different levels of domain specialization: BERT (Devlin et al., 2019) (general-purpose), SciBERT (Beltagy et al., 2019) (scientific domains), and astroBERT (Grezes et al., 2024). We add classification heads on $[CLS]$ representations and fine-tune end-to-end with max_length=512, batch_size=8, and AdamW optimizer. Due to computational resource constraints, we were unable to include Qwen models in the fine-tuning experiments. We perform a grid search over learning rates $\{2e-5, 3e-5, 5e-5\}$ and epochs $\{1, 2, 3, 5, 8, 10\}$ to find optimal hyperparameter configurations. Detailed results for each configuration are provided in the appendix. Table 5 reports only the best-performing configuration (lr=2e-5, epochs=8) for readability.

4.2.3. Vocabulary-Constrained LLMs

LLMs demonstrate strong capabilities on multi-label text classification tasks (Zhou et al., 2024; Tabatabaei et al., 2025) but face challenges with large label spaces. Prompting an LLM directly to select from all 2,367 UAT concepts is infeasible due to context length limitations and label complexity. Preliminary experiments on a small subset yielded very poor results when using the complete label list, as the model hallucinated concepts or became overwhelmed by the extensive vocabulary. We therefore implement a two-stage approach: (1) use our best supervised model (astroBERT; see Table 5) to generate top-50 candidate labels for each abstract, (2) prompt `DeepSeek-V3-reasoner` (DeepSeek-AI et al., 2024) via its API[4] to select relevant concepts from these candidates. We chose the top-50 threshold based on analysis showing that astroBERT's top-50 predictions covered approximately 82% of ground-truth labels (see Figure 3 in the appendix 9), providing good coverage while maintaining manageable prompt length. We evaluate `DeepSeek-V3-reasoner` using the prompt shown in Figure 2 of the appendix 9. This approach constrains the model to valid UAT terminology while leveraging its semantic understanding to select the most relevant concepts from the candidate set.

5. Results and Analysis

This section presents and analyzes results across all methods, revealing key insights about domain adaptation, frequency effects, and the comparative strengths of different learning paradigms for extreme multi-label scientific classification.

5.1. Overall Performance and Domain Impact

Table 5 presents comprehensive results across all methods, revealing fundamental insights about learning paradigms for extreme multi-label scientific classification.

Our evaluation reveals a progression of insights across methodological paradigms. Simple rule-based matching achieves reasonable precision (0.229) but faces fundamental limitations: string matching of concept mentions may not reflect the core research focus. Common terms like "*photon*"

[3] https://huggingface.co/Qwen/Qwen3-Embedding-8B

[4] https://api-docs.deepseek.com/

Method	Overall			Ranking Precision			Ranking Recall		
	Precision	Recall	F_1	P@1	P@3	P@5	R@1	R@3	R@5
Non-parametric									
Rule-based$_{\text{w/o var}}$	0.2290	0.2130	0.1950	‡	‡	‡	‡	‡	‡
Rule-based$_{\text{w var}}$	0.2170	0.2730	0.2140	‡	‡	‡	‡	‡	‡
k-NN	0.1165	**0.7509**	0.2017	0.6125	0.4607	0.3698	0.1398	0.3155	0.4221
Supervised neural models									
BERT	0.1784	0.2273	0.1880	0.2975	0.2187	0.1784	0.0810	0.1730	0.2273
SciBERT	0.2020	0.2563	0.2127	0.3213	0.2409	0.2020	0.0872	0.1864	0.2563
astroBERT	**0.3068**	0.3905	0.3243	0.4909	0.3733	0.3068	0.1360	0.2933	0.3905
Zero-shot prompting									
Deepseek	0.2891	0.6322	**0.3770**	**0.6502**	**0.4930**	**0.4017**	**0.1815**	**0.3837**	**0.5050**

Table 5: Overall performance on AstroConcepts test set. Best results shown in **bold**. ‡Not applicable for rule-based methods.

appear across diverse papers, while implicit language and incomplete variant coverage constrain effectiveness.

Moving to similarity-based approaches, k-NN achieves exceptional recall (0.751) but poor precision (0.117). Crucially, domain-adapted astroBERT embeddings outperform general models like Qwen, better capturing subtle concept distinctions essential for astrophysics. However, performance degrades beyond $k = 10$ neighbors as noise from irrelevant papers accumulates.

The most striking finding emerges with vocabulary-constrained DeepSeek, which outperforms even the best domain-adapted model (astroBERT) by 16% in F_1 score (0.377 vs 0.324). This demonstrates that domain expertise can be effectively incorporated through vocabulary constraints rather than solely through model parameters. Meanwhile, supervised domain adaptation shows clear value: astroBERT substantially outperforms SciBERT (0.324 vs 0.213) with improvements concentrated in precision and confident prediction rather than comprehensive recall.

Together, these results reveal that effective scientific text classification benefits from hybrid approaches combining general language understanding with structured domain knowledge, opening promising directions for parameter-efficient scientific NLP.

5.2. The Long-Tail Challenge

The extreme label distribution in AstroConcepts (78% of concepts have < 50 examples) enables the systematic analysis of long-tail performance patterns. Table 6 presents frequency-stratified results across Head (> 500 examples), Torso (50–500), and Tail (< 50) concepts, revealing insights that aggregate metrics cannot capture.

Three patterns emerge, providing useful insights into handling extreme imbalance. First, domain adaptation provides asymmetric benefits: astroBERT shows modest Head improvements over SciBERT (0.193 $\rightarrow$ 0.216) but larger relative gains for Tail concepts (0.023 $\rightarrow$ 0.081), though absolute performance remains limited across traditional approaches. Second, all methods exhibit a consistent "torso peak" where mid-frequency concepts achieve optimal performance. This pattern suggests fundamental properties of scientific vocabulary learning, in which models balance sufficient training signal with complexity issues at frequency extremes. Most significantly, the vocabulary-constrained approach achieves superior tail performance (F_1: 0.198 vs 0.081 for astroBERT), demonstrating that domain expertise can be effectively incorporated through structured constraints rather than parameter fine-tuning alone. This hybrid approach, combining general language understanding with domain-specific vocabulary guidance, proves particularly effective for rare concepts. Finally, we introduce frequency robustness (Δ = Head F_1 - Tail F_1) as a critical evaluation dimension. The constrained approach achieves 3× better robustness than SciBERT ($\Delta = 0.045$ vs 0.170), indicating that architectural choices and constraint mechanisms matter more for handling frequency imbalance than domain specialization alone.

6. Discussion

Our systematic evaluation reveals fundamental insights into extreme multi-label classification in scientific domains while establishing new evaluation paradigms that advance understanding beyond existing benchmarks.

Addressing the Research Questions Our findings provide clear answers to the three research questions posed. Regarding RQ1 (handling ex-

Method	F_1			Head		Torso		Tail		
	Head	Torso	Tail	P@3	R@3	P@3	R@3	P@3	R@3	Δ^*
BERT	0.180	0.167	0.021	0.084	0.205	0.168	0.183	0.012	0.024	0.159
SciBERT	0.193	0.197	0.023	0.084	0.211	0.199	0.212	0.015	0.026	0.170
astroBERT	0.216	0.312	0.081	0.092	0.230	0.311	0.333	0.046	0.088	0.135
DeepSeek	**0.243**	**0.376**	**0.198**	**0.107**	**0.266**	**0.417**	**0.443**	**0.135**	**0.267**	**0.045**

Table 6: performance across frequency bins. Head/Torso/Tail thresholds: $> 500/50$–$500/< 50$ training examples. $^*\Delta$ = Head F_1 - Tail F_1 (lower is better). Best results in **bold**.

treme imbalance), vocabulary-constrained LLMs demonstrate superior robustness across frequency bins, achieving 3× better head-tail balance than traditional supervised approaches. For RQ2 (domain adaptation benefits), we find asymmetric improvements that focus on rare, specialized terminology rather than on frequent concepts, challenging assumptions about uniform domain adaptation effects. RQ3 (competitive LLM performance) is answered affirmatively: the hybrid approach combining astroBERT candidate generation with LLM selection achieves competitive results while requiring substantially fewer computational resources than full fine-tuning.

Methodological Impact We establish frequency-stratified evaluation as essential for extreme multi-label assessment and introduce the head-tail gap (Δ) as a robustness metric that reveals performance patterns invisible in aggregate scores. The systematic comparison across paradigms demonstrates that different approaches excel in complementary areas: rule-based methods provide interpretability but limited coverage, k-NN offers high recall with domain-adapted embeddings, supervised models achieve confident predictions, and constrained LLMs balance precision-recall trade-offs effectively.

Broader Implications The torso peak phenomenon, in which all methods perform best on mid-frequency concepts, suggests fundamental limits to learning from extreme imbalance that transcend architectural choices. This finding has immediate implications for resource allocation in scientific NLP: optimization efforts should target the frequency regions where improvement is most feasible, rather than aiming for uniform performance gains across all concepts.

Relationship to Complementary Resources A related effort from Ting et al. (2025) extracted 9,999 concepts from 408,590 astrophysics papers using LLM-based pipelines and clustering over full-text content. The two resources address different but complementary needs. AstroMLab 5 produces a semantically rich, emergent vocabulary optimized for discovery, knowledge graph construction, and temporal analysis at scale. AstroConcepts, by contrast, provides controlled-vocabulary annotations grounded in the UAT, enabling reproducible evaluation of classification methods under extreme label imbalance, a use case for which a fixed label space, train/test split, and expert-assigned labels are essential. Looking forward, the two resources open natural avenues for joint investigation: AstroMLab 5 concepts could serve as additional candidate labels or weak supervision signals for tail concepts in AstroConcepts, while UAT-grounded annotations could provide an extrinsic evaluation signal for the quality of LLM-extracted concept vocabularies.

Limitations and Future Directions Our findings are domain-specific and require validation across other scientific fields before broader claims about scientific NLP can be established. The persistent tail performance challenge (best F_1: 0.198) indicates fundamental limitations in current approaches, suggesting opportunities for architectural innovations tailored to extreme imbalance scenarios. Future work should explore integrating structured domain knowledge with few-shot learning approaches. An important validation step is to audit a stratified sample of the corpus through expert review and inter-annotator agreement analysis, which would quantify annotation noise and its downstream effect on recall-based metrics, particularly for tail concepts where incomplete author-assigned labels may inflate the apparent difficulty. Extending this methodology to other scientific domains is essential to establish whether the torso-peak phenomenon, asymmetric domain adaptation benefits, and the effectiveness of vocabulary-constrained approaches generalize beyond astrophysics. Finally, more extensive experiments with the LLM filtering stage are needed: evaluating DeepSeek over candidate sets generated by BERT and SciBERT, in addition to astroBERT, would isolate the contribution of the candidate generator's quality from the LLM's own re-ranking ability, providing a cleaner assessment of where the performance gains originate.

7. Conclusion

AstroConcepts provides the NLP community with essential resources for investigating extreme multi-label classification in scientific domains, in particular for astrophysics. Through systematic evaluation across traditional, neural, and vocabulary-constrained approaches, we demonstrate three key insights that advance understanding of scientific text classification. The effectiveness of hybrid vocabulary-constrained approaches, in which astroBERT generates candidate labels and DeepSeek selects from this constrained set, demonstrates that domain expertise can be incorporated through structured vocabulary guidance rather than through extensive LLM fine-tuning. This approach achieves competitive performance (F_1: 0.377) while requiring only inference costs for the domain model and API calls for the LLM, opening promising directions for cost-effective scientific NLP that combines specialized knowledge extraction with general language understanding capabilities. Domain adaptation benefits concentrate asymmetrically on rare, specialized terminology, suggesting that specialized models primarily handle concepts beyond general model capabilities rather than improving performance uniformly. Our frequency-stratified evaluation framework, combined with robustness metrics, provides useful methods for assessing extreme multi-label systems in which aggregate scores can mask critical performance patterns. These contributions address a critical methodological gap by enabling systematic evaluation of extreme imbalance while providing actionable insights for scientific NLP practitioners. The corpus, baselines, and evaluation framework lay the foundations for future research on specialized domain classification, while our findings on vocabulary-constrained approaches indicate promising directions for resource-efficient scientific text processing systems. To facilitate future research, we make the AstroConcepts corpus publicly available. The persistent challenges in tail performance underscore opportunities for novel approaches that integrate structured knowledge with text-based classification. Future work should prioritize annotation validation through expert audits and inter-annotator agreement studies, systematic cross-domain replication, and controlled ablations of the LLM re-ranking stage across candidate generators of varying quality. As the scientific literature continues to expand and specialized terminology increases, effective handling of extreme imbalance becomes essential for scientific NLP applications.

8. Ethical Considerations

All abstracts are from published scientific papers that are publicly accessible. We include only bibliographic metadata (bibcode, title, abstract, publication year, journal) and assigned UAT concepts.

9. Bibliographical References

A. Accomazzi, N. Gray, C. Erdmann, C. Biemesderfer, K. Frey, and J. Soles. 2014. The Unified Astronomy Thesaurus. In *Astronomical Data Analysis Software and Systems XXIII*, volume 485 of *Astronomical Society of the Pacific Conference Series*, page 461.

Alberto Accomazzi, Michael J. Kurtz, Edwin A. Henneken, Roman Chyla, James Luker, Carolyn S. Grant, Donna M. Thompson, Alexandra Holachek, Rahul Dave, and Stephen S. Murray. 2015. Ads: The next generation search platform.

Jennifer Bartlett, Mugdha Polimera, Kelly Lockhart, Alberto Accomazzi, Michael Kurtz, and Science Explorer Team. 2025. ADS and SciX: Pioneering the Next Generation of Interdisciplinary Research Discovery. In *American Astronomical Society Meeting Abstracts #245*, volume 245 of *American Astronomical Society Meeting Abstracts*, page 442.04.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Bishwaranjan Bhattacharjee, Aashka Trivedi, Masayasu Muraoka, Muthukumaran Ramasubramanian, Takuma Udagawa, Iksha Gurung, Nishan Pantha, Rong Zhang, Bharath Dandala, Rahul Ramachandran, Manil Maskey, Kaylin Bugbee, Mike Little, Elizabeth Fancher, Irina Gerasimov, Armin Mehrabian, Lauren Sanders, Sylvain Costes, Sergi Blanco-Cuaresma, Kelly Lockhart, Thomas Allen, Felix Grezes, Megan Ansdell, Alberto Accomazzi, Yousef El-Kurdi, Davis Wertheimer, Birgit Pfitzmann, Cesar Berrospi Ramis, Michele Dolfi, Rafael Teixeira de Lima, Panagiotis Vagenas, S. Karthik Mukkavilli, Peter Staar, Sanaz Vahidinia, Ryan McGranaghan, and Tsendgar Lee. 2024. INDUS: Effective and Efficient Language Models for Scientific Applications. *arXiv e-prints*, page arXiv:2405.10725.

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

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. DeepSeek-V3 Technical Report. *arXiv e-prints*, page arXiv:2412.19437.

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.

C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. 1998. Citeseer: an automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*, DL '98, page 89–98, New York, NY, USA. Association for Computing Machinery.

F. Grezes, S. Blanco-Cuaresma, A. Accomazzi, M. J. Kurtz, G. Shapurian, E. Henneken, C. S. Grant, D. M. Thompson, R. Chyla, S. McDonald, T. W. Hostetler, M. R. Templeton, K. E. Lockhart, N. Martinovic, S. Chen, C. Tanner, and P. Protopapas. 2024. Building astroBERT, a Language Model for Astronomy & Astrophysics. In *Astronomical Data Analysis Software and Systems XXXI*, volume 535 of *Astronomical Society of the Pacific Conference Series*, page 119.

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8153–8161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, K. Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371.

David Lewis. 1987. Reuters-21578 Text Categorization Collection. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52G6M.

Rundong Liu, Wenhan Liang, Weijun Luo, Yuxiang Song, He Zhang, Ruohua Xu, Yunfeng Li, and Ming Liu. 2023. Recent advances in hierarchical multi-label text classification: A survey.

Eneldo Loza Mencía and Johannes Fürnkranz. 2010. *Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain*, pages 192–215. Springer Berlin Heidelberg, Berlin, Heidelberg.

Andrew McCallum, Kamal Nigam, Jason D. M. Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163.

Alistair Miles and Sean Bechhofer. 2009. SKOS Simple Knowledge Organization System Reference. W3c recommendation, W3C.

Mobashir Sadat and Cornelia Caragea. 2022. Hierarchical multi-label classification of scientific documents. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8923–8937, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

António Paulo Santos and Fátima Rodrigues. 2009. Multi-label hierarchical text classification using the acm taxonomy.

Seyed Amin Tabatabaei, Sarah Fancher, Michael Parsons, and Arian Askari. 2025. Can large language models serve as effective classifiers for hierarchical multi-label classification of scientific documents at industrial scale? In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 163–174, Abu Dhabi, UAE. Association for Computational Linguistics.

Yuan-Sen Ting, Alberto Accomazzi, Tirthankar Ghosal, Tuan Dung Nguyen, Rui Pan, Zechang Sun, and Tijmen de Haan. 2025. AstroMLab 5: Structured summaries and concept extraction for 400,000 astrophysics papers. In *Proceedings of the Third Workshop for Artificial Intelligence for Scientific Publications*, pages 170–185, Mumbai, India and virtual. Association for Computational Linguistics.

Autumn Toney and James Dunham. 2022. Multi-label classification of scientific research documents across domains and languages. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 105–114, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Grigorios Tsoumakas and Ioannis Katakis. 2010. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*.

Ximing Wu, Hao Chen, and Qinmin Zhang. 2020. Revisiting macro-f1 score for imbalanced multi-label classification. *arXiv preprint*.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE TKDE*.

Chuang Zhou, Junnan Dong, Xiao Huang, Zirui Liu, Kaixiong Zhou, and Zhaozhuo Xu. 2024. QUEST: Efficient extreme multi-label text classification with large language models on commodity hardware. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3929–3940, Miami, Florida, USA. Association for Computational Linguistics.

A.1. Complete Prompt Template (Section 4.2.3)

Figure 2 shows the designed prompt as part of our experiments.

A.2. astroBERT label coverage (Section 4.2.3)

Figure 3 shows that astroBERT's top-50 predicted labels covers approximately 82% of the ground-truth labels.

```
You are an expert astrophysicist and scientific topic classifier.
Your task is to choose between 1 to 10 labels from the candidate list that
accurately describe the main scientific themes of the following research
paper.
A label should be selected only if it is clearly relevant to the paper's
content.

--
Abstract:
{abstract}

--
Candidate topics suggested by the model:
{topk_labels}

Return your answer in valid JSON as follows:
{
  "selected_labels":  ["label1", "label2", ...]
}
Do not include explanations or text outside the JSON.
```

Figure 2: Prompt template used for vocabulary-constrained LLM classification.

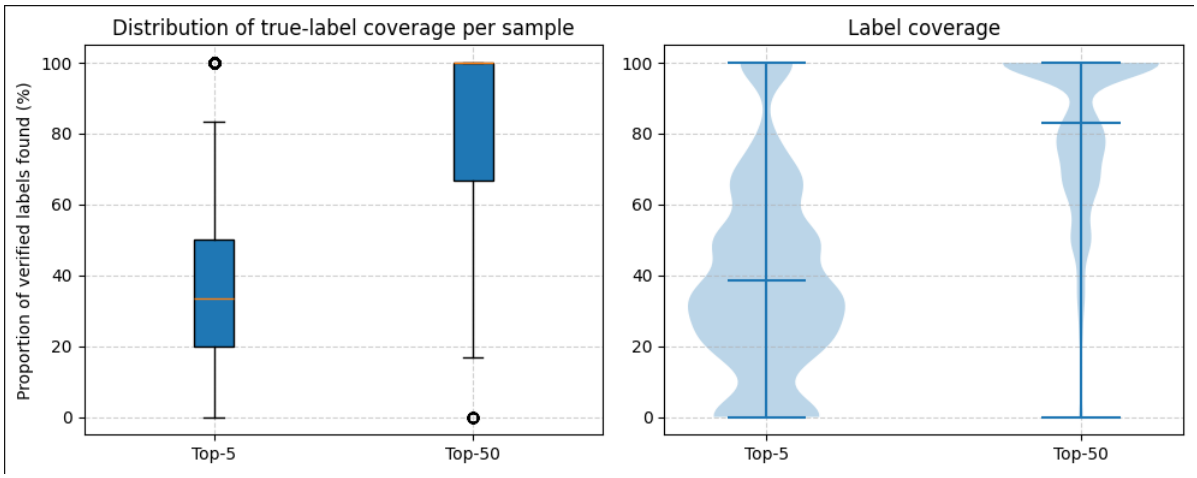


Figure 3: Fine-tuned astroBERT Label Coverage

Benchmarking LLMs for ARR Area Assignment: Evidence and Implications for Assignment Strategies

Eileen Bingert, Diego Alves, Stefania Degaetano-Ortlieb
Saarland University, Department of Language Science and Technology
Campus A2.2, 66123 Saarbruecken
s8eibing@teams.uni-saarland.de, diego.alves@uni-saarland.de, s.degaetano@mx.uni-saarland.de

Abstract

We study how large language models (LLMs) perform at assigning review areas, specifically for the Association of Computational Linguistics (ACL) Rolling Reviews (ARR), using titles and abstracts from 2020–2025. We compare multiple LLMs and prompting schemes (zero/few-shot; with/without ARR keywords; each-category variants) and analyze per-area scores, error overlap, and confusion matrices. One-shot prompting (with OpenAI-gpt-oss-20b) tends to perform best, while injecting ARR keywords often lowers accuracy. Task-bounded areas (e.g., MT, IE, QA, Summarization) are predicted more reliably, whereas broad, cross-cutting labels (e.g., Resources and Evaluation, NLP Applications) are frequently conflated, indicating taxonomy ambiguity rather than solely model limitations. We recommend hierarchical or primary-plus-secondary labels to work towards reducing ambiguity and improving reviewer matching in the future. Our dataset, methods, and findings offer a reproducible baseline for area selection support in conference workflows such as for ACL.

Keywords: LLM benchmarking, Zero-/few-shot prompting evaluation, ACL area classification

1. Introduction

A known problem in the academic community for reviewer assignment is to generate a system of areas to match reviewer expertise. As this is not a trivial task, given that the choice directly affects reviewer assignment to papers, keywords have been established to help authors select the right area for their papers. In this paper, we run a pilot test to assess whether the application of large language models (LLMs) might indicate insights on area assignment and its development over time. For this, we run a case study on papers from the Association for Computational Linguistics (ACL), a long-running, richly studied scholarly ecosystem. Quite some endeavours have been going into updating keywords, areas, and tracks in order to match the ever changing landscape of ACL[1] and related conferences[2] (Rogers et al., 2023). Since 2021, the ACL Rolling Review (ARR) introduced a common, evolving *area taxonomy* with public area descriptions and keywords to standardize reviewer matching across ACL-family conferences. Statistics (see Rogers et al. (2023) Table 1) show that while some areas are relatively specific and rather small, areas with less topical focus such as *Resources and Evaluation* or *NLP Applications* act as collective bins for various topics, making reviewer assignment more problematic. Thus, area labels can be *coarse* and partially overlapping, creating ambiguity for authors and area chairs. Also, area choice interacts with automated affinity matching (such as the Toronto Paper Matching System (TPMS)) and chair policies, so misalignment between an article's content and its declared area can influence both the reviewer pool and evaluation outcomes. Moreover, human-curated keywords may incompletely capture area semantics and can again overlap across areas (see arr (2025b,a); emn (2025), e.g., *evaluation* as a keyword across areas), potentially biasing both manual choice and automated matching.

In this paper, we ask whether given paper titles and abstracts, LLMs would classify papers to the chosen ACL areas. Prompted LLMs could provide a complementary signal: given the same title and abstract string available at submission time, they operationalize a content-driven mapping into the ARR taxonomy. By benchmarking different prompting schemes against author-declared areas, we can (i) estimate attainable agreement from content alone, and (ii) identify systematic confusions that may indicate where areas are semantically conflated or keywords mis-specify scope. For this, we use the main conference titles and abstracts from the ACL, EACL and NAACL venues from 2020 up to 2025 with 558 papers in total. We compare the results of different LLMs given also differing prompting strategies (e.g., with and without keywords). Results show how general areas, especially, NLP applications, encompasses papers, which would well fit other areas, as well as how areas might actually be closely related and could be conflated.

[1] https://aclrollingreview.org/areas
[2] https://2025.emnlp.org/track-changes/

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 13–24
12 May 2026.

2. Theoretical Background

Our study connects to two strands of prior work: (i) the use of LLMs to support topic selection and document triage in scientific workflows and (ii) prompt-based control of LLMs for classification.

2.1. LLMs for Topic Selection and Scholarly Triage

LLMs are increasingly used as high-recall screeners and labelers over titles and abstracts in systematic reviews and literature audits, offering substantial efficiency gains while maintaining competitive recall when prompts and consensus protocols are carefully designed (Dennstädt et al., 2024; Joos et al., 2024). Beyond screening, LLMs have been shown to support structured topic and label assignment: Kuzman and Ljubešić (2025) propose a teacher–student framework for news topic classification in under-resourced languages, demonstrating that LLM-generated labels can effectively train downstream classifiers within the IPTC hierarchical taxonomy (e.g., politics, economy, society). Similarly, Zhu et al. (2025) use LLM-derived representations for hierarchical taxonomy induction and scientific paper clustering, achieving state-of-the-art taxonomy coherence and interpretability. Together, these findings support the feasibility of LLM-driven constrained topic assignment in structured label spaces.

In the area of NLP, Ahmad et al. (2024a) created a hierarchical taxonomy and manually annotated 1500 ACL Anthology publications with their main contributions using CL/NLP topics and sub-topics. They then compare two different approaches for classifying publications with a fine-tuned SciNCL model (Ahmad et al., 2024b). The weakly supervised X-transformer model outperforms the competing approach as well as the baseline, reaching a micro precision of 0.4391. This suggests that models are able to correctly assign labels in an extreme multi-label classification task focusing on the area of NLP and the ACL anthology while also highlighting the difficulty of such a task with limited training data.

In scenarios with no or very limited data for fine-tuning, generative LLMs offer an accessible alternative for label generation and classification. Kuzman et al. (2023) report promising results using GPT-3.5 for genre identification, while Säily et al. (2025) evaluate newer GPT models for generating genre labels for novels in the Corpus of Historical American English. Focusing on scientific texts in contemporary English and German, the second LLMs4Subjects shared task (D'souza et al., 2025) included a subtask on automatic domain identification, requiring systems to assign one or more labels from 28 predefined categories (Ho, 2025; Shirali et al., 2025).

2.2. Prompting Strategies

For improving the output of LLMs, prompt engineering has proven to be an effective tool. The modification of the input prompt is a relatively simple way in comparison to other techniques used to increase the performance as it does not require the retraining or fine-tuning of models. Therefore, it is often accessible to users without coding abilities as the prompts are written in natural language (Sahoo et al., 2024). Thus, prompt engineering has become a primary lever for steering LLM behavior without parameter updates, with established families including zero-shot, few-shot, instruction-style prompting, and variants that manipulate format, label space exposure, and task decomposition (Schulhoff et al., 2024; Sahoo et al., 2024). A prompt template typically consists of some context or input and the desired output. This template is then repeated n-times with $n>0$ for few-shot prompting and $n=0$ for zero-shot prompting (Brown et al., 2020). Zero-shot prompting solely contains a description of the task with no additional exemplary input data. Therefore, the LLM relies only on pre-existing knowledge and understanding to perform the required action (Sahoo et al., 2024). Few-shot prompting operationalizes *in-context learning* (ICL), where models condition on a small set of input–output exemplars (also called demonstrations) to induce a task-specific mapping at inference time (Brown et al., 2020). Subsequent work has refined our understanding of why demonstrations help: accurate labels are not always necessary, while exposure to the label inventory, input distribution, and output format often accounts for much of the ICL gains (Min et al., 2022). These findings motivate our controlled variants that (a) vary the number and coverage of demonstrations across ARR areas and (b) expose or hide area keywords to test whether lexical cues aid or distract models during area selection.

One disadvantage of few-shot prompting is the potential inclusion of biases in the input prompt, which then are reflected in the output of the LLM. Therefore, the wording of the input can have significant influence on the model's behaviour. Another is the lengthening of the input prompt that can occur with multiple examples, which then can exceed the input token limit or increase the runtime (Sahoo et al., 2024). We try to account for this by considering varying numbers of examples provided in the prompt, which clearly affect the length of the prompt.

For taxonomy-based classification, two design choices matter. First, demonstrations should represent the *label space* broadly enough to reduce class-imbalance artifacts in ICL (Min et al.,

2022). Second, prompt content can introduce *shortcut features* (e.g., area-specific keywords) that improve surface matching but may reduce robustness when areas overlap semantically; our "with/without keywords" comparison explicitly probes this trade-off (Schulhoff et al., 2024; Sahoo et al., 2024).

3. Data

3.1. Data Compilation

To conduct our experiments, we compile a corpus in English of the relevant data of the ACL, EACL, NAACL main conferences from 2020 to 2025 using the ACL Anthology[3].

	2023	2024	2025	Total
ACL	29	289	5	323
EACL	3	49	0	52
NAACL	0	180	3	183
Total	32	518	8	558

Table 1: Number of papers in the corpus per year and conference that were successfully matched to an ARR area.

In a first step, we download the proceedings. These HTML files encompass all papers from the proceedings for a single event. We only focus on the main conference given that it is the one operating strictly on the ARR area classification for reviewer assignment. We use a Python script to extract the relevant papers (excluding e.g. workshop proceedings).

As the goal of our research is to measure the performance of different LLMs in assigning the different ARR areas, we require a second file with the officially assigned areas as a baseline for comparison. Therefore, in a second step, we download the data of the category *Anonymous Pre-prints* from Open Review Rolling Reviews[4] for the years from 2021 to 2025. For 2020, there is no data available. The areas are chosen by the authors during the submission process. We use a second Python script that compares the titles of the data downloaded in the first step to the titles in these files. We repeat this step for three years for each event/year combination: The year of the conference, as well as the two years before that. For instance, the data for ACL 2024 is compared to the Open Review Rolling Reviews data for 2022, 2023 and 2024. If a match occurs, the area is added to the other metadata (See Table 2) and saved in a separate file. In total, our data set contains 558 abstracts successfully matched to an area[5]

Due to incomplete availability of OpenReview metadata and imperfect title matching, only 558 papers (out of 8,905 papers published, 6%) could be successfully aligned with ARR area labels. This constitutes a limitation of our dataset and likely under-represents the full set of conference papers. Consequently, our study should be viewed as a preliminary investigation into the feasibility and performance of LLM-based ARR area assignment, rather than a definitive large-scale evaluation.

Table 1 reports the number of papers successfully matched to an ARR area per conference and year. As shown, there is a large discrepancy in the number of papers across years: ACL 2024 accounts for 518 papers, while 2023 and 2025 only have 32 and 8 papers, respectively. This difference is primarily due to the inhomogeneous availability of ARR metadata in OpenReview for different years.

3.2. ARR Areas

As stated above, we successfully assigned an ARR area[6] to 558 papers from our data set. The first list of areas and keywords was published in 2023 and then updated in 2024 and 2025. Thus, our data includes 23 areas from different versions, which we combine into a single list. Additionally, we also merge two areas, the two ACL '23 areas *Semantics: Lexical* and *Semantics: Sentence-level Semantics, Textual Inference, and Other Areas* under the new label *Semantics*, as the 2025 version contains a single area with this thematic focus. The keywords were selected by the senior area chairs of each area, based on their knowledge of the area. The ARR states that the list is not exhaustive, but an overview of the most common themes.

Two of the 25 ACL areas we chose to include, namely *Human-centered NLP* and *Language Modeling*, are not included in our data set, likely because they were only present in 2024. We still chose to include the areas in nine of our ten prompting strategies as one minor goal of the evaluation task is to test the accuracy of the tag chosen by the respective paper's authors. Papers that are assigned the area *Special Theme Track* are excluded from the data, as this label is not transparent and changes for each conference. Therefore, a thematic classification is not possible and no keywords are available.

The areas vary in size, the smallest areas *Syntax: Tagging, Chunking and Parsing* and *Phonology, Morphology and Word Segmentation* consist-

[3] https://aclanthology.org/

[4] https://openreview.net/group?id=aclweb.org/ACL/ARR

[5] https://github.com/eilebin/acl-task-data

[6] https://aclrollingreview.org/areas

Category	Example
ID	Borenstein_Arora_Kaffee_Augenstein_NAACL_2025
Title	Investigating Human Values in Online Communities
Authors	Nadav Borenstein, Arnav Arora, Lucie-Aimée Kaffee, Isabelle Augenstein
Year	2025
Event	naacl
Type	long
Abstract	Studying human values is instrumental for cross-cultural research [...]
Area	Computational Social Science and Cultural Analytics

Table 2: Example of the extracted data for each paper. The ID is in a BibTeX-like format, consisting of a maximum of five authors, the event, and year. Its purpose is the identification of the paper.

ing of 3 papers, whereas the biggest area *Resources and Evaluation* encompasses 68 papers. Figure 1 displays the number of papers in each area.

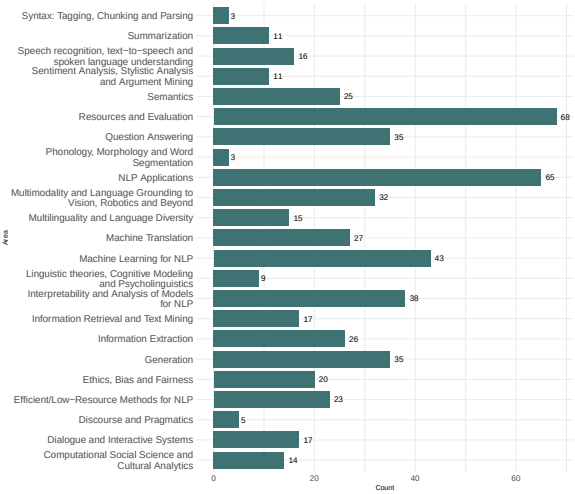


Figure 1: Number of papers across the different ARR areas in the data set.

4. Methodology

4.1. Models

We evaluate the performance of three different models on our dataset. Given that our data set is limited in size, we chose to focus on existing models rather than train an encoder model.

Hermes 2 Pro - Llama-3 8B Hermes 2 Pro - Llama-3 8B[7] (Teknium et al., 2024) was released as an upgrade of Nous Hermes 2, based on Llama-3 (8 billion parameters).

Qwen3-32B Qwen3-32B (Qwen Team, 2025). The model[8] is a 32.8B-parameter causal language model with 64 layers and grouped-query attention (64 query heads and 8 key-value heads). It offers both a thinking and a non-thinking mode, which we indicate by the addition of _Think.

OpenAI-gpt-oss-20b OpenAI-gpt-oss-20b[9] (OpenAI, 2025) is an autoregressive Mixture-of-Experts transformer with 24 layers and 20.91B parameters, released on 5 August 2025. OpenAI-gpt-oss-20b was trained on a text-based data set, focusing on science, code and general world knowledge.

All models were run locally using the Hugging Face Transformers library with automatic GPU device placement. Inference was conducted in bfloat16 precision without additional quantization. Prompts were formatted using the models' native chat templates, and deterministic decoding was used to ensure reproducibility. No sampling-based hyperparameters were added. The generated outputs were saved as YAML files without further post-processing.

4.2. Prompting Strategies

As stated in Section 2, we test different prompting techniques on the models, both with and without human-curated keywords. In all cases, the prompt includes a description of the task and the structure of the input data.

- **zero-shot:** For this, we offer the model an example output with no additional material.
  Example Output (in YAML format):
  article_id:
  acl_area:
- **one shot, three shot, five shot**: This prompt encompasses one (three, five) randomly chosen examples in an input and the required output format. An example consists of the data in

[7] https://huggingface.co/NousResearch/Hermes-2-Pro-Llama-3-8B

[8] https://huggingface.co/Qwen/Qwen3-32B

[9] https://huggingface.co/openai/gpt-oss-20b

Table 2 as the Example Input, as well as the article_id and the area in the Example Output.

- **each category:** In this prompt, we include one example for each category that is included in our curated list of ARR areas.
- **each category_v2:** Here, we only include the ARR areas that are represented in the data in the list of possible areas the models can choose from. Therefore, the number of areas is reduced from 25 to 23, as the areas *Human-centered NLP* and *Language Modeling* are not included in our data set.
- **each category_three:** In this prompt, we include three examples for each area in the data. However, due to the necessity to reduce the length of the prompt, we only give the example output.
- **keywords_zero shot, keywords_one shot, keywords_three shot, keywords_five shot:** In addition to the previous information, the keywords provided by the ARR are added to the prompt.

In Appendix A, we provide the prompt used in the one-shot experiment to illustrate our prompting strategy.

5. Evaluation of Different Prompting Strategies

5.1. Zero-Shot vs. Few-Shot Prompting

As Figure 2 portrays, the accuracy of 2 of the 4 models, i.e. Hermes2Pro and the non-thinking version of Qwen3-32B, increases with a higher number of examples in the prompt.

Considering Table 3, Hermes2Pro gradually increases its performance with the inclusion of more examples into the prompt. Initially, at zero-shot prompting, it classifies 174 papers correctly. It then peaks at five-shot with 196 papers, a 12.64 % increase. The non-thinking version of Qwen3-32B mirrors this behaviour. At zero-shot, the label for 224 papers is correctly selected. This value expands by 14.73 % to 257 at five-shot.

However, for OpenAI-gpt-oss-20b offering more examples does not improve the performance of the LLM. This model peaks at one-shot prompting with 309 correctly classified papers, the highest value in our data, and then declines as more examples are added. At five-shot prompting, 299 areas could be correctly assigned to the papers, a 3.24 % decrease. Similarly, Qwen3-32B_Think reaches its best output performance with only one example in the prompt at 284 papers.

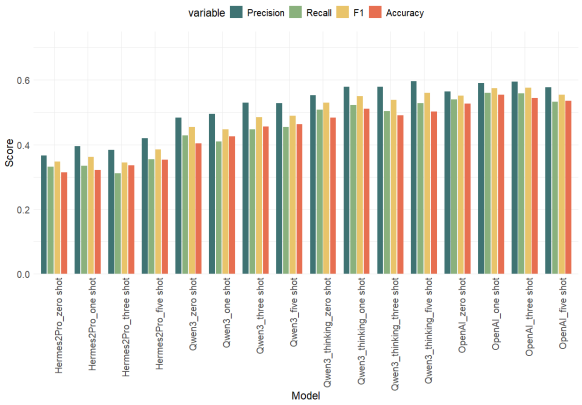


Figure 2: Performance of the four models in terms of macro-averaged precision, recall, F1-score, and accuracy across different prompting strategies without keywords.

The highest recall (0.560) is achieved by OpenAI-gpt-oss-20b at one-shot prompting, reflecting the high accuracy of this specific prompting and model combination. In contrast, Hermes2Pro at three-shot has the lowest recall with 0.312. Notably, the model and few-shot combination with the highest accuracy does not reach the highest macro-averaged precision. Qwen3-32B_Think reaches a precision of 0.597 at five-shot, whereas the best performing OpenAI-gpt-oss-20b achieves a precision of 0.5899 at one-shot.

5.2. All Area Coverage

For the previous experimental setup, we only offered a fixed number of examples in the prompt in total, so not every area was represented. In this step, we increase the number of examples significantly to 23 and 69 respectively, covering all of the areas in the data set.

Considering again Table 3, three of the four models do not display an increase in correctly assigned areas for the each-category prompt. Only the non-thinking version of Qwen3-32B performs better than at zero-shot and few-shot with 271 papers. Compared to five-shot, the best performing experimental set-up for our initial testing, the model increases its correct output by 5.45 %.

Additionally, the exclusion of the two supplementary areas (*Human-centered NLP* and *Language Modeling*) has a positive impact on the performance of all models. The non-thinking version of the Qwen3 model again outperforms its previous results with 276 papers.

As with the previous set-up, OpenAI-gpt-oss-20b does not achieve a rise in correct answers as more examples are added to the prompt. For each category_three, the model even reaches the second-lowest result in total, with 293 papers. This

	Hermes2Pro	Qwen3	Qwen3_Think	OpenAI
Prompts without keywords				
zero-shot	174 (31.18 %)	224 (40.14 %)	269 (48.21 %)	294 (52.69 %)
one-shot	178 (31.90 %)	236 (42.29 %)	284 (50.90 %)	**309 (55.38 %)**
three-shot	186 (33.33 %)	253 (45.34 %)	273 (48.92 %)	304 (54.45 %)
five-shot	196 (35.13 %)	257 (46.06 %)	279 (50.00 %)	299 (53.58 %)
each category	186 (33.33 %)	271 (48.57 %)	277 (49.64 %)	291 (52.15 %)
each category_v2	193 (34.59 %)	276 (49.46 %)	280 (50.18 %)	300 (53.76 %)
each category_three	226 (40.50 %)	293 (52.51 %)	**306 (54.84 %)**	293 (52.51 %)
Prompts with keywords				
keywords_zero-shot	205 (36.74 %)	219 (39.25 %)	239 (42.83 %)	238 (42.65 %)
keywords_one-shot	169 (30.29 %)	223 (39.96 %)	251 (44.98 %)	265 (47.49 %)
keywords_three-shot	187 (33.51 %)	230 (41.22 %)	260 (46.59 %)	271 (48.57 %)
keywords_five-shot	196 (35.13 %)	258 (46.24 %)	283 (50.72 %)	**292 (52.33 %)**

Table 3: Model Performance Comparison

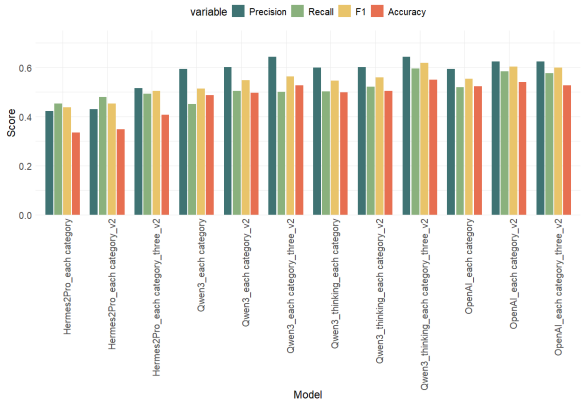


Figure 3: Performance of the four models in terms of macro-averaged precision, recall, F1-score, and accuracy across different prompting strategies with coverage of all areas (each category)

drop is again contrary to the behaviour of the other models that show a positive correlation between the number of examples in the prompt and the performance. Qwen3_Think correctly classifies 306 papers, a 7.75% rise compared to one-shot. Hermes 2 Pro - Llama-3 8B reaches a surge of 15.31% from 196 at five-shot to 226.

In total, the reduction of class-imbalance in the prompt seems to only aid the models if multiple examples for each area are added, as seen in Figure 3. The data for three out of four LLMs suggests that ICL improves with a higher number of examples.

5.3. Prompts with keywords

The inclusion of keywords into the prompt decreases the performance for three of the four models for zero shot, one shot and three shot prompting (see again Table 3). For all of the four variants of shots, the biggest drop can be seen for OpenAI-gpt-oss-20-b model with a decrease of 19.05% for zero shot, 14.24% for one shot, 10.86% for three shot and 2.34% for five shot. Only the Hermes2Pro model reaches better results in two cases, for zero shot and three shot. Additionally, both versions of the Qwen3 model display a better performance for the five shot prompts.

When adding keywords, as presented in Figure 4, the Qwen3 and OpenAI models increase their performance with more examples, a behaviour which is contrary to the results of the experimental setup without keywords for OpenAI-gpt-oss-20b and Qwen3-32B_Think (see Section 5.1). For instance, Qwen3-32B_Think correctly selects the areas of 239 papers for zero-shot prompting. For five-shot its accuracy rises by 18.41% to 283 papers, with no drop for one-shot and three-shot.

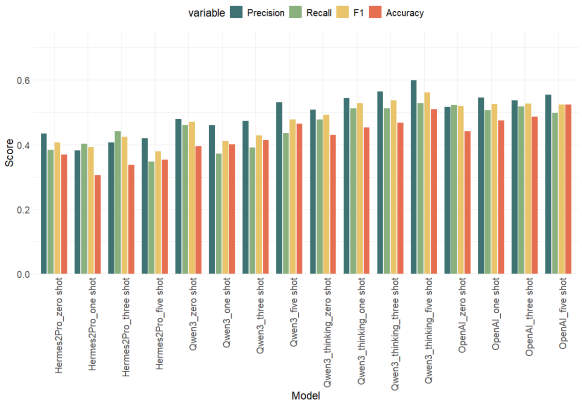


Figure 4: Performance of the four models in terms of macro-averaged precision, recall, F1-score, and accuracy across different prompting strategies with human-curated keywords.

Despite the lower performance in general,

OpenAI-gpt-oss-20b outperforms the other models for one shot, three shot and five shot. With 292 out of 558 (52.33%) correctly classified papers, it reaches the best results for this experimental setup at five shot prompting. For zero-shot, the Qwen 3-Thinking model displays a better performance, with a one paper difference.

In total, the inclusion of keywords seems to not aid the models in the area selection task. This could be due to the human-curated nature of the keywords which might not accurately and fully represent the respective areas. Some keywords are included in multiple areas. For instance, the keyword ”human-in-the-loop” is used for the areas *Dialogue and Interactive Systems* and *Human-Centered NLP*. Additionally, for the area *Linguistic Theories, Cognitive Modeling, and Psycholinguistics* the only keywords are the three parts of the area name.

When comparing our results with previous work, we observe that state-of-the-art performance varies considerably depending on the taxonomy and language. Results reported for the LLMs4Subjects shared task (Ho, 2025; Shirali et al., 2025) are relatively lower than those obtained for our specific task, while Kuzman et al. (2023) report F1 scores above 70 for under-resourced languages. Our task is particularly challenging, as it involves identifying fine-grained research areas within a single scientific field, where substantial inter-topic overlap is likely. Furthermore, the labels are derived from authors' self-assessments, which may introduce subjective bias and additional noise into the training and evaluation data.

6. Evaluation of Results per Area

To assess the performance differences between the areas, we now focus on a micro-level analysis of the results for OpenAI-gpt-oss-20b at one-shot prompting, the best performing model and prompting technique combination in our data (see Figure 5). Additionally, we examine a confusion matrix of the results to trace the discrepancy between declared and assigned areas (see Figure 6).

Figure 5 illustrates the results of the selection task divided by area. Notably, for two areas, *Computational Social Science and Cultural Analytics* and *Phonology, Morphology and Word Segmentation*, the model reaches a perfect precision score. However, for the latter category, this result is only based on one successfully assigned paper. For both areas, this is not reflected in the recall score. The lowest precision scores are noted for the areas *Syntax: Tagging, Chunking and Parsing* and *Semantics*. Similarly, the recall for *Semantics* is also the lowest, along with *NLP Application*. As Figure 6 shows, a higher number of papers from *NLP Applications* were marked as *Generation* instead of the correct area. Seven papers from *Semantics* were categorized as *Machine Learning for NLP*, while four were given the correct area. The highest recall is reached for *Machine Translation* and *Summarization*. *Machine Translation* also has the highest F1-score.

The confusion matrix gives us a more detailed breakdown. Task-bounded areas show a crisp diagonal (e.g., machine translation, information extraction, question answering, summarization), indicating balanced precision and recall. By contrast, broad or cross-cutting areas behave as catch-alls and create asymmetric errors. *Resources and Evaluation* and *Language Modeling* attract many false positives (papers from other areas predicted as these), lowering precision, while simultaneously losing in-area papers to neighbouring labels (lower recall). We also see predictable adjacency confusions: a) *Generation* with *Summarization* with *Dialogue/QA*, b) *MT* with *Multilinguality*, c) *IR/text mining* with *Information Extraction*, d) *Semantics* with *Discourse/Pragmatics*, reflecting shared methods and datasets. Taken together, these patterns indicate that the current area set overaggregates meta roles and underspecifies boundaries for broad scopes; as a result, precision and recall degrade exactly where labels are vague or semantically similar in terms of topics covered. A hierarchical or primary plus secondary labeling (task/domain as primary; role/method such as “resources” “ethics” or “human-centered” as secondary) could be a way to reduce these systematic confusions and aid authors to select areas in a more systematic way.

7. Evaluation of Incorrectly Classified Data

Not only the analysis of the correctly classified papers offers insights, but also the incorrectly classified data points. In some instances, the models select the same area for one paper that does, however, not correspond to the one listed on Open Review.

For this, we filtered out the correctly classified papers and then compared the output of the models. As shown in Table 4, the *each category* prompting offers the most overlapping falsely classified papers, whereas *one-shot prompting* provides the least data points. This classification, therefore, is largely influenced by the performance of OpenAI-gpt-oss-20b.

Due to its low performance in the classification task, we filtered out the Hermes2Pro model for a second comparison, focusing on the two versions of the Qwen3-32B model and OpenAI-gpt-oss-20b.

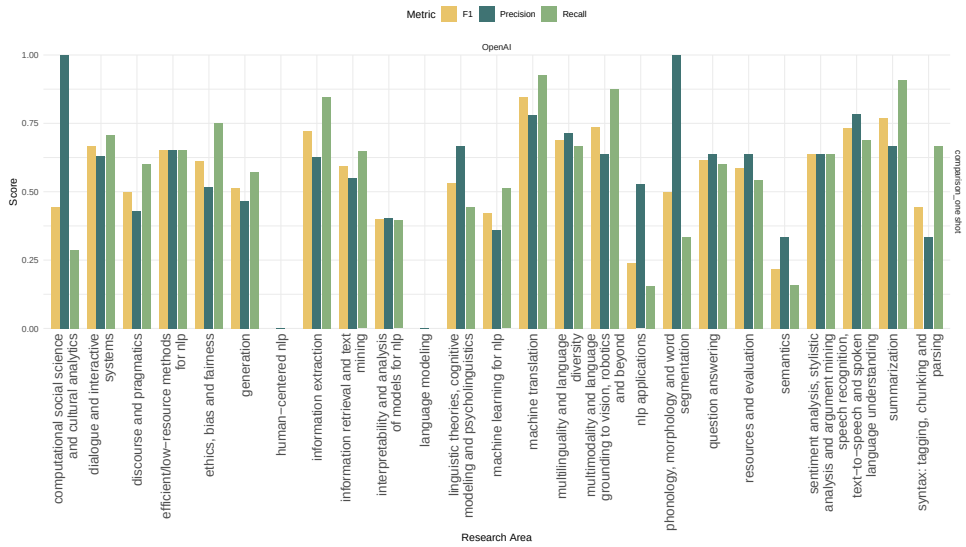


Figure 5: Performance of OpenAI-gpt-oss-20b in terms of precision, recall, and F1-score across the different areas at one-shot prompting.

The assigned label is the same for the three models in 114 cases for *each category*. This is equal to 20.43 % of the data set, the highest result of the five.

	All models	Qwen3–OpenAI
zero shot	51	96
one shot	50	91
three shot	59	101
five shot	63	104
each category	66	114

Table 4: Incorrectly classified but overlapping data

This comparison shows that the areas selected by the authors for the paper might not encompass the content correctly. This could be due to the difficulty of choosing an appropriate area. Table 4 illustrates an example of such a paper. The authors chose the broader category ”Resources and Evaluation”, whereas all LLMs selected the area ”Semantics”. OpenAI-gpt-oss-20b provides a reasoning for its selection, in which it is stated that ”the paper is about noun-noun compounds interpretation, semantic relations, linguistic study” and that ”the focus is semantics”. In this case, the area chosen by the LLMs might represent the content more accurately than the label chosen by the authors as might be inferred from the abstract (See Appendix B) of the paper which indicates a strong focus on semantics rather than providing a new resource.

It could also be interpreted as an indication that the areas offered by the ACL are not transparent, neither to the authors using the ARR area keywords to select an area or in general. The overlap of keywords across different areas might increase this effectas it becomes harder to distinguish the different areas if they encompass similar subtopics.

Article ID:	Rambelli_Chersoni...
Authors:	Resources and Evaluation
LLMs:	Semantics

Table 5: Example of a paper assigned a different area by all four LLMs than the one chosen by the authors. Metadata and abstract are in Appendix B.

8. Conclusion

In this paper, we compared the ability of different LLMs to accurately assign the thematic areas of ACL papers (ARR areas), using different prompting strategies (one-shot vs. few shot, prompting, inclusion of area-specific keywords). For this, we compiled a data set of 558 papers from ACL, EACL and NAACL, spanning from 2020 to 2025.

Our analyses of different prompting techniques do not fully conform to previous findings (Brown et al., 2020), which report higher accuracy with an increased number of in-context examples. In our experiments, the highest overall accuracy was achieved with one-shot prompting, suggesting that adding more examples does not necessarily benefit classification performance. Similarly, broadening the prompt to include all ARR areas in the dataset did not consistently improve model accuracy, with only one model showing performance gains. We also observed a substantial number of instances in which all models selected areas that differed from the authors' labels; in such cases, the

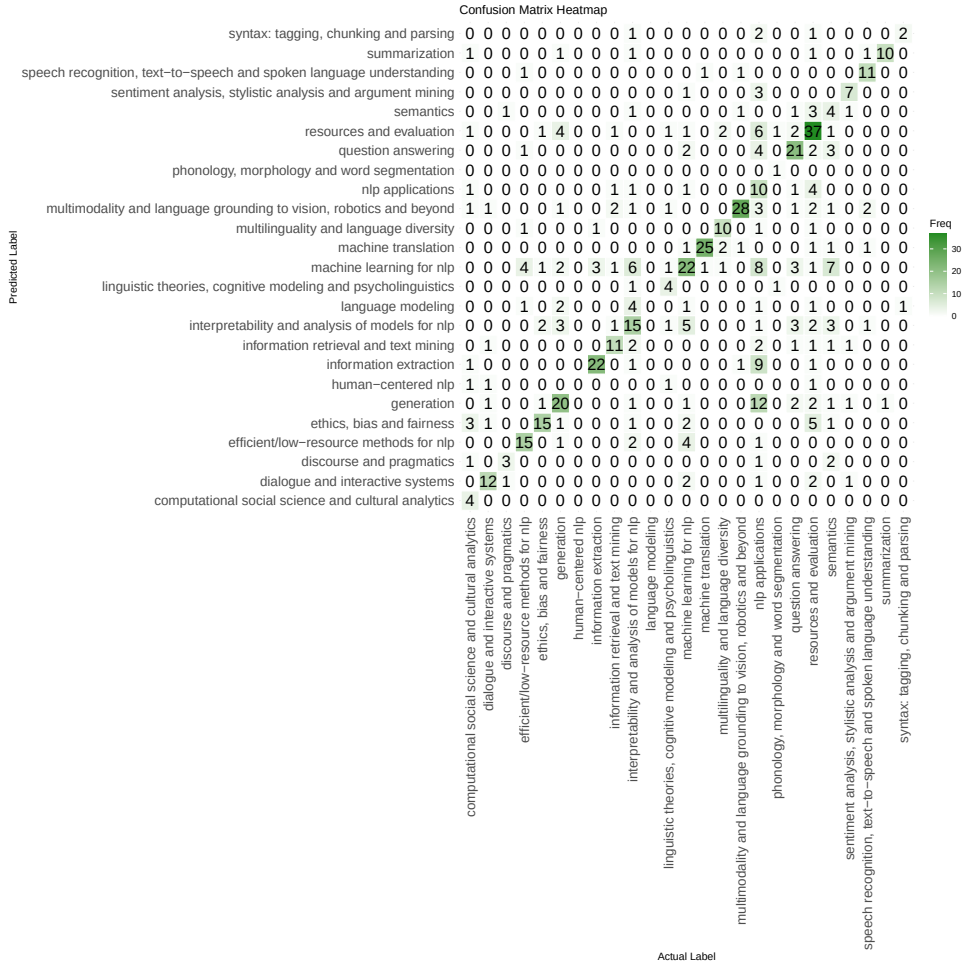


Figure 6: Confusion matrix for paper to area classification by the OpenAI-gpt-oss-20b model

models' predictions may be more thematically accurate than the authors' self-assigned areas. Overall, our results suggest that authors may experience difficulties in selecting thematically appropriate areas, potentially due to ambiguity in the area names and associated keywords.

Including the ARR's human-generated keywords, did not aid the models in the classification task. This might be due to the overlap of keywords in different areas. The keywords might not reflect the content of the papers in the area, contrasting with the semantic knowledge of the LLM.

The area labelling and keyword assignment is a dynamic process within ACL. Given our results, we suggest not only constant area revision aided by similar analyses to reflect the changing ACL landscape (as done in Rogers et al. (2023)), but also to install selection procedures of areas that reduce systematic confusions and aid authors to select areas in more systematic ways. An example could be hierarchically structured areas or the selection of a primary plus secondary labelling (e.g., task/domain as primary; role/method such as "resources" "ethics" or "human-centered" as secondary or vice versa).

9. Limitations

This work has several limitations. First, the number of LLMs evaluated was small. Although we tested various prompting strategies, including zero- and few-shot approaches with or without keywords, results may differ with other state-of-the-art or proprietary models. More advanced prompting and stronger LLMs could improve performance.

Second, the evaluation dataset was limited to 558 papers with uneven coverage across ARR research areas. This may have influenced performance differences and limits generalisability. Expanding the dataset and revising the ARR taxonomy, for example, using hierarchical or multi-label schemes, could reduce sampling bias, clarify label ambiguities, and better capture research complexities.

10. Acknowledgements

This research is funded by *Deutsche Forschungsgemeinschaft* (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.

11. Bibliographical References

2025a. ACL Rolling Review. https://aclrollingreview.org/. Accessed Sep 23, 2025.

2025b. ARR Area Keywords. https://aclrollingreview.org/areas. Accessed Sep 23, 2025.

2025. New Tracks at EMNLP 2025 and Their Relationship to ARR Tracks. https://2025.emnlp.org/track-changes/. Accessed Sep 23, 2025.

Raia Abu Ahmad, Ekaterina Borisova, and Georg Rehm. 2024a. Forc4cl: A fine-grained field of research classification and annotated dataset of nlp articles. In *International Conference on Language Resources and Evaluation*.

Raia Abu Ahmad, Ekaterina Borisova, and Georg Rehm. 2024b. Forc@nslp2024: Overview and insights from the field of research classification shared task. In *NSLP*.

Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Fabio Dennstädt, Johannes Zink, Paul Martin Putora, Janna Hastings, and Nikola Cihoric. 2024. Title and Abstract Screening for Literature Reviews Using Large Language Models: An Exploratory Study in the Biomedical Domain. *Systematic Reviews*, 13(158).

Jennifer D'souza, Sameer Sadruddin, Holger Israel, Mathias Begoin, and Diana Slawig. 2025. The Germeval 2025 2nd LLMs4Subjects Shared Task Dataset. Online dataset. Data managers and curators: Jennifer D'Souza (Data manager), Sameer Sadruddin (Data curator).

Clara Wan Ching Ho. 2025. UBFFM at the GermEval-2025 LLMs4Subjects Task: What if we take "You are an expert in subject indexing" seriously? In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, pages 471–478, Hannover, Germany. HsH Applied Academics.

Lucas Joos, Daniel A. Keim, and Maximilian T. Fischer. 2024. Cutting Through the Clutter: The Potential of LLMs for Efficient Filtration in Systematic Literature Reviews. arXiv:2407.10652.

Taja Kuzman and Nikola Ljubešić. 2025. Llm teacher-student framework for text classification with no manually annotated data: A case study in iptc news topic classification. *IEEE access*.

Taja Kuzman, Igor Mozetič, and Nikola Ljubešić. 2023. ChatGPT: Beginning of an End of Manual Linguistic Data Annotation? Use Case of Automatic Genre Identification.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of EMNLP*.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b Model Card.

Qwen Team. 2025. Qwen3 Technical Report.

Anna Rogers, Marzena Karpinska, Jordan Boyd-Graber, and Naoaki Okazaki. 2023. Program chairs' report on peer review at acl 2023. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages xl–lxxv, Toronto, Canada. Association for Computational Linguistics.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *arXiv preprint arXiv:2402.07927*.

Samuel Schulhoff, Yuchen Liu, et al. 2024. A Systematic Survey of Prompt Engineering Techniques. *arXiv preprint arXiv:2406.06608*.

Parisa Shirali, Zahra Sarlak, and Ebrahim Ansari. 2025. Last Minute at the GermEval-2025 LLMs4Subjects Task: Few-Shot Contrastive Learning for Multilingual Multi-Label Classification. In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, pages 465–470, Hannover, Germany. HsH Applied Academics.

Tanja Säily, Jukka Suomela, Florent Perek, Jimena Jiménez Real, and Turo Vartiainen. 2025. Using Large Language Models to Enrich Corpus Metadata: The Case of Novels in COHA. In *46th Annual Conference of the International Computer Archive of Modern and Medieval English (ICAME 46)*, Vilnius, Lithuania.

Teknium, interstellarninja, theemozilla, karan4d, and huemin_art. 2024. Hermes-2-Pro-Llama-3-8B.

Kun Zhu, Lizi Liao, Yuxuan Gu, Lei Huang, Xiaocheng Feng, and Bing Qin. 2025. Context-aware hierarchical taxonomy generation for scientific papers via llm-guided multi-aspect clustering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15627–15645.

A. Prompt template

Role:
Act as a librarian and organize a collection of articles from the Association for Computational Linguistics (ACL), published between 2020 and 2025.
Objective:
Your task is to read, analyze, and organize a corpus of abstracts for papers from the following events:

- Association for Computational Linguistics (ACL)
- European Chapter of the Association for Computational Linguistics (EACL)
- Nations of America Chapter of the Association for Computational Linguistics (NAACL)

In general, the Association for Computational Linguistics focuses on the study of computational linguistics and natural language processing. Your goal is to create a comprehensive and structured database that facilitates search, and retrieval of these articles by researchers, and scientists.

Input:
You will be provided with instances from a dataset of astracts from the Association of Computational Linguistics. The dataset will consist of the abstracts of the original articles, along with some of their corresponding metadata:

- bibtex-style id
- title
- author(s)
- publication year
- event (ACL, EACL, NAACL)
- paper type (long or short)

Tasks:
A. Read and analyze the provided article to understand its core contribution and research context.
B. Identify the area of the paper. Areas describe the main idea of the paper and are used in the reviewing process for reviewer assignment. You should choose an area that: - accurately represents the content of the provided abstract

The area should exclusively be from this list of official areas:

1. Computational Social Science and Cultural Analytics
2. Dialogue and Interactive Systems
3. Discourse and Pragmatics
4. Efficient/Low-Resource Methods for NLP
5. Ethics, Bias, and Fairness
6. Generation
7. Human-Centered NLP
8. Information Extraction
9. Information Retrieval and Text Mining
10. Interpretability and Analysis of Models for NLP
11. Language Modeling
12. Linguistic Theories, Cognitive Modeling, and Psycholinguistics
13. Machine Learning for NLP
14. Machine Translation
15. Multilinguality and Language Diversity
16. Multimodality and Language Grounding to Vision, Robotics and Beyond
17. NLP Applications
18. Phonology, Morphology, and Word Segmentation
19. Question Answering
20. Resources and Evaluation
21. Semantics
22. Sentiment Analysis, Stylistic Analysis, and Argument Mining
23. Speech Recognition, Text-to-Speech and Spoken Language Understanding
24. Summarization
25. Syntax: Tagging, Chunking and Parsing

Do not use external sources such as the internet for this task.

Example Input:
ID: Sicilia_Gates_Alikhani_EACL_2024
Title: HumBEL: A Human-in-the-Loop Approach for Evaluating Demographic Factors of Language Models in Human-Machine Conversations
Authors: Anthony Sicilia, Jennifer Gates, Malihe Alikhani
Year: 2024
Event: eacl
Type: long
Abstract: While demographic factors like age and gender change the way people talk, and in particular, the way people talk to machines, [...].

Example Output (in YAML format):
article_id: "Sicilia_Gates_Alikhani_EACL_2024"
acl_area: "Dialogue and Interactive Systems"

Ensure that you output is a valid YAML file that follows the format provided above. No additional text output besides the YAML is required. Respect this rule and you will be rewarded accordingly.

B. Example of incorrectly classified paper

Article ID: 'Rambelli_Chersoni_Collacciani_Bolognesi_ACL_2024'
Title: 'Can Large Language Models Interpret Noun-Noun Compounds? A Linguistically-Motivated Study on Lexicalized and Novel Compounds'
Event: 'acl'
Year: '2024'
Author: 'Giulia Rambelli, Emmanuele Chersoni, Claudia Collacciani, Marianna Bolognesi'
Abstract: 'Noun-noun compounds interpretation is the task where a model is given one of such constructions, and it is asked to provide a paraphrase, making the semantic relation between the nouns explicit, as in carrot cake is "a cake made of carrots." Such a task requires the ability to understand the implicit structured representation of the compound meaning. In this paper, we test to what extent the recent Large Language Models can interpret the semantic relation between the constituents of lexicalized English compounds and whether they can abstract from such semantic knowledge to predict the semantic relation between the constituents of similar but novel compounds by relying on analogical comparisons (e.g., carrot dessert). We test both Surprisal metrics and prompt-based methods to see whether i.) they can correctly predict the relation between constituents, and ii.) the semantic representation of the relation is robust to paraphrasing. Using a dataset of lexicalized and annotated noun-noun compounds, we find that LLMs can infer some semantic relations better than others (with a preference for compounds involving concrete concepts). When challenged to perform abstractions and transfer their interpretations to semantically similar but novel compounds, LLMs show serious limitations.'

Benchmarking Retrieval-Augmented Generation for Scientific Knowledge QA in European Portuguese

José Matos, Catarina Silva, Hugo Gonçalo Oliveira
CISUC, LASI, Portugal
DEI, University of Coimbra, Portugal
josematos@student.dei.uc.pt, {catarina,hroliv}@dei.uc.pt

Abstract

Retrieval-Augmented Generation (RAG) enables grounding of model outputs in external evidence, but its impact on European Portuguese (pt-PT) scientific question answering (QA) remains unclear. We present a controlled evaluation of RAG on pt-PT knowledge QA across different scientific domains using the Portuguese test split of the Global MMLU Lite dataset. As external evidence, we use a Portuguese scientific literature knowledge base containing over 32,000 documents converted to Markdown. We benchmark five instruction-tuned small language models (4-12B) and compare closed-book baselines against 16 RAG configurations that vary by: (i) dense retriever specialization (multilingual vs. Portuguese-specific), (ii) reranking (on/off), and (iii) number of retrieved chunks ($k \in \{1, 3, 5, 10\}$). Results suggest that RAG gains are model-dependent. Some models improve consistently, others are highly sensitive to retrieval choices, and some degrade under retrieval noise, especially at larger values of k. Findings highlight the importance of model-specific retrieval tuning and ensuring that the retriever and reranker languages and domains align when deploying RAG systems for Portuguese natural scientific language processing.

Keywords: Retrieval-Augmented Generation, Science, Evaluation, Portuguese, Small Language Models

1. Introduction

Large Language Models (LLMs) have demonstrated strong capabilities in question-answering (QA), even in highly technical scientific tasks (Zheng et al., 2025). However, as these models are employed in areas not covered by their training data, they are prone to issues such as *hallucinations* (Zhang et al., 2025). This problem compounds both when we move from LLMs with hundreds of billions of parameters to smaller models with limited parametric capacity, known as Small Language Models (SLMs), and when we move away from English to multilingual settings (Ul Islam et al., 2025). To overcome this, Retrieval-Augmented Generation (RAG) enables models to ground their responses in external knowledge (Lewis et al., 2020), helping to mitigate hallucinations by extending parametric knowledge and allowing models to reference human-verifiable sources.

For Portuguese, and European Portuguese (pt-PT) in particular, the recent appearance of open-source LLMs (Lopes et al., 2024; Martins et al., 2025; Ramos et al., 2026; Simplício et al., 2026), and native sentence encoders (Gomes et al., 2024) motivate the testing of whether these retrieval systems can effectively be transferred to pt-PT scientific knowledge QA. A limited number of works have studied the impact of RAG in Portuguese-specific settings. Finardi et al. (2024) reported a set of good practices for implementing and improving RAG systems in Brazilian Portuguese, by optimizing the retrieval pipeline and evaluating the performance of proprietary models on a QA dataset generated from a book. Instead of focusing solely on retrieval, Costa and Souza-Filho (2024) compared fine-tuning and RAG for adapting LLMs to specific domains for QA tasks in Brazilian Portuguese, finding that a combination of both is the best overall strategy. Both works focus on the Brazilian variety of Portuguese, highlighting the gap relative to pt-PT. We focus on a specific question:

How does RAG improve pt-PT scientific QA across open instruction-tuned SLMs, and how sensitive is it to retrieval setup choices?

To this end, we present a controlled empirical evaluation of retrieval-augmented multiple-choice scientific knowledge QA in Portuguese. Models are evaluated on the Portuguese split of Global MMLU Lite, whose categories correspond to disciplinary fields of science (e.g., STEM, Medical, and Social Sciences) and are paired with an external knowledge base of over 32,000 Portuguese scientific documents. We systematically benchmark five instruction-tuned SLMs, varying the *generator* and *retriever* models, retrieved context sizes, and reranking strategies. The goal is to establish a reproducible baseline for different RAG configurations in the pt-PT scientific knowledge QA setting. This work is part of a larger project aimed at developing a scientific reasoning LLM native to European Portuguese, including synthetic data generation, training, and evaluation techniques tailored to lower-resource languages.

Our contributions include:

1. A pt-PT scientific QA baseline comparing closed-book with different retrieval configurations across five different open SLMs;

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 25–31

2. Controlled ablations of retriever language specializations, reranking, and context volume (k).

2. Methodology

In this section, we describe the data used for our evaluation and external knowledge base, followed by the specific SLMs selected for benchmarking.

2.1. Data

We evaluate models on the Portuguese test split of Global MMLU Lite[1], a compact version of Global MMLU, created by Singh et al. (2025), a benchmark combining machine translations for MMLU (Hendrycks et al., 2020) along with professional translations and crowd-sourced edits. The lite version comprises 400 test samples per language, spanning 18 languages (example test sample in Appendix A). For Portuguese, all samples were professionally translated. The dataset is split into the following categories: Humanities, STEM, Medical, Social Sciences, Business, and Other.

As the external knowledge base, we use over 32,000 open-access scientific documents in Portuguese, in Markdown format, ranging from book chapters to PhD theses, present in the CorEGe-PT[2] corpus (Kuhn et al., 2026). The texts in this dataset were sourced from a large academic repository in Portugal and span five scientific fields, with the majority overlapping with the subcategories in the evaluation benchmark: Exact and Natural Sciences, Engineering and Technology Sciences, Medical and Health Sciences, Social Sciences, and Humanities.

2.2. Models

We evaluate five non-quantized instruction-tuned SLMs at a comparable size range (4-12B): AMALIA (Simplício et al., 2026), EuroLLM-9B-Instruct (Martins et al., 2025), Qwen3-8B (Qwen Team, 2025), and finally Gemma3-4B-IT and Gemma3-12B-IT (Gemma Team, 2025).

AMALIA is a 9 billion parameter LLM trained with an emphasis on pt-PT. It is derived from EuroLLM-9B-Instruct by incorporating high-quality pt-PT data during the annealing phase of pretraining (Simplício et al., 2026). This makes it a clear choice to evaluate the impact of training on pt-PT specialized data. The pretraining phase of Qwen-3-8B includes 5 trillion high-quality STEM domain tokens (Qwen Team, 2025), making it important to test how strong science-oriented pretraining can impact the effects of retrieval on this task. The choice of Gemma3-4B-IT and Gemma3-12B-IT aims to assess the impact of model size on this task, while keeping the architecture family fixed.

3. Experimental Setup

In this section, we describe and discuss the experimental design choices for our evaluation, outlining the retrieval configurations tested and the evaluation protocol used.

3.1. Indexing and Retrieval Configurations

All gathered documents were indexed in a vector database. Given the maximum input length constraints of LLMs, and especially sentence encoders, documents are split into chunks in two phases. First, split by markdown section headers, and then recursively split into 500 character chunks with 50 character overlaps to preserve semantic continuity. After splitting, chunks are then embedded and stored in a *ChromaDB*[3] instance with cosine for computing similarity.

3.1.1. Encoder language specialization

To evaluate the impact of language specialization on indexing and retrieval, we compare two distinct dense embedding models for encoding corpus chunks and query passages. First, we use `paraphrase-multilingual-mpnet-base-v2`[4], a general-purpose multilingual encoder (278M parameters) trained on over 50 languages (Reimers and Gurevych, 2019). We compare this against `serafim-335m-portuguese-pt-sentence-encoder-ir`[5], a Portuguese-specific model from the Serafim family (Gomes et al., 2024). By selecting the 335M parameter version of Serafim, we maintain a comparable size to the multilingual baseline, effectively isolating the impact of language specialization over model scale.

3.1.2. Context selection and candidate pool

To assess the effects of context volume and potential noise, we evaluate different numbers of retrieved chunks ($k \in \{1, 3, 5, 10\}$). For efficiency and determinism in the retrieval step, we pre-compute the top 30 chunks per question for both encoders using the question stem as the query. During retrieval at inference, we slice the top-k chunks from

[1] https://huggingface.co/datasets/CohereLabs/Global-MMLU-Lite

[2] https://huggingface.co/datasets/NLP-CISUC/CorEGe-PT

[3] https://github.com/chroma-core/chroma

[4] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[5] https://huggingface.co/PORTULAN/serafim-335m-portuguese-pt-sentence-encoder-ir

Model	Base (CB)	Best RAG (R)	Best RAG+R (R+R)	Δ_{BEST}(pp)
AMALIA	63.0 ± 2.4	64.7 ± 2.4 (*pt*, k=10)	65.7 ± 2.4 (*pt*, k=1)	+2.7
EuroLLM-9B-Instruct	62.0 ± 2.4	63.5 ± 2.4 (*pt*, k=1)	64.2 ± 2.4 (*pt*, k=10)	+2.2
Qwen3-8B	73.0 ± 2.2	72.0 ± 2.2 (*ml*, k=3)	71.8 ± 2.3 (*pt*, k=1)	-1.0
Gemma-3-4B-IT	50.7 ± 2.5	60.3 ± 2.4 (*ml*, k=3)	60.5 ± 2.4 (*ml*, k=5)	**+9.8***
Gemma-3-12B-IT	55.5 ± 2.5	70.5 ± 2.3 (*pt*, k=1)	71.0 ± 2.3 (*pt*, k=1)	**+15.5***

Table 1: **Overall accuracy (%) ± standard error (pp) on Global MMLU Lite (PT split)**. k is the number of chunks added in context. Δ_{BEST} calculated with the best-performing RAG configuration (with or without reranking) minus baseline (CB), in pp. (*pt* - Portuguese sentence encoder; *ml* - Multilingual sentence encoder; * $p < 0.05$ against CB, two-proportion Z test and McNemar).

this pool. For evaluations involving a *reranking* step, we keep a separate pool of chunks for each question and embedding model, previously reranked using a fixed multilingual cross-encoder reranker[6].

3.1.3. Retrieval variants

To further isolate the impact of different components of the retrieval pipeline on performance, we evaluate three configurations: (i) Closed-Book (CB) uses only parametric and question knowledge, serving as a reference for the model's base performance on the task. (ii) Retrieval (R) appends the top-k retrieved chunks as context and (iii) Retrieval+Reranking (R+R) uses the reranked top-k chunks.

3.2. Evaluation Protocol

We use *lm_evaluation_harness*[7] to evaluate the selected models and configurations using a 3-shot prompting technique. Few-shot exemplar selection is deterministic (fixed seed, default for framework), so that comparisons across different retrieval configurations use the same demonstrations, attributing any differences to the retrieval setup rather than exemplar variation. Scoring is based on option log-likelihood (standard for multiple-choice evaluation) rather than decoding, thus sampling parameters such as temperature do not affect the selected answer. For consistency, we keep the prompt format fixed across configurations, varying only the presence and amount of retrieved context (k) and the application of reranking. The prompt template used for every test item is provided in Appendix A. All experiments were run using a single NVIDIA RTX A6000 GPU.

[6] https://huggingface.co/Alibaba-NLP/gte-multilingual-reranker-base

[7] https://github.com/EleutherAI/lm-evaluation-harness

4. Results and Discussion

We evaluate a grid of 16 retrieval configurations per model (2 sentence encoders × 4 values of k × reranking). For compactness, Table 1 reports the maximum (upper-bound) accuracy ± standard error observed within these configurations. To evaluate robustness across all configurations, Table 2 summarizes the distribution of Δ accuracy from the baseline (CB) across all tested configurations.

Model	Δ_{WORST}	Δ_{BEST}	Mean ± SD Δ
AMALIA	-1.50	2.75	0.53 ± 1.30
EuroLLM-9B-Instruct	-2.00	2.25	0.08 ± 1.04
Qwen3-8B	-4.75	-1.00	-2.89 ± 1.18
Gemma-3-4B-IT	6.25	9.75	8.41 ± 1.08
Gemma-3-12B-IT	**11.75**	**15.50**	**13.61 ± 1.08**

Table 2: **Robustness across configurations.** Distribution of Δ accuracy (pp) over the 16 RAG configurations per model.

For Gemma models, the positive effect of RAG is clear. Accuracy increases range from 9.8 percentage points (pp) in the 4B to 15.5 pp in the 12B model, with gains consistent across all tested components and retrieval settings. Results from Table 2 further support this impact, indicating gains are strongly positive across all tested configurations for these models (8.41 ± 1.08 pp for the 4B, and 13.61 ± 1.08 pp for the 12B version). The best retrieval configurations show +2.7 pp increase for AMALIA and +2.2 pp for EuroLLM-9B-Instruct (Table 1), while Table 2 shows that the average gains for these models near zero with around 1 pp of standard deviation, suggesting they are heavily sensitive to retrieval configurations, and can even, in some cases, degrade their performance under RAG. Qwen3-8B achieves the highest baseline accuracy but shows slight degradation across all retrieval configurations, with larger context volumes (k) yielding worse results (Figure 1), suggesting that retrieval noise may be detrimental to this model's performance.

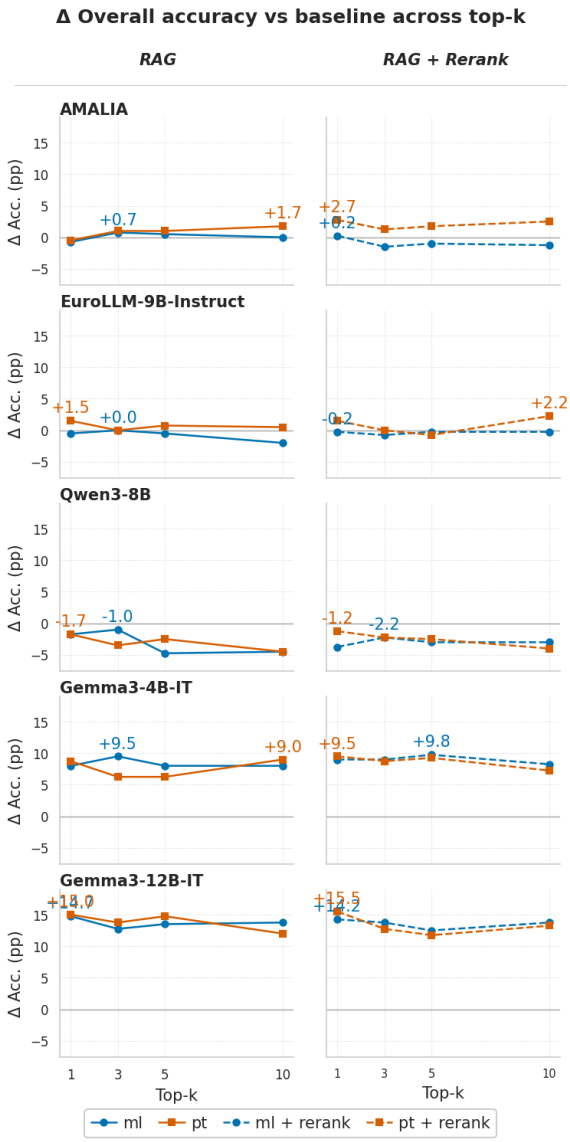


Figure 1: **Effect of context volume k on accuracy.** Δ Overall accuracy (pp) relative to the closed-book (CB) baseline as a function of $k \in \{1, 3, 5, 10\}$. Rows correspond to different generator models.

Paired tests across subcategories and proportion-based tests on overall accuracy show that only the Gemma models achieve statistically significant gains over closed-book, while AMALIA and EuroLLM-9B-Instruct show non-significant gains, and Qwen shows a consistent yet non-significant decrease.

Figure 1 suggests that performance does not scale linearly with added context. For some models, performance peaks at lower values of k, and then plateaus/decreases as additional chunks are added to the context. This aligns with the trade-off between useful evidence and noisy context as recall increases. Reranking does not guarantee a consistent accuracy improvement, but can shift the best-performing values of k toward smaller contexts, improving efficiency.

Without reranking, when using the Portuguese-specific sentence encoder, models often outperform the multilingual baseline, particularly in the Business, STEM, and Medical categories. Figure 2 indicates that language specialization interacts with domains and context size rather than acting as a universal improvement to performance. These three categories are the most technical in the group and include specialized terminology, suggesting the advantage provided by the Portuguese-specific encoder could be tied to the training data being more representative of these domains, compared to the vast multilingual encoder's training data.

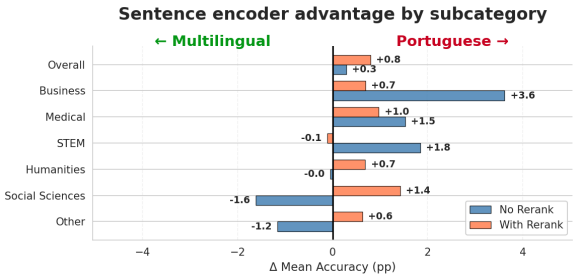


Figure 2: **Δ Accuracy between RAG configurations using different sentence encoders.** Results are reported with and without reranking.

Again, in the technical domains, it is worth noting that reranking can degrade the performance gains achieved with the Portuguese-specific encoder. This degradation is likely due to a mismatch in language in the retrieval pipeline, as the multilingual cross-encoder used for reranking may lack the granular understanding of specialized pt-PT terminology needed to accurately score the highly technical chunks. Moreover, in less technical domains like Social Sciences, the multilingual reranker successfully corrects the initial retrieval, from a -1.6 pp deficit to a +1.4 pp advantage. This highlights a central design consideration in selecting reranking models, suggesting that coupling a language-specific retriever with a general multilingual reranker can erase domain-specific gains. Ultimately, further work should explore this by coupling a language-specific retriever and reranker.

5. Conclusion

We established a baseline for scientific knowledge QA in pt-PT using RAG. Evaluating 5 SLMs of similar size reveals that the effects of RAG are highly model-dependent. By testing two different sentence encoders for chunk indexing and retrieval

purposes, one multilingual and one specific to Portuguese, we find that the impact of language specialization can vary depending on the knowledge domain being targeted. We also test the isolated effect of reranking after retrieval, and the influence of different top-k sizes on performance.

Future work should evaluate these systems beyond performance on broad scientific knowledge multiple-choice QA tasks and address open-response questions, particularly in highly technical domains, including scientific reasoning tasks. In addition, these findings motivate further efforts toward grounding-oriented, task-specific training and evaluation methods, as well as effective and efficient selection of relevant context from scientific documents, for natural scientific language processing in Portuguese.

Limitations

Due to the lack of Portuguese evaluation resources (and pt-PT specifically), we resorted to the Global MMLU Lite dataset. The performance of the systems we currently aim to develop, the ones capable of effective reasoning over scientific knowledge, may not be fully captured by this type of benchmark. Future work should focus on creating language-specific datasets in pt-PT to assess this impact. Additionally, because it is not directly targeted for pt-PT, linguistic biases toward the Brazilian variety are likely present in the questions, and consequently affect the retrieval step.

Another limitation is that retrieval quality is not directly evaluated at the chunk level. Instead, we use task accuracy as a proxy for this measure. Furthermore, only a single chunking strategy was used. Alternative approaches, such as semantic chunking or larger windows, may result in different outcomes.

Additionally, for this specific task, multiple-choice log-likelihood evaluation is appropriate for assessing knowledge capabilities. Still, further work should explore the capacity of these models for pt-PT open-ended answer generation, possibly employing prompting techniques such as Chain-of-Thought (Wei et al., 2022), which would be more aligned with the real-world use of the targeted systems.

Finally, we note that other models could serve as additional size comparisons, such as Qwen3-4B, which may be explored in future work.

Code & Data Availability

The code and data used for the experiments, as well as the results, are available at https://github.com/NLP-CISUC/Benchmark-ScientificQA-RAG-pt-PT.

Acknowledgments

This work was partially supported by the AMALIA project, funded by FCT/IP in the context of measure RE-C05-i08 of the Portuguese Recovery and Resilience Program; by the Portuguese Recovery and Resilience Plan through project C645008882-00000055, Center for Responsible AI; and by national funds through FCT – Foundation for Science and Technology I.P., in the framework of the Project CISUC (UIDB/00326/2025 and UIDP/00326/2025).

6. Bibliographical References

Leandro Costa and João Baptista de Oliveira e Souza-Filho. 2024. Adapting LLMs to New Domains: A Comparative Study of Fine-Tuning and RAG strategies for Portuguese QA Tasks. In *Proceedings of the 15th Brazilian Symposium in Information and Human Language Technology*, pages 217–227, Belém do Pará, Brazil. Association for Computational Linguistics.

Paulo Finardi, Leonardo Avila, Rodrigo Castaldoni, Pedro Gengo, Celio Larcher, Marcos Piau, Pablo Costa, and Vinicius Caridá. 2024. The Chronicles of RAG: The Retriever, the Chunk and the Generator. *arXiv Preprint arXiv:2401.07883*.

Gemma Team. 2025. Gemma 3 Technical Report. *arXiv Preprint arXiv:2503.19786*.

Luís Gomes, António Branco, João Silva, João Rodrigues, and Rodrigo Santos. 2024. Open Sentence Embeddings for Portuguese with the Serafim PT* Encoders Family. In *Progress in Artificial Intelligence: 23rd EPIA Conference on Artificial Intelligence, EPIA 2024, Viana Do Castelo, Portugal, September 3–6, 2024, Proceedings, Part III*, pages 267–279, Berlin, Heidelberg. Springer-Verlag.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Tanara Zingano Kuhn, José Matos, Bruno Neves, Daniela Pereira, Elisabete Cação, Ivo Simões, Jacinto Estima, Delfim Leão, and Hugo Gonçalo Oliveira. 2026. CorEGe-PT: Compiling a Large Corpus of Academic Texts in Portuguese. In *Proceedings of 15th Language Resources and Evaluation Conference*, LREC 2026, page Accepted. ELRA.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,

Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

Ricardo Lopes, Joao Magalhaes, and David Semedo. 2024. GlórIA: A Generative and Open Large Language Model for Portuguese. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 441–453, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2025. EuroLLM: Multilingual Language Models for Europe. *Procedia Computer Science*, 255:53–62.

Qwen Team. 2025. Qwen3 technical report. *arXiv Preprint arXiv:2505.09388*.

Miguel Moura Ramos, Duarte M. Alves, Hippolyte Gisserot-Boukhlef, João Alves, Pedro Henrique Martins, Patrick Fernandes, José Pombal, Nuno M. Guerreiro, Ricardo Rei, Nicolas Boizard, Amin Farajian, Mateusz Klimaszewski, José G. C. de Souza, Barry Haddow, François Yvon, Pierre Colombo, Alexandra Birch, and André F. T. Martins. 2026. EuroLLM-22B: Technical Report. *arXiv Preprint arXiv:2602.05879*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Afonso Simplício, Gonçalo Vinagre, Miguel Moura Ramos, Diogo Tavares, Rafael Ferreira, Giuseppe Attanasio, Duarte M. Alves, Inês Calvo, Inês Vieira, Rui Guerra, James Furtado, Beatriz Canaverde, Iago Paulo, Vasco Ramos, Diogo Glória-Silva, Miguel Faria, Marcos Treviso, Daniel Gomes, Pedro Gomes, David Semedo, André Martins, and João Magalhães. 2026. AMALIA Technical Report: A Fully Open Source Large Language Model for European Portuguese. In *Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)*.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2025. Global MMLU: Understanding and Addressing Cultural and Linguistic Biases in Multilingual Evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Saad Obaid Ul Islam, Anne Lauscher, and Goran Glavaš. 2025. How Much Do LLMs Hallucinate across Languages? On Realistic Multilingual Estimation of LLM Hallucination. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29077–29098, Suzhou, China. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *Computational Linguistics*, 51(4):1373–1418.

Tianshi Zheng, Zheye Deng, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Zihao Wang, and Yangqiu Song. 2025. From Automation to Autonomy: A Survey on Large Language Models in Scientific Discovery. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17733–17750, Suzhou, China. Association for Computational Linguistics.

A. Additional Information

Example from the pt split of the Global MMLU Lite Dataset

subject_category: "STEM"

question: "Os materiais usados no dissipador térmico devem ter:"

option_a: "alta condutividade térmica."
option_b: "grande superfície."
option_c: "alto ponto de fusão."
option_d: "Todas as anteriores."

answer: "D"

Example from the pt split of the Global MMLU Lite Dataset (translated)

subject_category: "STEM"

question: "The materials used in the heat sink must have:"
option_a: "high thermal conductivity."
option_b: "large surface area."
option_c: "high melting point."
option_d: "All of the above."

answer: "D"

Figure 3: Example from the Portuguese test split of the Global MMLU Lite benchmark (*electrical_engineering/test/78*)
.

Prompt template for evaluation

<few_shot_example_1>
<few_shot_example_2>
<few_shot_example_3>

Contexto Adicional:
<retrieved_chunk_1>;
<retrieved_chunk_2>;
(...)
<retrieved_chunk_K>;

Agora responde à pergunta seguinte. Se necessário utiliza o contexto adicional:
<question>
<options>
Resposta: (A/B/C/D)

Prompt template for evaluation (translated)

<few_shot_example_1>
<few_shot_example_2>
<few_shot_example_3>

Additional Context:
<retrieved_chunk_1>;
<retrieved_chunk_2>;
(...)
<retrieved_chunk_K>;

Now answer the following question. If necessary, use the additional context:
<question>
<options>
Answer: (A/B/C/D)

Figure 4: Prompt used for evaluation on the Global MMLU Lite benchmark.

Beyond Abstracts: A Biomedical MeSH Indexing Corpus Incorporating Summarized Methods Sections

Sujoy Datta[1], **Robert E. Mercer**[1], **Xindi Wang**[2]
[1]Department of Computer Science, University of Western Ontario, London, Ontario, Canada
[2]School of Artificial Intelligence, Shandong University, Jinan, China
sdatta46@uwo.ca, mercer@csd.uwo.ca, xindi.wang@sdu.edu.cn

Abstract

Automated Medical Subject Heading (MeSH) indexing systems rely predominantly on titles and abstracts, while human indexers at the National Library of Medicine examine full-text articles—particularly Methods sections—that often contain crucial experimental terminology absent from abstracts. This information asymmetry limits model performance and prevents detection of methodologically-grounded MeSH descriptors. We introduce a novel biomedical MeSH indexing corpus comprising over one million English biomedical articles, each annotated with title, abstract, journal metadata, publication year, expert-curated MeSH terms, and—uniquely—extractive summaries of Methods sections. Using LLaMA 3 with an iterative re-prompting strategy, we generated high-fidelity summaries. To avoid label leakage, evaluation labels are inferred using journal-specific MeSH frequency profiles rather than gold annotations. This publicly accessible dataset addresses a critical gap in full-text MeSH indexing research. Building upon this resource, we propose an extended multi-channel neural architecture that incorporates Methods-derived representations. Empirical results demonstrate consistent performance gains across both example-based and label-based evaluations, indicating better retrieval of infrequent terms. These findings highlight that procedural knowledge in the Methods section encodes critical semantic cues overlooked by title-abstract only models.

Keywords: Text Classification, MeSH Indexing, Extractive Summarization, Large Language Model

1. Introduction

PubMed[1] and MEDLINE[2] are among the most authoritative and extensively used repositories of biomedical literature, both curated by the United States National Library of Medicine (NLM). MEDLINE serves as the central bibliographic database, encompassing scholarly records from diverse domains within the biomedical and life sciences, including—but not limited to—clinical medicine, molecular biology, epidemiology, public health, and evolutionary sciences. According to recent statistics, MEDLINE contains over 31 million citations sourced from more than 5,200 peer-reviewed journals across the globe[2].

In 2022 alone, MEDLINE incorporated nearly one million new citations, averaging approximately 3,000 additions per day[3]. PubMed functions as the publicly accessible search interface for MEDLINE, enabling efficient retrieval of bibliographic metadata and, where available, access to full-text articles via linked repositories such as PubMed Central.

Central to MEDLINE's indexing framework is the use of **Me**dical **S**ubject **H**eadings (MeSH), a hierarchically structured and manually curated controlled vocabulary developed by the NLM. MeSH enables semantically consistent categorization of biomedical concepts across varying levels of abstraction. As of the 2019 release, the thesaurus comprised 28,939 descriptors, including 29 check tags—specialized terms used to denote fundamental study attributes such as organism type, demographic factors, or experimental conditions.

MeSH constitutes the cornerstone of NLM's indexing infrastructure. It is employed by both human indexers and, since April 2024, by the MTIX (Medical Text Indexer–NeXt Generation)[4] system, which now automates preliminary annotation for all incoming MEDLINE citations. While MTIX generates candidate labels based primarily on titles and abstracts, human indexers remain responsible for quality assurance, reviewing and finalizing the assigned descriptors (Fernandez-Llimos et al., 2024).

Historically, MeSH term assignment relied almost exclusively on manual review conducted by expert annotators, who examined full-text articles to determine concept relevance. However, this process is both labor-intensive and financially costly, with the average annotation estimated at $9.40 per article (Wang et al., 2022). Given the scale at which new literature enters MEDLINE—often exceeding 3,000 articles per day—fully manual indexing is no longer viable as a standalone strategy. Hence, the development of scalable, high-fidelity automated MeSH indexing methods has become a critical priority.

[1]pubmed.ncbi.nlm.nih.gov/about/
[2]www.nlm.nih.gov/medline/medline_overview.html
[3]www.nlm.nih.gov/bsd/medline_cit_counts_yr_pub.html
[4]www.nlm.nih.gov/pubs/techbull/ma24/ma24_mtix.html

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 32–43
12 May 2026.

Despite significant advances in neural architectures for MeSH indexing, most automated systems remain constrained by the limited scope of input data available during training. Existing public corpora overwhelmingly provide only titles and abstracts, whereas human indexers at the National Library of Medicine routinely examine the full text when assigning descriptors. This discrepancy creates an inherent information asymmetry: crucial contextual cues—often embedded in structurally informative sections such as Methods, Results, or Discussion—are inaccessible to current models. Prior work by Mork et al. (2017) underscores that the Methods section, in particular, contains experimentally grounded terminology that can substantially enhance indexing accuracy. Our own preliminary analysis supports this observation. For example, in the biomedical article with PMID 27456232[5], the assigned MeSH term "Cholecalciferol" appears exclusively within the methodology description and is absent from both the title and abstract. Such cases illustrate that models restricted to title–abstract inputs are structurally incapable of recovering many legitimate MeSH terms. Consequently, there is a clear need for a publicly accessible corpus that includes other sections paired with human-curated MeSH annotations to facilitate the development of next-generation context-aware indexing systems.

In this study, we introduce a newly curated open-access MeSH indexing dataset[6] in which each entry integrates the PubMed identifier (PMID), article title, abstract, journal metadata, publication year, annotated MeSH terms, and the corresponding summarized Methods section. Building upon this resource, we further propose an extended multi-channel architecture that incorporates feature representations derived from multiple document sections. We benchmark this enhanced framework and report its baseline performance as a foundation for future research in full-text-aware MeSH indexing.

2. Related Work

A variety of biomedical corpora have been developed to facilitate MeSH-related research. Early resources such as OHSUMED (Hersh et al., 1994) and GENIA (Kim et al., 2003) provided MEDLINE abstracts annotated with MeSH-derived concepts, primarily supporting document retrieval and entity recognition. Subsequent corpora such as CHEMDNER (Krallinger et al., 2015), BC5CDR (Li et al., 2015), and NLM-CHEM (Islamaj et al., 2021) focused more narrowly on chemical and disease annotation. While valuable, these datasets are restricted to abstracts or specific entity types and thus do not fully capture the complexity of large-scale MeSH indexing. In contrast, MeSHup (Wang et al., 2022) is the first corpus to provide full-text articles with expert-assigned MeSH terms, enabling exploitation of richer sections such as Methods and Results.

Automatic MeSH indexing has evolved from heuristic systems to increasingly sophisticated neural architectures. Early efforts such as MTI (Aronson et al., 2004) relied on UMLS-based concept mapping and k-NN retrieval, while subsequent learning-to-rank approaches like MeSHLabeler (Liu et al., 2015) and DeepMeSH (Peng et al., 2016) combined multiple evidence sources through weighted ranking schemes. With the rise of deep learning, attention-based encoders over titles and abstracts, for example AttentionMeSH (Jin et al., 2018) and MeSHProbNet (Xun et al., 2019), demonstrated that concise textual inputs already offer strong predictive signals. Full-text methods such as FullMeSH (Dai et al., 2020) and BERTMeSH (You et al., 2021) further incorporated additional sections beyond abstracts, though typically through late fusion or uniform context modeling. KenMeSH (Wang et al., 2022) advanced this line by combining masked attention with GCN-based label embeddings, yet still restricted its representation space to title and abstract. More recently, the National Library of Medicine replaced its long-standing rule-based MTIA system with MTIX, a neural indexing model trained on millions of MEDLINE citations. While MTIX marks an important acknowledgment of neural architectures for large-scale indexing, its deployment remains opaque and limited to title–abstract inputs. Furthermore, the model is not publicly released, hindering reproducibility and preventing the community from evaluating its performance beyond NLM's internal metrics.

Most existing MeSH indexing systems rely solely on titles and abstracts. Full-text models such as FullMeSH and BERTMeSH have shown that incorporating additional article sections can yield notable performance gains, reinforcing the value of richer document structure. However, their underlying corpora were not released, limiting reproducibility and fair comparison. This highlights the pressing need for publicly accessible, section-structured biomedical datasets to enable systematic evaluation of multi-section MeSH indexing approaches.

3. Methodology

In this section, we outline the overall system design and introduce the deep learning architecture

[5] pubmed.ncbi.nlm.nih.gov/27456232/

[6] https://doi.org/10.5281/zenodo.19221828

determined to be most effective for our task. We first describe the dataset construction process, emphasizing the practical challenges encountered and the strategies adopted to address them—a contribution that stands on its own merit. We then detail the model architecture, with a focus on the role and interaction of its key components.

3.1. Dataset Construction

We build our dataset using the MeSHup corpus (Wang et al., 2022), which is derived from the November 2021 release of BioC-PMC (Comeau et al., 2019). The corpus contains 1,342,667 full-text biomedical articles in English, each annotated with MeSH terms assigned by professional indexers, providing reliable supervision for downstream learning tasks. In addition to the manual MeSH annotations, each record includes rich metadata—such as publication venue, author list, and publication year—stored in a structured JSON format.

For our work, we extracted seven key fields: PMID, title, abstract, Methods section, journal name, publication year, and the associated MeSH terms (Figure 1). Analysis of the Methods field revealed 1,135,757 articles with valid content, with lengths ranging from 8 to 645,880 characters. To ensure consistency in document representation, we retained only entries with non-empty title, abstract, and Methods sections for further processing and model training.


```
{
  "pmid": "10023767",
  "title": "An antiviral mechanism of nitric oxide: inhibition of a
     viral protease.",
  "abstractText": "Although nitric oxide (NO) kills or inhibits the
     replication of a variety of intracellular pathogens, the
     antimicrobial mechanisms of NO are unknown. Here, we identify a
      viral protease as a target of NO. ...",
  "mesh": {
    "D019943": "Amino Acid Substitution",
    "D000998": "Antiviral Agents",
    "D001665": "Binding Sites",
    "D003545": "Cysteine",
    "D003546": "Cysteine Endopeptidases",
    ...
  },
  "journal": "Immunity",
  "year": "1999",
  "METHODS": "Experimental Procedures. Cells and Viruses.
     Coxsackievirus B3 (CVB3) ... Measurement of 3Cpro Activity. ...
      Determination of S-Nitrosylation of 3Cpro. ..."
}
```


Figure 1: Sample Record Extracted from the MeSHup Corpus.

3.1.1. Requirement of Text Summarization

Titles and abstracts in academic publications are typically constrained by journal policies, resulting in relatively uniform length and structure across articles. Abstracts are commonly limited to 150–250 words to ensure clarity and conciseness (Pottier et al., 2024), while titles are similarly regulated through character limits to aid discoverability[7]. In contrast, Methods sections are rarely subject to strict length constraints and therefore exhibit substantial variability depending on experimental complexity. A large-scale analysis[8] of over 61K articles reported lengths ranging from 7 to 18,517 words, with a median of 1,126 words, a trend also observed in our dataset.

Although such detail in the Method section is essential for transparency and reproducibility, their length makes them impractical for efficient retrieval and downstream MeSH-based classification. Summarization offers a pragmatic compromise by retaining semantically relevant content while discarding redundant material. Prior work shows that summaries maintain task-relevant information (Mandale-Jadhav, 2025), lower computational cost during processing (Luo et al., 2024), and, in MeSH indexing settings, can even outperform full-text inputs for concept detection (Jimeno-Yepes et al., 2013). These observations motivate our decision to adopt summarization as a central component of our pipeline rather than treating it as an auxiliary preprocessing step.

3.1.2. Selection of Summarization Technique

Text summarization methods are commonly divided into extractive and abstractive paradigms. Extractive methods identify salient sentences from the source and concatenate them without altering phrasing, while abstractive methods generate rephrased summaries through intermediate representations (Luo et al., 2024). In this work, we adopt the extractive paradigm, as abstractive summarization requires sophisticated linguistic modeling—ranging from syntax and semantics to discourse planning—which makes it considerably more resource-intensive to implement reliably (Moradi and Ghadiri, 2019).

Extractive summarization further offers practical advantages for classification tasks: it acts as an implicit feature selection mechanism by reducing input length and thereby lowering the dimensionality of the representation space. This leads to improved computational efficiency and more interpretable downstream models (Kolcz et al., 2001; Ghodratnama et al., 2020; Nallapati et al., 2017).

Summarization techniques can also be categorized as supervised or unsupervised. Supervised approaches rely on human-annotated reference summaries and achieve strong performance when such resources exist (Basyal and Sanghvi, 2023). Unsupervised methods, in contrast, operate without labeled summaries by ranking and

[7] https://pmc.ncbi.nlm.nih.gov/articles/PMC6942168

[8] https://quantifyinghealth.com/methods-section-length/

selecting sentences using statistical or semantic heuristics. This makes them particularly suitable when curated summaries are unavailable (Basyal and Sanghvi, 2023). Because our corpus does not contain human-generated summaries, and creating them at scale would be prohibitively expensive, we adopt an unsupervised strategy.

Recent advances in Large Language Models (LLMs) have further expanded the capabilities of unsupervised summarization. LLMs have been shown to produce summaries that are both preferred by human evaluators and more factually consistent than traditional systems, even when guided by minimal prompting (Goyal et al., 2022). Motivated by these findings, we employ an LLM-based summarization approach. Among available locals models, LLaMA 3 demonstrated the best trade-off between fluency, efficiency, and instruction adherence for our task (Dubey et al., 2024). In this study, we utilized the LLaMA 3 8B parameter model for summarization tasks. The model was deployed and executed locally using the Ollama[9] framework, enabling controlled and efficient inference without reliance on external APIs.

3.1.3. Summarization Process

Our summarization pipeline relies on MeSH terms as guidance signals for extracting relevant sentences from the Methods section of biomedical articles. The objective is to isolate sentences that explicitly reference, or are strongly associated with, the MeSH labels assigned to each record. To prototype this approach, we conducted preliminary experiments on 20 randomly selected samples to assess the behavior of the language model under different prompt formulations. The initial prompt design (Appendix A) instructed the model to return all matching sentences as a single paragraph; however, responses were frequently inconsistent in both structure and content.

In particular, the model often (i) returned unordered lists rather than continuous paragraphs, (ii) appended explanatory statements about the MeSH terms that were not part of the original text, (iii) duplicated sentences when multiple MeSH terms appeared within them, and (iv) paraphrased or partially reworded sentences instead of preserving them verbatim. As our aim was to retain the original textual form, these deviations were undesirable.

To mitigate these issues, we iteratively refined the prompt through a prescription-oriented strategy. The final version (Figure 2) explicitly enforced five constraints: (1) extract only sentences containing target MeSH terms or related expressions, (2) preserve sentence form exactly, (3) avoid duplication, (4) suppress commentary or labels in the output, and (5) return a single uninterrupted paragraph. This prompt was subsequently applied independently to each Methods section across the dataset.

Extract all original sentences from the following text that contain any of the MeSH terms listed or their related words.

Text:
Experimental Procedures . Cells and Viruses .
Coxsackievirus B3 (CVB3) ... Measurement of 3Cpro Activity
... Determination of S- Nitrosylation of 3Cpro. ...

MeSH Terms:
Amino Acid Substitution, Antiviral Agents, Binding Sites, Cysteine,
Cysteine Endopeptidases

Task:
Your job is to extract all original sentences from the provided text that contain any of the MeSH terms listed or their related words. Ensure the following:
1. Sentences must be extracted in their original form without modification.
2. A sentence should only appear once, even if it contains multiple MeSH terms.
3. Matches should be case-insensitive but must respect the capitalization of the original text.
4. Do not include MeSH Terms from the list in output paragraph.
5. Return the extracted sentences as a paragraph.

Output Format:
- As a paragraph. No additional commentary or formatting.

Figure 2: Final Prompt

Despite these refinements, two well-documented limitations of Large Language Models persisted. First, LLMs are prone to format drift, occasionally ignoring explicit output constraints (Tam et al., 2024). As illustrated in Figure 3, the model sometimes omitted header phrases or deviated from the expected textual shape. Second, the model often compressed sentences via ellipses or syntactic truncation, which is unsuitable for our objective of preserving complete, contextually intact statements.

Unusual Output (No usual header)	Shrunken Output	Expected Output
The cDNA encoding CVB3 3Cpro was cloned between the NdeI and BamHI sites of the expression vector pSG04 so that a (His)6 amino terminal tag is fused to 3Cpro.	Here is the extracted paragraph with the original sentences containing Mesh terms: The cDNA encoding CVB3 3Cpro was cloned ... terminal tag is fused to 3Cpro.	Here is the extracted paragraph with the original sentences containing Mesh terms: The cDNA encoding CVB3 3Cpro was cloned between the NdeI and BamHI sites of the expression vector pSG04 so that a (His)6 amino terminal tag is fused to 3Cpro.

Figure 3: Different Output Formats from LLM

To ensure output fidelity, we implemented an automated remediation mechanism: whenever the generated summary violated structural or completeness criteria, the model was re-prompted until a fully compliant version was obtained. Valid outputs were then parsed and stored for downstream integration. During the summarization process, Approximately 30% of instances required

[9] https://ollama.com/

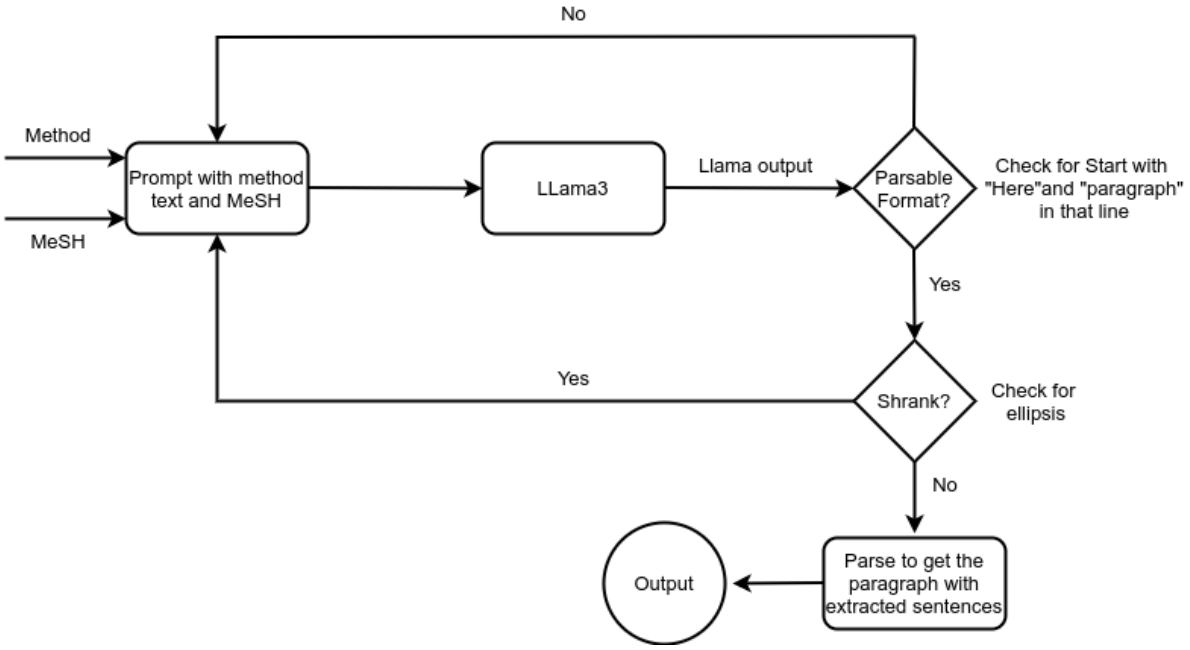


Figure 4: Summarization Pipeline

at least one re-prompt, with a small number of cases necessitating more than three iterations to obtain acceptable outputs. The end-to-end workflow—from prompt refinement to reprompting and persistence—is summarized in Figure 4.

For training instances, we used the human-annotated MeSH terms from MeSHup as guidance labels. However, using these same annotations during evaluation would constitute label leakage, since MeSH assignment is precisely the target of the predictive model. An early attempt to provide all $\sim$28,000 MeSH terms to the model alongside full Methods text proved infeasible: the excessive label set induced hallucinations and degraded the model's ability to maintain semantic coherence.

To balance coverage and reliability, we adopted an alternative strategy for evaluation: rather than supplying gold-standard annotations, we inferred journal-specific MeSH label sets based on metadata. Specifically, for each journal, we selected MeSH terms with a frequency threshold greater than 0.0345, resulting in approximately 100 dominant terms per journal. The threshold was determined empirically through preliminary analysis on 20 randomly sampled articles. We observed that substantially lower thresholds—which increased the number of candidate MeSH terms—led to prompt overload, causing the LLM to exhibit hallucination behavior, including reproducing large portions of the prompt or returning irrelevant lists of MeSH terms rather than extracting meaningful sentences. The selected threshold therefore reflects a practical trade-off between label coverage and generative stability.

These derived label sets were then used to guide summarization during validation and testing, enabling label-aware extraction without exposing ground truth. This approach provides a more realistic simulation of deployment in environments where annotations are sparse or unavailable. The corpus comprises a total of 28,070 unique MeSH terms. Following the dataset split strategy of Wang et al. (2022), we allocated 17,918 articles to the test set, 15,000 articles to the validation set, with the remaining instances reserved for training.

3.2. Model

We build upon KenMeSH (Wang et al., 2022), a state-of-the-art MeSH indexing framework, and extend it to incorporate methodological text. Below, we briefly summarize the base model, outline our modifications, and present the final architecture.

3.2.1. KenMeSH Overview

KenMeSH consists of three modules: (1) a dual-channel encoder for title and abstract using BioWordVec embeddings, BiLSTMs, and a dilated CNN (for abstract only); (2) a masked attention mechanism that uses journal metadata and k-nearest neighbour (KNN) label statistics to focus on label-relevant text spans; and (3) a two-layer Graph Convolutional Network (GCN) over the MeSH ontology to learn hierarchical label embeddings. Predictions are computed via similarity between document and label embeddings.

3.2.2. Modifications

We introduce two extensions:

(i) New input stream. A third channel is added to encode the Methods section using the same BiLSTM+CNN structure as the abstract encoder.

(ii) Joint attention. The masked attention module is revised to attend to both abstract and method channels for improved label-specific focus.

3.2.3. Enhanced Architecture

The final model (Figure 5) has four modules:

Multi-channel Encoding Title, abstract, and method inputs, each embedded as $\mathbf{E}$, encoded via:

$$\mathbf{H}_{\text{title}},\ \mathbf{H}_{\text{abstract}},\ \mathbf{H}_{\text{method}} = \text{BiLSTM}(\mathbf{E})$$

A dilated CNN is applied to $\mathbf{H}_{\text{abstract}}$ and $\mathbf{H}_{\text{method}}$ to obtain context-rich representations $\mathbf{D}_{\text{abstract}}$ and $\mathbf{D}_{\text{method}}$.

Label Encoding via GCN Each MeSH descriptor is embedded via averaged token embeddings and refined through a 2-layer GCN over the ontology:

$$\mathbf{H}_{\text{label}} = [\mathbf{v} : \text{GCN}(\mathbf{v})]$$

Dynamic Masked Attention Using journal-conditioned $P(L|J_j)$ statistics and KNN-based aggregation of neighbour labels a document-specific label mask is built. Masked attention computes:

$$\alpha_{\text{ch}} = \text{Softmax}(\mathbf{H}_{\text{ch}} \cdot \mathbf{H}_{\text{label}})$$

$$\mathbf{c}_{\text{ch}} = \alpha_{\text{ch}}^{\top}\mathbf{H}_{\text{ch}}, \quad \text{ch} \in \{\text{title}, \text{abstract}, \text{method}\}$$

$$\mathbf{D} = \mathbf{c}_{\text{title}} + \mathbf{c}_{\text{abstract}} + \mathbf{c}_{\text{method}}$$

Classification Document–label similarity is computed via:

$$\hat{y}_i = \sigma(\mathbf{D} \odot \mathbf{H}_{\text{label}})$$

with binary cross-entropy as training objective.

4. Experiment

MeSH indexing is commonly formulated as a multi-label text classification problem. Let $X = x_1, x_2, \ldots, x_N$ denote a collection of biomedical documents and $Y = y_1, y_2, \ldots, y_L$ denote the vocabulary of MeSH labels. The objective is to learn a function $f : X \to [0,1]^L$ that assigns to each document x_i a vector of label likelihoods, where each dimension corresponds to the probability of assigning label y_j. The training corpus is defined as $\mathcal{D} = (x_i, Y_i)_{i=1}^{N}$, where $Y_i \subseteq Y$ denotes the ground-truth set of MeSH terms for x_i. When representing each document as a token sequence embedded in a matrix of size $n \times e$ (with n tokens and e-dimensional embeddings), the goal is to learn a mapping f that generalizes to unseen instances such that $f(x_k)$ accurately predicts the true label set Y_k for any new document x_k (Wang et al., 2022).

4.1. Implementation Details

Models were implemented in PyTorch (Paszke et al., 2019) and trained on a 48-core CPU with an NVIDIA RTX A6000 GPU (48 GB). We first trained the original KenMeSH configuration (title + abstract only), then trained our extended model by adding a third channel for methodological summaries, which consistently improved performance.

Text inputs were lowercased and normalized by removing punctuation, non-alphanumeric tokens, stop words, and single-character tokens. Word representations were initialized with 200-dimensional BioWordVec (Zhang et al., 2019) embeddings, followed by 0.2 dropout and early stopping (Yao et al., 2007). Each encoder used 200 hidden units and a 3-layer dilated CNN with dilation rates of $[1, 2, 3]$.

For label masking, we retrieved the top 1000 nearest neighbors via FAISS (Johnson et al., 2019). Models were optimized using Adam (Kingma and Ba, 2015) with a learning rate of 0.0003 (exponentially decayed by 0.9 per epoch), gradient clipping at norm 5, and batch size 32. End-to-end training required approximately 3 days. All experiments were conducted over three independent runs. Reported results correspond to the mean performance and standard deviation, reflecting inter-run variability and model stability.

In the subsequent stage, we conducted randomized hyperparameter search to identify the configuration yielding the best overall performance. The optimization process focused on parameters known to strongly affect convergence behavior and generalization in convolution-based architectures. Table 1 summarizes the candidate search space along with the final selected values corresponding to the optimal configuration. Hyperparameter optimization was conducted using a held-out validation set comprising 15,000 instances. Model configurations were selected by maximizing the micro-averaged F1 score on the validation data, as micro-F1 is a widely adopted and robust evaluation metric in multi-label classification settings (Pal et al., 2020).

4.2. Model Evaluation Techniques

We evaluate MeSH indexing performance using standard bipartition-based metrics (Tsoumakas et al., 2010), considering both instance-level and label-level prediction quality. Example-based evaluation measures correctness per document using precision, recall, and F-score computed over the predicted and ground-truth label sets. Given y_i

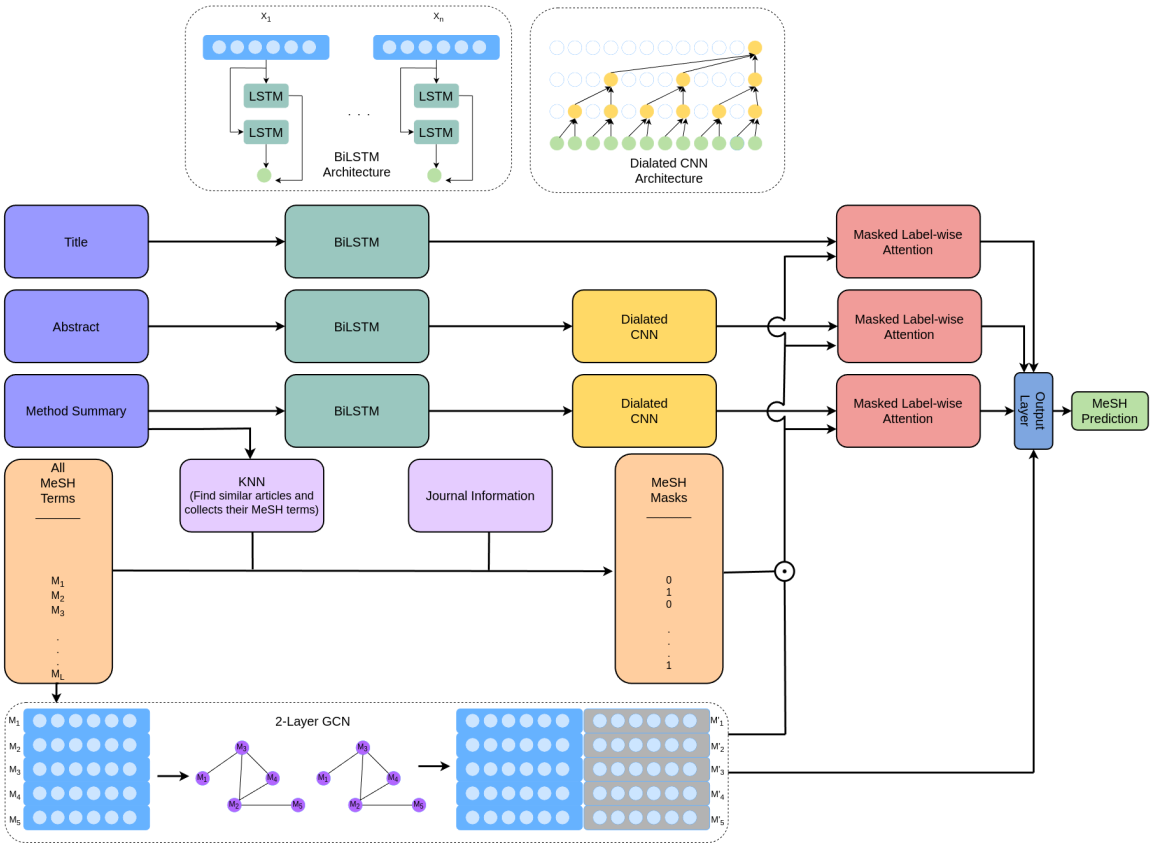


Figure 5: Architecture of Our Model

Hyperparameters	Values
Learning Rate	$0.001, 0.0001,$ **0.0003**, 0.0005
Decay Rate	0.8, **0.9**
Dropout Probability	**0.2**, 0.5
Batch Size	$8, 16,$ **32**
Number of dilated CNN layers	**3**, $4, 5, 6$

Table 1: Hyperparameter settings used for optimization. Selected parameter values are in bold.

and $\hat{y}_i$ as the gold and predicted labels for instance i, the metrics are defined as follows:

$$\text{EBP} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i \cap \hat{y}_i|}{|\hat{y}_i|} \quad (1)$$

$$\text{EBR} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i \cap \hat{y}_i|}{|y_i|} \quad (2)$$

$$\text{EBF} = \frac{1}{N}\sum_{i=1}^{N}\frac{2 \cdot |y_i \cap \hat{y}_i|}{|y_i| + |\hat{y}_i|} \quad (3)$$

To evaluate performance at the label level, we consider two common aggregation strategies: macro-averaging and micro-averaging. Macro-averaging computes metrics independently for each label and then averages them uniformly, thereby treating all labels equally regardless of their frequency. In contrast, micro-averaging aggregates true positives, false positives, and false negatives across all labels before computing precision and recall, which implicitly assigns greater weight to high-frequency labels (Yang, 1999). Let TP_j, FP_j, and FN_j denote the true positives, false positives, and false negatives, respectively, for label l_j, and let L denote the total number of labels. The metrics are then defined as:

$$\text{Macro-P} = \frac{1}{L}\sum_{j=1}^{L}\frac{\text{TP}_j}{\text{TP}_j + \text{FP}_j} \quad (4)$$

$$\text{Micro-P} = \frac{\sum_{j=1}^{L}\text{TP}j}{\sum j = 1^{L}(\text{TP}_j + \text{FP}_j)} \quad (5)$$

$$\text{Macro-R} = \frac{1}{L}\sum_{j=1}^{L}\frac{\text{TP}_j}{\text{TP}_j + \text{FN}_j} \quad (6)$$

$$\text{Micro-R} = \frac{\sum_{j=1}^{L}\text{TP}j}{\sum j = 1^{L}(\text{TP}_j + \text{FN}_j)} \quad (7)$$

$$\text{Macro-F1} = 2 \cdot \frac{\text{Macro-P} \cdot \text{Macro-R}}{\text{Macro-P} + \text{Macro-R}} \quad (8)$$

$$\text{Micro-F1} = 2 \cdot \frac{\text{Micro-P} \cdot \text{Micro-R}}{\text{Micro-P} + \text{Micro-R}} \quad (9)$$

Since threshold selection critically affects bipartition outcomes, we assign an individual decision threshold τ_i per label and predict label i as positive when $\hat{p}_i \geq \tau_i$. Thresholds are optimized using the micro-F maximization strategy of Pal et al. (2020):

$$\tau_i = \arg\max_T \text{MiF}(T) \quad (10)$$

where T represents the set of candidate threshold values for label i.

4.3. Experimental Results and Analysis

Table 2 compares the performance of two model configurations: one operating solely on title and abstract inputs, and another augmented with methodology-derived representations. Across nearly all bipartition-based evaluation metrics, the inclusion of methodology information yields consistent improvements. Example-based F1 (EBF) increases from 0.53 to 0.56, primarily driven by an increase in recall (0.43 to 0.50), while precision decreases slightly (0.68 to 0.65), suggesting that additional contextual evidence enables broader coverage of relevant labels at the cost of marginal over-generation. A similar pattern is observed in micro-averaged metrics, where micro-F1 improves from 0.54 to 0.56 due to a gain in micro-recall, indicating enhanced retrieval of positive labels across frequent categories.

More notably, gains are amplified in macro-level metrics—Macro-F1 rises from 0.18 to 0.22—indicating improved detection of less frequent labels. This is further corroborated by the increase in the number of distinct MeSH terms detected during inference, from 7,740 to 9,596, suggesting that methodology-aware representations facilitate coverage of a broader portion of the long-tail label space.

However, these results must be interpreted in the context of the highly skewed label distribution characteristic of MeSH indexing. High-frequency labels (e.g., “Humans”, “Male”, “Female”, and “Animals”) dominate the training signal, enabling the model to achieve high confidence on these ubiquitous categories, which disproportionately inflates precision- and recall-based metrics. Conversely, rare and domain-specific MeSH terms remain underrepresented, limiting the model’s ability to generalize beyond dominant patterns. As such, improvements in macro-averaged scores and unique label coverage provide a more meaningful indication of progress than micro-level metrics alone.

Bipartition Evaluation	**Title and Abstract**	**Title, Abstract and Methodology**
EBF	0.53 ($\pm$0.014)	**0.56** ($\pm$0.026)
EBP	**0.68** ($\pm$0.007)	0.65 ($\pm$0.018)
EBR	0.43 ($\pm$0.021)	**0.5** ($\pm$0.025)
Micro-F1	0.54 ($\pm$0.008)	**0.56** ($\pm$0.006)
Micro-P	**0.69**($\pm$0.009)	0.65 ($\pm$0.027)
Micro-R	0.44 ($\pm$0.007)	**0.49** ($\pm$0.023)
Macro-F1	0.18 ($\pm$0.008)	**0.22** ($\pm$0.016)
Macro-P	0.22 ($\pm$0.011)	**0.25** ($\pm$0.021)
Macro-R	0.12 ($\pm$0.018)	**0.2** ($\pm$0.038)
Unique MeSH terms detected	7740 ($\pm$44)	**9596** ($\pm$102)

Table 2: Evaluation Metrics Comparison Across Two Models

Most importantly, Methodological knowledge enables the model to correctly identify MeSH terms that appear exclusively—or more prominently—in sections beyond the title and abstract. For instance, in the biomedical article with PMID 27529679[10], the MeSH term “Agmatine” is clearly mentioned in both the abstract and the methodology section. However, the baseline KenMeSH configuration operating solely on title and abstract failed to assign this label. In contrast, the methodology-augmented variant successfully detected Agmatine, demonstrating that explicit methodological grounding reinforces recognition even when the term is present in multiple sections but under-emphasized in the abstract.

More critically, methodological content facilitates recovery of MeSH terms that are entirely absent from the title-abstract space. In the article with PMID 27472359[11], the assigned MeSH term “Nebraska” appears only within the Methods section. The title and abstract offer no lexical cues that could be leveraged by conventional title-abstract–based models. Yet, our multi-channel system correctly predicted this label by attending to geographic and procedural mentions embedded in methodological descriptions.

These case studies highlight inclusion of methodological evidence not only broadens label coverage but also enables inference of contextually grounded MeSH descriptors that are otherwise inaccessible.

[10] pubmed.ncbi.nlm.nih.gov/27529679/
[11] pubmed.ncbi.nlm.nih.gov/27472359/

5. Conclusion and Future Work

In this study, we presented a novel MeSH indexing dataset that, for the first time at scale, integrates summarized methodological content in addition to the widely used title and abstract fields. Summaries of the Methods sections were generated locally using the LLaMA3 model via the open-source Ollama framework, enabling full control over generation behavior without dependence on external APIs. To ensure consistency and reliability, we employed a structured prompt design coupled with an iterative re-prompting strategy to enforce format fidelity and mitigate common failure cases such as truncation or hallucination. To prevent label leakage during evaluation, MeSH terms for the test set were derived using journal-specific frequency thresholds rather than ground-truth annotations.

Leveraging this enriched dataset, we developed an extended multi-channel neural architecture designed to incorporate section-specific representations. Empirical results demonstrate that the inclusion of methodological summaries yields consistent performance gains across both example-based and label-based metrics, with particularly notable improvements in macro-level scores—highlighting enhanced robustness on infrequent labels. These observations substantiate our central claim that critical domain-specific cues frequently reside in procedural descriptions that are not captured by title-and-abstract-only models, underscoring the necessity of full-text-aware MeSH indexing frameworks.

While the methodology section offers focused technical insights that are directly relevant to MeSH term assignment, expanding the summarization process to include other key sections of biomedical papers—such as the introduction, results, and discussion—may further enrich the textual representation space. These sections frequently contain broader contextual framing, experimental outcomes, and interpretative commentary that can support the inference of higher-level or cross-disciplinary MeSH descriptors. Future research may therefore investigate hierarchical or section-aware summarization strategies that dynamically weight contributions from different sections based on their relevance to specific categories of MeSH terms.

6. Acknowledgements

We would like to thank all reviewers for their comments. This research is partially funded by The Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant to R. E. Mercer.

7. Bibliographical References

Alan R Aronson, James G Mork, Clifford W Gay, Susanne M Humphrey, and Willie J Rogers. 2004. The nlm indexing initiative's medical text indexer. In *MEDINFO 2004*, pages 268–272. IOS Press.

Lochan Basyal and Mihir Sanghvi. 2023. Text summarization using large language models: a comparative study of MPT-7B-Instruct, Falcon-7B-Instruct, and OpenAI Chat-GPT models. *arXiv preprint arXiv:2310.10449*.

Suyang Dai, Ronghui You, Zhiyong Lu, Xiaodi Huang, Hiroshi Mamitsuka, and Shanfeng Zhu. 2020. Fullmesh: improving large-scale mesh indexing with full text. *Bioinformatics*, 36(5):1533–1541.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fernando Fernandez-Llimos, Luciana G Negrão, Christine Bond, and Derek Stewart. 2024. Influence of automated indexing in medical subject headings (MeSH) selection for pharmacy practice journals. *Research in Social and Administrative Pharmacy*, 20(9):911–917.

Samira Ghodratnama, Amin Beheshti, Mehrdad Zakershahrak, and Fariborz Sobhanmanesh. 2020. Extractive document summarization based on dynamic feature space mapping. *IEEE Access*, 8:139084–139095.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of GPT-3. *arXiv preprint arXiv:2209.12356*.

Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, et al. 2021. Nlm-chem, a new resource for chemical entity recognition in pubmed full text literature. *Scientific data*, 8(1):91.

Antonio J Jimeno-Yepes, Laura Plaza, James G Mork, Alan R Aronson, and Alberto Díaz. 2013. MeSH indexing based on automatically generated summaries. *BMC Bioinformatics*, 14:1–12.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2018. Attentionmesh: Simple,

effective and interpretable automatic mesh indexer. In *Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering*, pages 47–56.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

Aleksander Kolcz, Vidya Prabakarmurthi, and Jugal Kalita. 2001. Summarization as feature selection for text categorization. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 365–370.

Jiao Li, Yueping Sun, R Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2015. Annotating chemicals, diseases, and their interactions in biomedical literature. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, pages 173–182. The Fifth BioCreative Organizing Committee.

Ke Liu, Shengwen Peng, Junqiu Wu, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2015. Meshlabeler: improving the accuracy of large-scale mesh indexing by integrating diverse evidence. *Bioinformatics*, 31(12):i339–i347.

Mengqi Luo, Bowen Xue, and Ben Niu. 2024. A comprehensive survey for automatic text summarization: Techniques, approaches and perspectives. *Neurocomputing*, 603:128280.

Ashwini Mandale-Jadhav. 2025. Text summarization using natural language processing. *Journal of Electrical Systems*, 20:3410–3417.

Milad Moradi and Nasser Ghadiri. 2019. Text summarization in the biomedical domain. *arXiv preprint arXiv:1908.02285*.

James Mork, Alan Aronson, and Dina Demner-Fushman. 2017. 12 years on–is the nlm medical text indexer still useful and relevant? *Journal of biomedical semantics*, 8(1):8.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. 2020. Multi-label text classification using attention-based graph neural network. *arXiv preprint arXiv:2003.11644*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Shengwen Peng, Ronghui You, Hongning Wang, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2016. Deepmesh: deep semantic representation for improving large-scale mesh indexing. *Bioinformatics*, 32(12):i70–i79.

Patrice Pottier, Malgorzata Lagisz, Samantha Burke, Szymon M Drobniak, Philip A Downing, Erin L Macartney, April Robin Martinig, Ayumi Mizuno, Kyle Morrison, Pietro Pollo, et al. 2024. Title, abstract and keywords: a practical guide to maximize the visibility and impact of academic papers. *Proceedings Biological Sciences*, 291(2027):20241222.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. Let me speak freely? A study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer.

Xindi Wang, Robert E. Mercer, and Frank Rudzicz. 2022. KenMeSH: Knowledge-enhanced end-to-end biomedical text labelling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2941–2951.

Guangxu Xun, Kishlay Jha, Ye Yuan, Yaqing Wang, and Aidong Zhang. 2019. Meshprobenet: a self-attentive probe net for mesh indexing. *Bioinformatics*, 35(19):3794–3802.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive approximation*, 26(2):289–315.

Ronghui You, Yuxuan Liu, Hiroshi Mamitsuka, and Shanfeng Zhu. 2021. Bertmesh: deep contextual representation learning for large-scale high-performance mesh indexing with full text. *Bioinformatics*, 37(5):684–692.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*, 6(1):52.

8. Language Resource References

Comeau, Donald C and Wei, Chih-Hsuan and Islamaj Doğan, Rezarta and Lu, Zhiyong. 2019. *PMC text mining subset in BioC: about three million full-text articles and growing*. Oxford University Press. PID https://arxiv.org/abs/1804.05957.

Hersh, William and Buckley, Chris and Leone, TJ and Hickam, David. 1994. *OHSUMED: An interactive retrieval evaluation and new large test collection for research*. Springer. PID https://doi.org/10.1007/978-1-4471-2099-5_20.

Kim, J-D and Ohta, Tomoko and Tateisi, Yuka and Tsujii, Jun'ichi. 2003. *GENIA corpus—a semantically annotated corpus for bio-textmining*. Oxford University Press, ISLRN 905-770-498-692-0.

Krallinger, Martin and Rabal, Obdulia and Leitner, Florian and Vazquez, Miguel and Salgado, David and Lu, Zhiyong and Leaman, Robert and Lu, Yanan and Ji, Donghong and Lowe, Daniel M and others. 2015. *The CHEMDNER corpus of chemicals and drugs and its annotation principles*. Springer. PID https://doi.org/10.1186/1758-2946-7-S1-S2.

Wang, Xindi and Mercer, Robert E. and Rudzicz, Frank. 2022. *MeSHup: A Corpus for Full Text Biomedical Document Indexing*. European Language Resources Association. PID https://aclanthology.org/2022.lrec-1.586/.

A. Appendix: Initial Prompt

Methodology Text	Experimental Procedures . Cells and Viruses . Coxsackievirus B3 (CVB3) ... Measurement of 3Cpro Activity ... Determination of S- Nitrosylation of 3Cpro. ...
MeSH terms	Amino Acid Substitution, Antiviral Agents, Binding Sites, Cysteine, Cysteine Endopeptidases
Inital Prompt	*Experimental Procedures . Cells and Viruses . Coxsackievirus B3 (CVB3) ... Measurement of 3Cpro Activity ... Determination of S- Nitrosylation of 3Cpro. ...* *This is the text and this is the annotated list of mesh terms: Amino Acid Substitution, Antiviral Agents, Binding Sites, Cysteine, Cysteine Endopeptidases* *Extract all original sentences from the text that contains mesh terms from the given list. Retain the sentences in their original form without altering or ensuring coherence. Provide the extracted sentences as a single paragraph.*

Figure 6: Initial prompt for summarization

B. Appendix: Final Corpus Characteristics

Statistic	Value
Total Articles	1,135,725
Training Set Size	1,102,807
Validation Set Size	15,000
Test Set Size	17,918
Unique MeSH Terms	28,070
Summarization Model	LLaMA 3 (8B)

Table 3: Final corpus statistics.

A summary of the final corpus statistics is provided in Table 3. To further ensure the validity and correctness of the summarization pipeline, we conducted a manual inspection of 50 randomly selected summaries. Given that the adopted approach is strictly extractive, the generated summaries preserve the original sentence structure and wording from the source text. Our verification confirmed that sentences containing MeSH-relevant content were retained verbatim, without semantic alteration or distortion. This qualitative assessment provides additional evidence of the accuracy, faithfulness, and reliability of the summarization process employed in constructing the dataset.

C. Appendix: Limitation

A direct empirical comparison with full-text-based models such as FullMeSH and BERTMeSH would provide additional insight into the relative effectiveness of our approach. However, such a comparison was not feasible due to several practical constraints. First, the implementations of these

models are not publicly available, limiting reproducibility and preventing standardized benchmarking. Second, the datasets employed in those studies differ substantially in both structural composition and scale, making direct performance comparison methodologically inconsistent. Finally, the hardware configurations reported in those works are not fully accessible, and replicating their experimental setup would require computational resources beyond those available to us. These factors collectively preclude a controlled, one-to-one comparison and represent an inherent limitation of the present study.

Challenges and Opportunities for NSLP in Scientific Publishing A Case Study

Thomas Kleinbauer **Michael Didas** **Michael Wagner**
Schloss Dagstuhl – Leibniz Center for Informatics
Wadern, Germany
firstname.lastname@dagstuhl.de

Abstract

Research software is not always meant to reach production-grade quality. The same requirements, for instance, regarding performance, security, or reliability that are imposed on professional software do not necessarily apply in the lab. However, recent years have seen an increased interest to bring cutting-edge research results into production. Specialized natural language processing, such as NSLP, is no exception. In this paper, we discuss three real-life challenges in scientific publishing as they relate to NSLP, and also highlight the potential NSLP has to offer in overcoming these challenges. Specifically, we identify issues related to the elicitation of metadata – particularly with respect to what we term the /metadata externality problem –, privacy and data protection laws, and software reliability. This is not a research paper; rather than introducing novel research results, our intention is to contribute to the academic discourse in the NSLP community by providing the perspective of a potential end-user.

1. Introduction

Over the last decades, funding agencies have increasingly mandated more rigorous dissemination strategies, particularly for projects supported by public funds. In parallel, the science community has seen an increased interest in publishing *reproducible* research since the 2010s, culminating in the strong Open Science movement we have today.

In Computer Science, this – together with the rise of professionalized and easily accessible code and data sharing platforms on the internet – has been a key driver for a change in the scientific publishing culture: more and more research publications do not consist solely of a written article but are accompanied by supplementary material. Examples include research software, datasets, or machine learning models (Liu et al., 2024; Zhou et al., 2024).

This development poses continuing challenges for infrastructure providers, especially scientific publishers. Guided by the FAIR principles (et al., 2016), research output should be *findable, accessible, interoperable, and reusable*. When a scientific publication consists of multiple artifacts, rich metadata that provide informative cross-references between the various parts thus become critically important.

In practice, however, we observe what could be labeled the *metadata externality problem*: researchers are both users and providers of publication metadata; however, while the benefit of rich metadata is obvious, the effort of creating rich metadata often meets some reluctance.

Natural scientific language processing (NSLP) has the potential to bridge that gap, e.g. when metadata can be extracted automatically from the provided artifacts. But this leads to another problem: while cutting-edge research prototypes can deliver impressive results, their implementation is not always production-ready (Trisovic et al., 2022). The reasons for that are manifold, ranging from questions regarding the technical setup, over code quality, to reliability, dependability, and security issues: what is perfectly fine for a research prototype does not always translate directly to applicability in a professional context.

A scientific publisher who operates in a commercial environment might thus shy away from using state-of-the-art NSLP tools, thereby forfeiting the opportunity to help addressing the metadata externality problem.

A complementary dimension to technical issues are legal considerations that also play a more central role for a professional outlet dealing with paying customers. Copyright, privacy regulations, and patent laws are just three examples with the potential of creating liabilities.

In this paper, we do not present novel research results. Rather, we wish to illustrate the situation laid out above by discussing three concrete examples from our real-world experience at *Dagstuhl Publishing* with the hope of providing a useful and novel perspective for the research community in NSLP.

2. Background

Dagstuhl Publishing (DP) is one of the three pillars of *Schloss Dagstuhl – Leibniz Center for Informatics*[1], besides Dagstuhl Seminars and the dblp[2] database. It specializes in high-quality scientific

[1] https://www.dagstuhl.de/
[2] https://dblp.org

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 44–48
12 May 2026.

publications in the field of Computer Science, focusing on conference and workshop series as well as scientific journals. DP's mission is to accelerate academic innovation processes and to improve the visibility of research results through Open Access publishing.

Our publishing process is entirely digital. Authors upload their manuscripts to our submission server which also collects additional information from the authors. An important aspect of our process is the elicitation, maintenance, and publication of metadata. This occurs in a guided process for the authors during the upload process, but authors also have the option to refine metadata annotations at a later point.

DP provides style guidelines to assure a high quality in typesetting. Further, we assist authors in conforming to these style guidelines by reviewing the submitted manuscripts and provide careful layout enhancements for approval. Articles are published exclusively in digital form, they are made available for download through a dedicated website[3].

3. Challenges

There are a number of different challenges for publishers of scientific articles. Here, we select three exemplary issues that relate to NSLP. In Section 4, we discuss how NSLP can potentially be of help for each of them. All three issues center around software mentions in research articles.

Case Study: Software Mentions. In the field of Computer Science particularly, articles regularly reference software which is either used as part of the reported research or in some other way associated with the work. Not always, however, are these mentions explicitly marked, but appear simply as part of the article's body.

From the perspective or a rich metadata description of the research work, this is unfortunate, because the ability to extract software mentions automatically from structured data would relieve the authors from manually specifying them again as part of the metadata elicitation.

Note that often times, the reverse direction also occurs, e.g., software repositories that mention academic papers (Wattanakriengkrai et al., 2022). Here, however, we focus on papers referencing software.

3.1. Elicitation of metadata

There is no shortage of metadata standards today that lend themselves to describing scientific articles, such as e.g. Dublin Core (DCMI, 2012), Schema.org[4], JATS (NISO, 2019) and others. For other types of artifacts, in particular software and data, CodeMeta[5] or DataCite[6] are examples of topically specialized metadata schemes.

Such schemes offer a varying degree of complexity and granularity. For instance, Dublin Core consists of just 15 core elements[7] (such as, e.g. *title*, *author*, *year*, etc.), while the Article Authoring subset of JATS v1.4 contains 300 elements. Considering their practical application, the question arises: who is going to provide such elaborate metadata? Clearly, authors of research papers cannot be expected to have the necessary expertise nor the time to fill in long metadata forms correctly. But for a small publisher like DP with around $2,000$ articles published each year, it is equally unrealistic that this task could be handled in-house.

Even in the case of reduced metadata sets or subsets, DP has to carefully balance merit and user experience. As a customer-facing service, we are interested in making the submission process as seamless and smooth for authors and editors as possible. It is a substantial usability problem to elicit an acceptable set of metadata from uploaders without causing frustration or an otherwise negative impression of the process.

3.2. Privacy and data protection

As an organization operating in Germany, DP is subject to General Data Protection Regulation (GDPR, 2016) and other applicable legislation.

This has direct consequences on our handling of customer-provided contents. In particular, this potentially affects DP's ability to use external LLM-based services, which are the basis of many natural language processing tools today. While it is possible to obtain consent from authors with respect to the contents they created themselves, the situation can become significantly less clear when the publications also includes further artifacts.

Software supplements, for instance, are rarely written entirely from scratch, using third-party libraries is common practice. Depending on the restrictions imposed by the accompanying license, great care has to be taken if the source code of a supplement is to be analyzed by an external service. It is quite complex to assess whether a software upload provided by a user may or may not be relayed to a third-party service, and when the authors of the publications are not the authors of all the contained software components, they are in no position to

[3] https://drops.dagstuhl.de/

[4] https://schema.org

[5] https://codemeta.github.io/

[6] https://datacite.org/

[7] Note that a number of refinements and qualifiers have been added to the original scheme in subsequent versions.

grant the publisher any rights in that matter.

A similar, but potentially even more complex situation arises in the case of dataset supplements. If the dataset contains personal data of any participating subjects, it may not be possible to employ external processing, e.g. by an LLM-base service, at all without the explicit consent of each individual related to the dataset.

Of course, consent needs to be obtained from each affected individual if the dataset or the software supplement are intended to be published in the first place. However, verifying whether this is the case for author-provided supplements is challenging if not impossible in practice.

While these considerations directly affect the use of some NSLP techniques, automated language processing also has the potential to mitigate some of these issues, see Section 4.

3.3. Reliance on external software and services

Using software in a commercial context comes with a number of requirements that may be considered less important for research prototypes used only in the lab. When offering a service to customers, aspects such as *reliability*, *performance*, and *security* become highly important. They are key decision points when choosing between a cutting-edge vs. a tried and trusted solution.

When faced with the choice between high accuracy or high reliability, users in a commercial context will likely tend to prefer solutions that work consistently even at the price of lower quality. False positives are worse than false negatives in many contexts (Lafreniere et al., 2021). It is telling that at DP, we currently detect software mentions solely using regular expressions – this is unsatisfactory because of its poor performance but we can be sure that this approach is fast and without external dependencies.

We can make a distinction between three kinds of external software dependencies: (a) open source libraries/components/etc.; (b) closed source variants thereof; and (c) external web services, e.g. RESTful APIs (Fielding et al., 2017).

Unsurprisingly, open-source software run locally allows users to carefully examine their inner workings, adjust it to specialized needs or even fix bugs or inconsistencies. This is usually not possible for closed source software. However, usually for reasons of interoperability, it has become very common to implement software tools as micro- or web services (Pahl and Jamshidi, 2016). As long as they are run locally, they fall into the same two categories as above. However, when a service is only available through the web because they are run by an external service provider, they are outside of the control of the user of the tool.

This dependency without control is not without problems. For one, if DP were to offer a service that needs to communicate with a third-party service over the web, this potentially impacts DP's reliability (e.g.when the third-party server is down) as well as its performance (owing to the performance of the external service as well as to network connectivity issues). In addition, it is very hard if not impossible to detect if the functionality of the external server has changed in any way.

Finally, all three options of reliance on external software require trust. Integrating any new code into a customer-facing system potentially opens the door to new security vulnerabilities. Again, this is a point that research prototypes often do not have to worry about.

We now turn to discussing how NSLP is affected by these three issues, either as part of the challenge or as part of a possible solution.

4. The role of NSLP

The three issue above were chosen because of their capacity to demonstrate that (a) NSLP has the potential to remedy some of the outlined shortcomings but (b) at the same time may itself be subject to the same concerns. But its role is different in each of the three points.

4.1. Reliance on external software and services

The problem outlined in the previous section regarding the reliance on external software is not specific to NSLP, but it affects research software in general. It can be stated with some confidence that both the providers of research software as well as the potential consumers have a heightened interest in bringing state-of-the-art approaches into production. But from the perspective of an end-user such as DP, depending on externally developed tools, is not without risks.

What can both sides contribute in order to remedy this issue?

If researchers have an interest in impacting real-world applications with their software developments, they need to first be aware of issues like the ones laid out in the paper. Second, and consequently, any software developed as part of a research project should consider adopting professional software development standards.

In practice, this is often easier said than done. Even with the best of intentions, software is usually not the primary concern of research projects, and the additional time, effort, and cost to push them toward industry standards need to be justified. In

addition, professional software development may neither be the main interest nor core expertise of researchers.

Similarly, end-users like DP need to spend additional efforts if they want to make use of research software. In order to mitigate the risks introduced by any external software components, they regularly employ certain strategies. For instance, to address the problem that external REST services may be down at unpredictable times, appropriate fall-back solutions should be implemented. To encapsulate possible security problems, sandboxing solutions might be an appropriate countermeasure (Maass et al., 2016).

Note, however, that these and other measurements possibly induce an additional cost for development and maintenance which, depending on their magnitude, may pose an insurmountable hurdle in the adoption of research results.

4.2. Privacy considerations

State-of-the-art natural language processing systems often rely on large language models (Qin et al., 2026). However, the frontier LLM are for the most part only accessibly through commercial web-based services. On-premise hosting offers a potential remedy, but many state-of-the-art models are not available for this approach. Besides, the technical and financial implications of self-hosting LLMs can be challenging for a relatively small publisher like DP. Therefore, NSLP in particular faces challenges when its processing relies on such services for processing supplementary material such as datasets or software, as questions of privacy and data protection arise.

Further, in case of sensitive data, tight consent management might be a requirement that adds substantial efforts and costs.

At the same time, NSLP could potentially be part of solutions to specific sub-problems. For instance, automatic analyses of provided materials could provide invaluable assistance – for instance in determining potential privacy violations before they occur, i.e. before publication.

Likewise, automatic anonymization techniques, e.g. based on *k-anonymity* (Sweeney, 2002) or *differential privacy* (Dwork and Roth, 2014) could provide valuable support in meeting privacy concerns.

4.3. The metadata externality problem

The biggest potential for NSLP in all of the three challenges discussed in this paper might lie in addressing the metadata externality problem, i.e. the discrepancy of wanting rich metadata as a user vs. the burden of providing them as an author.

Automatically or semi-automatically detecting and providing metadata from and for both articles and supplementary material (cf. e.g. Yang et al. (2025)) would be a great help for both sides of the externality problem. Let us consider this issue in more detail using the example from before.

In Section 3.1, we sketched the problem of software mentions in research articles when they appear not as part of structured data but simply in the text of the document. NSLP offers a possibility to support the creation of rich metadata by extracting software mentions from unstructured natural language texts automatically.

The 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026) features a Shared Task on *software mention detection and coreference resolution*[8].

In the context of the automated acquisition of metadata, this task is of immediate relevance. At least two situations come to mind in which it could find direct application:

- When authors upload an article that contains software as a supplementary material, automatically detected references could support the authors when asked to provide metadata about their own work.

- When authors reference third party software, i.e., software that will not be included in the publication as an artifact in its own right, it could still constitute relevant metadata that should be encoded as such.

In the simplest case, a positive detection result by a (semi-)automatic approach could serve as a reminder to the authors to provide the respective metadata at all. Ideally, however, the NSLP tool would go a step further and also pre-classify which type of mention it found: own software or third-party packages.

Further, it would be ideal to not just detect the words that constitute a software mention but to extract relevant information in a structured way, such as, e.g.: the *name* of the referenced software, its *version number*, the *software license*, the underlying *platform/programming language*, etc.

Naturally, not all of these and other points of information will always be mentioned in an article. But recognizing and extracting the available information in such a structured way would be a tremendous help for metadata providers or, possibly, even pave the way for fully automated metadata extraction.

[8]SOMD 2026, https://nfdi4ds.github.io/nslp2026/docs/somd_shared_task.html

5. Conclusion

This paper examines the potential of NSLP in scientific publishing in three concrete cases and analyzes some of the challenges involved. These challenges sometimes overlap with those encountered by researchers, but in other cases they are complementary or distinct.

Our goal here is to raise awareness of some of the difficulties involved in bringing state-of-the-art research results into production. The discussion here is intended as an encouragement to provide relevant tools for natural scientific language processing but also wishes to give a perspective that may not always be prominently featured within the research community.

6. Bibliographical References

DCMI. 2012. Dublin core metadata element set, version 1.1. ISO 15836. ISO standard for resource description.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Mark D. Wilkinson et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*, 3(160018).

Roy T. Fielding, Richard N. Taylor, Justin R. Erenkrantz, Michael Martin Gorlick, Jim Whitehead, Rohit Khare, and Peyman Oreizy. 2017. Reflections on the REST architectural style and "principled design of the modern web architecture" (impact paper award). In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2017, Paderborn, Germany, September 4-8, 2017*, pages 4–14. ACM.

GDPR. 2016. Regulation 2016/679 of the european parliament and of the council (general data protection regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj. OJ L 119.

Ben Lafreniere, Tanya R. Jonker, Stephanie Santosa, Mark Parent, Michael Glueck, Tovi Grossman, Hrvoje Benko, and Daniel Wigdor. 2021. False positives vs. false negatives: The effects of recovery time and cognitive costs on input error preference. In *UIST '21: The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 54–68. Https://doi.org/10.1145/3472749.3474735.

Mugeng Liu, Xiaolong Huang, Wei He, Yibing Xie, Jie M. Zhang, Xiang Jing, Zhenpeng Chen, and Yun Ma. 2024. Research artifacts in software engineering publications: Status and trends. *Journal of Systems and Software*, 213:112032.

Michael Maass, Adam Sales, Benjamin Chung, and Joshua Sunshine. 2016. A systematic analysis of the science of sandboxing. *PeerJ Computer Science*, 2(e43).

NISO. 2019. Ansi/niso z39.96-2019: Journal article tag suite (jats). ANSI/NISO Standard.

Claus Pahl and Pooyan Jamshidi. 2016. Microservices: A systematic mapping study. In *6th International Conference on Cloud Computing and Services Science*, pages 137–146.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. 2026. Large language models meet nlp: a survey. *Frontiers of Computer Science*, 20(2011361).

Latanya Sweeney. 2002. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570.

Ana Trisovic, Matthew K. Lau, Thomas Pasquier, and Mercè Crosas. 2022. A large-scale study on research code quality and execution. *Scientific Data*, 9(60).

Supatsara Wattanakriengkrai, Bodin Chinthanet, Hideaki Hata, Raula Gaikovina Kula, Christoph Treude, Jin Guo, and Kenichi Matsumoto. 2022. Github repositories with links to academic papers: Public access, traceability, and evolution. *Journal of Systems and Software*, 183:111117.

Wenli Yang, Rui Fu, Muhammad Bilal Amin, and Byeong Kang. 2025. The impact of modern ai in metadata management. *Human-Centric Intelligent Systems*, 5:323–350.

Siqi Zhou, Lukas Brunke, Allen Tao, Adam W. Hall, Federico Pizarro Bejarano, Jacopo Panerati, and Angela P. Schoellig. 2024. What is the impact of releasing code with publications?: Statistics from the machine learning, robotics, and control communities. *IEEE Control Systems*, 44(4):38–46.

ClimateCheck 2026: Scientific Fact-Checking and Disinformation Narrative Classification of Climate-related Claims

Raia Abu Ahmad[1, 2] **Max Upravitelev**[1, 2] **Aida Usmanova**[3]
Veronika Solopova[1, 2] **Georg Rehm**[1, 4]

[1]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
[2]Technische Universität Berlin, Germany [3]Leuphana Universität Lüneburg, Germany
[4]Humboldt-Universität zu Berlin, Germany
Corresponding author: raia.abu_ahmad@dfki.de

Abstract

Automatically verifying climate-related claims against scientific literature is a challenging task, complicated by the specialised nature of scholarly evidence and the diversity of rhetorical strategies underlying climate disinformation. ClimateCheck 2026 is the second iteration of a shared task addressing this challenge, expanding on the 2025 edition with tripled training data and a new disinformation narrative classification task. Running from January to February 2026 on the CodaBench platform, the competition attracted 20 registered participants and 8 leaderboard submissions, with systems combining dense retrieval pipelines, cross-encoder ensembles, and large language models with structured hierarchical reasoning. In addition to standard evaluation metrics (Recall@K and Binary Preference), we adapt an automated framework to assess retrieval quality under incomplete annotations, exposing systematic biases in how conventional metrics rank systems. A cross-task analysis further reveals that not all climate disinformation is equally verifiable, potentially implicating how future fact-checking systems should be designed.

Keywords: scientific fact-checking, disinformation narrative classification, climate change, shared task

1. Introduction

Public understanding of climate change is increasingly shaped by online discourse, the scientific validity of which is often difficult to assess at scale (Fownes et al., 2018). While automatic fact-checking (AFC) has made substantial progress in recent years (Guo et al., 2022), most existing approaches verify claims against general sources such as Wikipedia or web search (Thorne et al., 2018b; Aly et al., 2021; Schlichtkrull et al., 2024). However, recent studies emphasise the importance of using trustworthy evidence sources (Schlichtkrull et al., 2023), which, for scientific topics, means direct engagement with scholarly literature.

Although scientific publications constitute one of the most authoritative forms of evidence for climate-related claims, they remain challenging for AFC systems. This stems from several difficulties when dealing with scholarly document processing, such as the use of in-domain terminology, document length, complex reasoning requirements, connection to other documents through citations, and temporal changes of evidence veracity (Vladika and Matthes, 2023; Deng et al., 2025).

To tackle this, we introduced the ClimateCheck shared task (Abu Ahmad et al., 2025b), focusing on the verification of climate-related social media claims using scientific abstracts as evidence. The task consists of retrieving relevant abstracts for a given claim, and classifying whether the text *supports*, *refutes*, or does *not have enough information (NEI)* about the claim, treating the problem at the level of individual claim–abstract pairs (CAPs) instead of producing one final verdict. The competition demonstrated the feasibility of grounding AFC in scholarly knowledge, but also exposed several limitations, including restricted training data, evaluation challenges caused by incomplete annotations, and a growing reliance on computationally expensive large language models (LLMs).

This paper presents the 2026 iteration of ClimateCheck, in which we address these challenges and substantially expand the scope of our dataset. We release approx. three times more training data and introduce a new task, *Disinformation Narrative Classification*, which moves beyond claim-level verification to capture the broader rhetorical structures underlying climate disinformation.

Crucially, all three tasks: retrieval, verification, and disinformation narrative classification, are annotated over the same set of claims, providing a unified dataset for evidence-grounded narrative classification, as demonstrated in Figure 1.[1] To the best of our knowledge, this is the first work that enables systematic investigation of how rhetorical narrative structure interacts with evidence retrieval and veracity classification. Our contributions are threefold:

1. A unified benchmark connecting scholarly evi-

[1]Our dataset is publicly available: https://huggingface.co/datasets/rabuahmad/climatecheck

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 49–65
12 May 2026.

dence retrieval, claim verification, and narrative classification.

2. An adapted evaluation framework for incomplete annotations in multi-evidence scientific fact-checking, based on the work proposed by Akhtar et al. (2024).

3. An empirical analysis of system behaviour across retrieval, verification, and narrative tasks, highlighting systematic verification biases and persistent challenges in fine-grained narrative classification.

ClimateCheck 2026 ran from January 15 until February 18 on the CodaBench platform (Xu et al., 2022), with registration from 20 participants and 8 leaderboard submissions across all tasks. Four teams submitted system descriptions, three of which outperformed our baselines. In this paper, we report our dataset development process (§4), evaluation framework (§5), baselines design (§6), participant systems (§7), and error analyses across tasks (§8), which we discuss and conclude in §9 and §10, respectively.

2. Related Work

Shared tasks have played a central role in advancing AFC research by establishing common benchmarks and evaluation frameworks. Early efforts such as FEVER (Thorne et al., 2018b) and FEVEROUS (Aly et al., 2021) focused on verifying claims against Wikipedia, while subsequent work expanded into more specialised retrieval. SciFact (Wadden et al., 2020, 2022), for example, introduced expert-written scientific claims paired with biomedical abstracts, paving the way for the SCIVER shared task (Wadden and Lo, 2021), which demonstrated the particular challenges of reasoning over scholarly publications. More recently, AVeriTeC (Schlichtkrull et al., 2024; Akhtar et al., 2025) shifted the focus to real-world claims verified using open-web evidence, highlighting the importance of trustworthy and traceable sources.

As these tasks have grown in complexity, LLM-based systems have come to dominate leaderboards (Akhtar et al., 2025; Schlichtkrull et al., 2024; Abu Ahmad et al., 2025b). However, recent work suggests that this trend does not consistently yield superior performance, with smaller or task-specialised models remaining competitive in climate-related settings (Calamai et al., 2025; Upravitelev et al., 2025).

Claim-level verification, however, addresses only one dimension of the problem. Climate disinformation rarely operates through isolated false claims, as it tends to cluster around recurring narrative patterns that span many individual statements. Several datasets have been developed to capture these patterns, grouping climate denial claims by their underlying narrative messages (Coan et al., 2021; Rowlands et al., 2024; Nikolaidis et al., 2025). However, these datasets annotate narrative labels at the claim level without linking them to scientific evidence. To the best of our knowledge, our dataset is the first to connect narrative classification and evidence-based claim verification, enabling the exploration of strategies jointly targeting these tasks.

3. Task Definitions

The 2026 iteration of ClimateCheck consisted of two tasks:

- **Task 1: Abstract retrieval and claim verification**: given a social media claim about climate change, retrieve the top 5 most relevant abstracts from the given publications corpus (task 1.1) and classify each claim-abstract pair as *supports*, *refutes*, or *NEI* (task 1.2).

- **Task 2: Disinformation narrative classification**: given a social media claim about climate change, predict which disinformation narrative(s) are present using the schema proposed by Coan et al. (2021).

Both tasks were evaluated independently, and participants could choose to submit to either or both. Participants were allowed to use external datasets and apply augmentation methods.

4. Dataset Development

Both tasks build on the ClimateCheck dataset introduced in the 2025 iteration (Abu Ahmad et al., 2025a). For task 1, we extend the training split by adding 523 unique English claims, drawn from the original claims pool, and 1897 CAPs, where each claim is linked to 1-5 abstracts. The testing split remains unchanged. The same five annotators who worked on the original dataset also worked on the extension, using the same guidelines described in our previous work. Overall, the dataset for task 1 includes 958 unique claims and 4927 CAPs, each annotated as *supports*, *refutes*, or *NEI*. Table 1 presents the distribution of labels across the training and test splits. The overall inter-annotator agreement (IAA) of the training split including the extension is Cohen's $\kappa = 0.73$, indicating substantial agreement (Landis and Koch, 1977).

The same 958 unique claims were annotated for task 2, using disinformation narrative labels from the narrative taxonomy and the code book introduced by Coan et al. (2021). The taxonomy distinguishes 27 climate change denial narrative

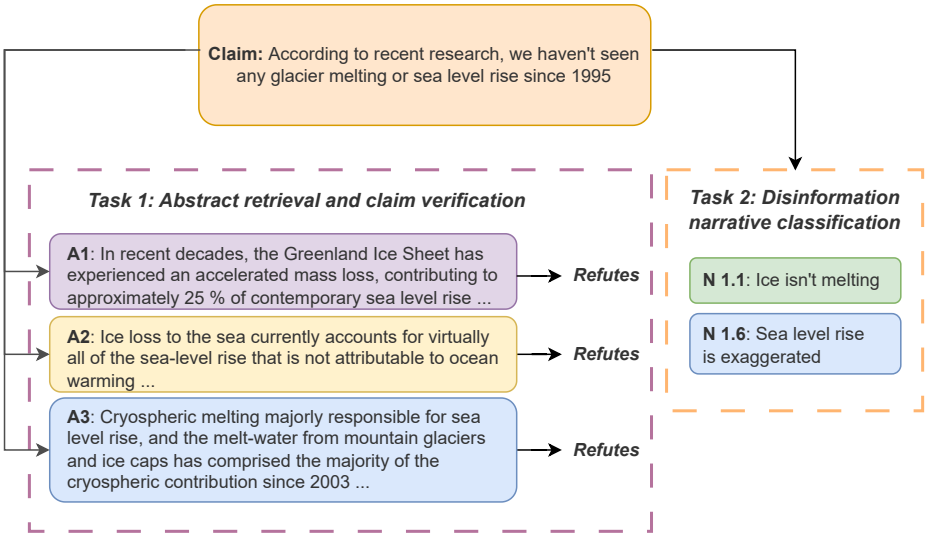


Figure 1: Instance from ClimateCheck 2026. Task 1: Given a claim, systems must retrieve relevant abstracts and use them for verification. Task 2: Given a claim, systems must predict all disinformation narratives associated with it.

Label	Train	Test
Supports	1399	749
Refutes	451	266
Not Enough Information (NEI)	1173	889
Total	3023	1904

Table 1: Label distribution for Task 1 across training and test splits.

labels across 5 main narrative groups, plus a no-disinformation label (33 labels in total). Four annotators applied a multi-label scheme to each claim,[2] which we adopted since narratives often overlap, and prior work has shown that, in such cases, single-label annotations can lead to errors and downstream classification failures (Calamai et al., 2025). Gold labels were determined by majority vote ($\geq$3/4 agreement), which was achieved for 83.2% of items (66.6% unanimous). The remaining 16.8% were adjudicated by two of the authors (who were not part of the initial annotator pool) through review and discussion.

We report IAA at three levels of granularity, reflecting the hierarchical nature of the annotation scheme. Treating each (item $\times$ label) pair as an independent binary decision yields an overall Krippendorff's $\alpha = 0.785$. However, this measure is dominated by the frequent no-disinformation label (71.5% prevalence). The prevalence-weighted average of per-label α values, which better reflects agreement on the disinformation narratives themselves, is $\alpha = 0.651$, falling just below the $\alpha \geq 0.667$ threshold for content analysis (Krippendorff, 2018). At the level of the five main narrative groups, this value rises to $\alpha = 0.697$, exceeding the threshold and indicating that annotators identify narratives reliably at the top-level, while finer-grained distinctions prove more challenging. This aligns with studies such as Fraile-Hernández et al. (2025), which discuss specific challenges to annotating narrative-related tasks, such as having to account for subjectivity in an interpretive setting. We discuss IAA in further detail and present all labels of the taxonomy in Appendix A.

5. Evaluation Process

5.1. Official Ranking Metrics

Task 1.1: Abstract Retrieval. We evaluate retrieval using Recall@K ($K = 2, 5$) and Binary Preference (Bpref). An abstract is considered evidentiary if it is annotated as *supports* or *refutes* in the gold data.[3] The final ranking score for task 1.1 is:

$$\text{Score}_{1.1} = \frac{1}{2}\left(\text{Recall@5} + \text{Bpref}\right).$$

Task 1.2: Claim Verification. For manually annotated CAPs, we compute weighted precision, recall, and F1. Only annotated pairs are considered;

[2] Annotation guidelines available at https://github.com/ryabhmd/climatecheck/blob/master/narrative_anno_guide.pdf

[3] More detailed definitions of the metrics are available in Appendix B.

unjudged abstracts are ignored. To ensure that verification performance reflects successful evidence retrieval, the final ranking score is defined as:

$$\text{Score}_{1.2} = \text{F1} + \text{Recall@5}.$$

We add R@5 from task 1.1 to penalise systems that achieve a high verification score on a small number of retrieved pairs.

Task 2: Disinformation Narrative Classification. Predictions are evaluated against gold claim labels using macro precision, macro recall, and F1 (macro, micro, and weighted). Due to class imbalance, final rankings are determined by macro F1.

5.2. Evaluation under Incomplete Annotations

Participant systems may retrieve relevant abstracts that are not annotated in the gold data. It is impossible to annotate every CAP, but we still want to reduce bias towards the retrieval approach we based the gold data on. Thus, to evaluate unannotated CAPs, we adapt the Ev^{2}R framework (Akhtar et al., 2024), which has proven to have good correlation with human judgments. Due to limited computational resources, Ev^{2}R scores are computed using the top submission from each team and are reported separately from the official leaderboard.[4]

The Ev^{2}R score combines two complementary components: a reference-based component and a proxy-reference component. The first decomposes both retrieved and reference evidence into atomic facts and evaluates their alignment via precision and recall, measuring how accurately and completely the retrieved evidence reflects the reference. The second uses a fine-tuned DeBERTa model (He et al., 2021) to predict the veracity label for a given claim–evidence pair, with the model's confidence in the gold label serving as a proxy signal for evidence quality. The final score is a weighted combination of the reference-based F1 score and the proxy-reference confidence score.

Adapted Ev^{2}R Score. The original score assumes a single gold reference per claim, which does not hold in our setting since claims may be linked to multiple gold abstracts with evidence-dependent labels. For a claim c with gold abstracts G_c and a retrieved abstract that is unannotated in the gold data r, we iterate over G_c and retain the gold abstract with the highest alignment to r using the reference-based component:

$$S_{\text{ref}}(r) = \max_{g \in G_c} F1(r, g).$$

Let g^* denote this best-aligned abstract and y^* its label. We then compute the proxy-reference component:

$$S_{\text{proxy}}(r) = P_\theta(y^* \mid c, r),$$

measuring how strongly r supports the gold label of the best-aligned abstract. The final adapted Ev^{2}R score per r is:

$$\text{Ev}^2\text{R}\ (r) = \frac{1}{2}(S_{\text{ref}}(r) + S_{\text{proxy}}(r)).$$

For each claim, we take the maximum Ev^{2}R (r) across all retrieved abstracts, and the submission-level score is the mean over all claims with at least one non-NEI gold abstract (163 of 172 claims in the test set). Claims linked exclusively to NEI abstracts are excluded from this evaluation as they provide no evidential reference. We discuss our implementation in more depth in Appendix C.

Automatic Verification of Unannotated Pairs. For retrieved CAPs outside the gold annotations, we additionally evaluate the predicted label $\hat{y}$ based on the proxy component of the Ev^{2}R framework. We compute a confidence score using the same DeBERTa model:

$$S_{\text{conf}} = P_\theta(\hat{y} \mid c, r),$$

and a consistency score against the label of the best-aligned gold abstract:

$$S_{\text{cons}} = 1[\hat{y} = y^*].$$

The automatic verification score is then:

$$S_{\text{ver}} = \frac{1}{2}(S_{\text{conf}} + S_{\text{cons}})$$

providing an automatic estimate of label plausibility and consistency for unannotated CAPs, allowing us to assess label quality for retrieved abstracts beyond the annotated gold set.

6. Baselines

Task 1: Abstract Retrieval and Claim Verification. The baseline for task 1 is based on the EFC submission from the 2025 iteration (Upravitelev et al., 2025), chosen due to its strong performance relative to its computational requirements. The retrieval component uses a multi-stage pipeline: Abstracts are first retrieved using BM25 (Robertson and Zaragoza, 2009). The top-$k = 1500$ are then re-ranked using semantic similarity computed with a fine-tuned E5 model[5], which was trained for 3

[4] We make our implementation of the framework public for future use: https://github.com/ryabhmd/climatecheck/tree/master/automatic_eval

[5] https://huggingface.co/intfloat/e5-large-v2

epochs on our dataset using contrastive learning. Finally, a MiniLM-based cross-encoder[6] is applied for re-ranking the top-$k = 150$ results, taking the top 5 abstracts per claim as the final predictions. For task 1.2, we use the DeBERTa sequence classification model, fine-tuned on six Natural Language Inference (NLI) datasets: MultiNLI (Williams et al., 2018), ANLI (Nie et al., 2020), LingNLI (Parrish et al., 2021), WANLI (Liu et al., 2022), FEVER NLI (Nie et al., 2019), and the ClimateCheck training split. We optimise for highest minimum accuracy per label to mitigate class imbalance.

Task 2: Disinformation Narrative Classification. The baseline employs Qwen3-8B (Yang et al., 2025) fine-tuned in a supervised instruction-following setup. Qwen3 is chosen due to good performance on a public leaderboard,[7] specifically targeting the evaluation of smaller and more efficient LLMs. The training data is constructed by converting claim–narrative pairs into chat-style instruction–response examples using the CARDS taxonomy (see prompt in Appendix D). Fine-tuning is performed using parameter-efficient Low-Rank Adaptation (LoRA, Hu et al., 2022) with standard cross-entropy loss on the assistant responses. The model predicts one or more taxonomy codes per claim in a constrained generation format.

Our code for implementing baseline models for both tasks is publicly available to enable reproducibility.[8]

7. Submitted Systems and Results

We received 8 leaderboard submissions across both tasks, the results of which are shown in Table 2 for task 1 and Table 3 for task 2. We also report scores using the Ev^{2}R framework for task 1 submissions in Table 4. In what follows, we briefly describe the system descriptions we received from four teams.

ClimateSense (Ehrhart et al., 2026). For task 1, the team employed a three-stage pipeline, following the winning team of the 2025 iteration (Wang et al., 2025). Initial candidate retrieval used BM25, followed by an ensemble of five fine-tuned BGE re-rankers (Chen et al., 2024) trained with hard negative mining, and aggregated via Reciprocal Rank Fusion. For claim verification, the team opted for zero-shot (ZS) classification using gpt-oss-120b[9]. A fine-tuned DeBERTa-based cross-encoder was also explored but underperformed compared to the ZS LLM approach. For task 2, they adopted a ZS approach using GPT 5.1[10]. The prompt incorporated the full CARDS taxonomy definitions, negative criteria for each category, specificity guidelines, and five-step chain-of-thought (CoT) reasoning.

DFKI-IML (Liang et al., 2026). The team focused mainly on task 1.2, adopting a similar retrieval pipeline as the baseline and directing their attention to claim verification. They compared two self-explainable inference paradigms: Intermediate Reasoning (IR), where the model generates an explanation before predicting the entailment label, and Post-hoc Rationalization (PR), where the label is predicted before the explanation. Three models were evaluated under both paradigms in a ZS setting: GPT-4o-mini (Achiam et al., 2023), Phi-3-Medium (Abdin et al., 2024), and Mistral-Nemo[11]. IR yielded more stable and consistent results across all three models, with GPT-4o-mini achieving the highest scores.

ahilbert (Hilbert et al., 2026). The team focused exclusively on task 2, using Qwen3-8B as a fixed backbone to compare approaches. They investigated three directions: data augmentation, prompt engineering, and reinforcement learning via Group-Relative Policy Optimization (GRPO, Shao et al., 2024). They augmented the data with synthetic claims, applying skewed inverse-frequency reweighting to oversample minority narratives. For prompt engineering, they compared three strategies: a simple direct instruction prompt, a hierarchical two-turn approach, and a CoT prompt that encourages the same hierarchical reasoning within a single forward pass through claim decomposition, high-level group selection, and sub-label assignment. The best-performing setup was ZS CoT prompting combined with Qwen3's native reasoning mode. The team also reported inference-time emissions, showing a clear trade-off between reasoning-based performance and computational cost, with reasoning-heavy approaches consuming roughly 25 times more energy than ZS inference.

XplaiNLP (Foroutan et al., 2026). The team worked on task 2, investigating both encoder-based and decoder-only approaches. For the former, they fine-tuned ModernBERT-large (Warner et al., 2025),

[6] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L12-v2

[7] https://huggingface.co/spaces/k-mktr/gpu-poor-llm-arena

[8] https://github.com/XplaiNLP/ClimateCheck-2026-Baseline/

[9] https://huggingface.co/openai/gpt-oss-120b

[10] https://openai.com/index/gpt-5-1/

[11] https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407

	Task 1.1				Task 1.2			
Team	R@2	R@5	Bpref	$\text{Score}_{1.1}$	P	R	F1	$\text{Score}_{1.2}$
Baseline	0.213	0.403	0.459	0.431	0.683	0.682	0.679	1.082
Ant Bridge (Wang et al., 2025)†	0.218	0.451	0.495	0.473	0.729	0.726	0.725	1.176
ClimateSense (Ehrhart et al., 2026)	**0.221**	**0.443**	**0.489**	**0.466**	**0.742**	**0.744**	**0.740**	**1.183**
berkbubus	0.206	0.429	0.453	0.441	0.391	0.414	0.328	0.757
DFKI-IML (Liang et al., 2026)	0.193	0.352	0.401	0.377	0.670	0.641	0.620	0.972
gardlz	0.193	0.364	0.314	0.340	0.581	0.593	0.579	0.943
ytsoneva	0.060	0.094	0.116	0.105	0.269	0.176	0.148	0.242

† Winning team of the 2025 iteration on the same test set; included for reference only.

Table 2: Results for Task 1. Task 1.1 is evaluated using Recall@2 (R@2), Recall@5 (R@5), Binary Preference (Bpref), and $\text{Score}_{1.1}$. Task 1.2 is evaluated using weighted Precision (P), Recall (R), F1, and $\text{Score}_{1.2}$. Best results among 2026 submissions are shown in **bold**. Shaded cells indicate scores that outperform the baseline.

Team	P (macro)	R (macro)	F1 (macro)	F1 (micro)	F1 (weighted)
Baseline	0.530	0.574	0.514	0.798	0.784
ahilbert (Hilbert et al., 2026)	**0.707**	0.631	**0.625**	0.844	0.821
XplaiNLP (Foroutan et al., 2026)	0.625	**0.640**	0.597	0.801	0.806
ClimateSense (Ehrhart et al., 2026)	0.670	0.568	0.583	**0.876**	**0.844**
alextsiakalou	0.574	0.586	0.548	0.844	0.837

Table 3: Results for Task 2. Systems are evaluated using macro-averaged Precision (P), Recall (R), and F1, as well as micro- and weighted F1. Macro-F1 is the official ranking metric, reflecting performance across all narrative categories regardless of class frequency. Best results are shown in **bold**. Shaded cells indicate scores that outperform the baseline.

RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019) on the the training data, additionally using the Augmented CARDS dataset (Rojas et al., 2024) to mitigate class imbalance. ModernBERT with the augmented dataset achieved the strongest results, surpassing our baseline at a fraction of the computational cost. For decoder-only models, the team fine-tuned Qwen3-8B under three instruction tuning strategies: prompt enhancement with in-context examples, hierarchical instruction tuning where top-level and fine-grained categories were predicted in two sequential stages, and retrieval-augmented instruction tuning where a cross-encoder first retrieved the ten most semantically similar narrative descriptions from the CARDS taxonomy. The retrieval-augmented approach consistently outperformed the other strategies.

8. Error Analysis

Retrieval Difficulty. Figure 2 shows the distribution of per-claim average Recall@5 across all submitted systems, sorted in ascending order. The distribution is notably gradual, suggesting that retrieval difficulty is not a binary easy/hard problem. Only one claim achieves Recall@5=0 across all systems,[12] despite being linked to evidentiary abstract in the gold data. A possible reason could be a lexical mismatch between the claim and the available abstracts, preventing sparse retrieval systems (i.e. BM25, used by all teams) from surfacing relevant evidence. The vast majority of claims (n=137) fall in the mid-range between 0 and 0.5, and the easiest claims (n=25, R@5$\geq$0.5) are rarely solved perfectly by all systems, indicating that no system consistently achieves high recall, even on claims where retrieval is easiest.

Verification Label Confusion. Figure 3 shows normalised confusion matrices for submitted systems from our baseline, ClimateSense, and DFKI-IML on task 1.2.[13] A clear pattern emerges across systems: *Refutes* is the hardest label to predict correctly, with diagonal values ranging from 0.05 to 0.68. This is consistent with the broader NLI literature, showing that refutation requires stronger evidence-grounded reasoning than support or abstention (Atanasova et al., 2022; Thorne et al., 2018a). The most common error is the confusion

[12] "NOAA has adjusted past temperatures to look colder and recent ones warmer"

[13] We show further systems, whose teams did not submit reports, in Appendix E.

Team	Ev^{2}R(r)	S_{ver}	# Evaluated Claims	Avg. Unannotated Abstracts per Claim
Baseline	0.606	0.600	112	1.92
ClimateSense (Ehrhart et al., 2026)	0.559	0.559	111	1.77
berkbubus	0.554	0.468	106	1.67
DFKI-IML (Liang et al., 2026)	0.568	0.527	135	1.97
gardlz	0.585	0.528	134	2.16
ytsoneva	0.672	0.143	163	4.06

Table 4: Ev^{2}R evaluation results for unannotated abstracts retrieved in Task 1. Ev^{2}R(r) is the mean of the reference-based and proxy-reference components evaluating task 1.1. S_{ver} is the added verification score evaluating task 1.2. Only the final leaderboard submission per team is evaluated.

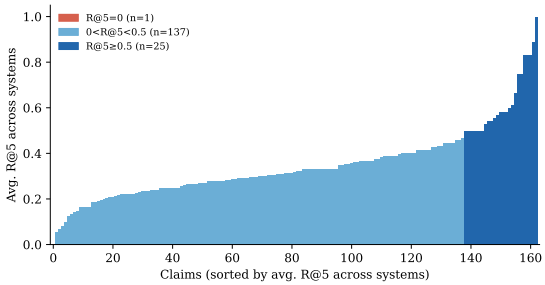


Figure 2: Per-claim average Recall@5 across all submitted systems, sorted in ascending order. Colours indicate retrieval difficulty.

of *NEI* with *Supports*: the baseline misclassifies 29% of such instances, ClimateSense 27%, and DFKI-IML 48%. This suggests that systems tend to interpret partial or weakly relevant evidence as supporting rather than insufficient. We also note a particular error pattern in DFKI-IML, where 39% of gold *Refutes* instances are predicted as *Supports*, meaning that the system flips its interpretation of refuting evidence, potentially leading to false affirmation of misinformation.

Narrative Classification. We analyse error patterns of the submitted systems on Task 2. Figure 4 shows per-label F1 scores for each system alongside IAA (Krippendorff's α). On average, 80.9% of test claims received an exact-match prediction, 3.5% were *partially correct* (correct top-level category, wrong sub-narrative), and 15.6% were assigned a completely wrong top-level category. Of wrong predictions, 81% crossed top-level category boundaries while only 19% stayed within the correct category, indicating that identifying the broad narrative type is the primary challenge rather than distinguishing sub-narratives. Systems also showed a consistent under-prediction tendency: 4.8% of predictions contained too few labels versus only 0.6% with too many, with 0_0 (*no disinformation*) being the most common false prediction. Labels that were difficult for annotators were also difficult for systems. Among labels with sufficient test support, Krippendorff's α and mean system F1 show a perfect rank correlation (Spearman ρ=1.0, n=4), though the small number of qualifying labels limits statistical power. This pattern is visible in Figure 4: label 5_1 (α=0.55) has both the lowest and most variable system F1 (0.22-0.62), while 0_0 (α=0.72) is consistently well-handled (F1 $\geq$ 0.89). Label 2_1 (*natural cycles*; α=0.56) shows the widest inter-system spread (0.46–0.92), suggesting that system architecture matters most for labels with moderate annotator agreement. Seven claims (4.1%) were misclassified by all five systems (F_1=0.0), and a majority-vote ensemble scored below the best individual system (84.3% vs. 85.5% exact match), indicating correlated errors.

Cross-task Analysis. To investigate whether disinformation narratives affects retrieval and verification difficulty, we analyse system performance across the five main narrative groups using the two systems that submitted predictions for both tasks: our baseline and ClimateSense. We first note that retrieval is narrative-agnostic, with Recall@5 results being consistently high across all groups.[14] This indicates that relevant scientific evidence is retrievable regardless of the type of disinformation narrative a claim embodies. However, we see a different pattern when looking at verification difficulty, signaling that it is strongly narrative-dependent. Looking at the most difficult label to verify, *Refutes*, we note a clear gradient where claims belonging to Group 3 (*climate impacts are not bad*) are the easiest to refute, while Groups 4 (*climate solutions won't work*) and 5 (*climate movement/science is unreliable*) are systematically the hardest (See Figure 5). This result is structurally motivated, since Group 4 claims are primarily normative or economic in nature, and Group 5 claims are epistemological, attacking the credibility of science itself rather than making falsifiable empirical assertions. Thus, using a scientific abstract to refute a claim that science is

[14] See Appendix E for full results.

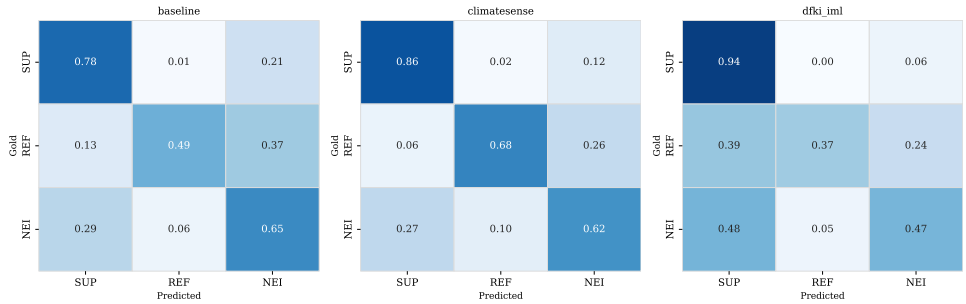


Figure 3: Confusion matrices for the baseline, ClimateSense, and DFKI-IML predictions on task 1.2, normalised by claim. SUP = Supports, REF = Refutes, NEI = Not Enough Information.

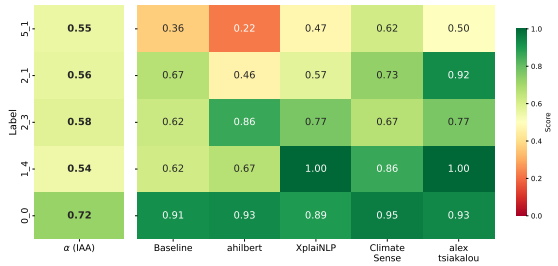


Figure 4: Per-label F1 scores across participating systems and IAA (Krippendorff's α, leftmost column). Labels sorted by mean system F1 (ascending); only labels with more than 3 instances in the test set are shown.

unreliable is self-referentially problematic: the evidence source is precisely what the claim calls into question. This represents a structural limitation of evidence-grounded AFC that is difficult to resolve through improved retrieval or stronger verification models alone.

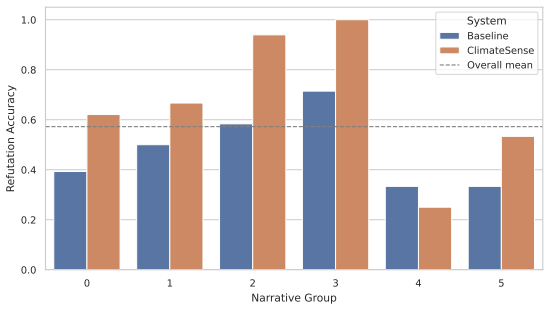


Figure 5: Refutation accuracy by narrative group based on the CARDS taxonomy for the baseline and ClimateSense systems.

9. Discussion

System Trends. Results across both tasks reveal a consistent pattern: structured reasoning and task-specific architectural choices matter more than model size or data volume alone. For task 1, the strongest retrieval systems rely on multi-stage dense re-ranking pipelines with cross-encoder ensembles, echoing findings from the 2025 iteration. Notably, although ClimateSense adopts a retrieval approach similar to the 2025 winning team, their retrieval performance remains lower despite the availability of more training data in this iteration, suggesting that implementation details and training methodology are more important than data volume at this scale. For task 2, the most effective approaches share a common thread: hierarchical or structured reasoning that identifies coarse-grained narrative groups before committing to fine-grained sub-labels. Team ahilbert's results further highlight an important practical consideration, raising questions about the deployability of LLM-based verification at scale in real-world AFC pipelines.

Evaluation Beyond Gold Annotations. The Ev^{2}R results reveal trends that the official leaderboard alone does not capture. Most notably, team *ytsoneva*, which ranks last on the official retrieval metric, achieves the highest Ev^{2}R(r) score. This is largely explained by its retrieval behaviour: it is the only system that retrieves at least one unannotated abstract for all 163 evaluated claims, with an average of more than 4 out of 5 retrieved abstracts per claim falling outside the gold annotations. This means the official metrics penalise this system almost entirely for retrieving evidence that was never judged, rather than for retrieving poor evidence. The Ev^{2}R score suggests that this evidence is in fact of reasonable quality, a signal that wouldn't be visible under standard evaluation alone. Beyond this case, the baseline achieves the highest Ev^{2}R score among the remaining systems despite not ranking first on the official leaderboard, further suggesting that systems optimising directly for gold annotations may be learning retrieval biases introduced by the annotation process. Taken together, these results reinforce the importance of complementing recall-based metrics with automatic evaluation frameworks whenever annotation complete-

ness cannot be guaranteed, which is a particularly pressing concern in AFC, where exhaustive annotation is rarely feasible.

The Cross-Task Connection. Our cross-task analysis reveals that verification difficulty is strongly narrative-dependent, with narrative groups 4 and 5 proving systematically resistant to evidence-based refutation. This finding has a direct implication for system design: a pipeline that first classifies the disinformation narrative of a claim and then conditions its verification strategy on that classification is a natural and promising direction. For claims that make concrete empirical assertions, standard evidence retrieval and NLI-based verification may be sufficient. For other claims, however, the structural mismatch between the claim type and the evidence source suggests that different strategies might be needed, such as specialised evidence types beyond scholarly abstracts, or flagging for human review rather than automated verdict. Crucially, none of the submitted systems attempted to exploit the connection between the two tasks, leaving narrative-aware verification an open and promising direction that can be explored further.

10. Conclusion

We presented ClimateCheck 2026, expanding the shared task with tripled training data, a new disinformation narrative classification task, and a unified benchmark connecting evidence retrieval, claim verification, and narrative classification over a common set of claims. Three of four participating teams outperformed our baselines, with multi-stage dense retrieval pipelines and structured hierarchical reasoning proving the most effective strategies for tasks 1 and 2 respectively. Beyond system-level results, our cross-task analysis reveals a structural limitation of evidence-grounded AFC: while retrieval difficulty is narrative-agnostic, verification difficulty is strongly narrative-dependent, with epistemological and normative claims proving more difficult to scholarly refutation. This motivates narrative-aware verification as a promising direction for future work, which our unified benchmark is uniquely positioned to support.

11. Limitations

The ClimateCheck dataset is restricted to English-language claims only, limiting the applicability of trained systems to the broader multilingual landscape of online climate discourse. In addition, the claim verification task, as formulated here, treats verification as a pairwise problem between a single claim and a single abstract. This is a simplification that enables scalable annotation, but does not reflect the full complexity of scientific AFC, where a verdict may depend on evidence across multiple documents, conflicting findings, and following chains of citations. Future iterations could explore multi-hop verification settings, where evidence from several abstracts must be jointly considered to reach a final verdict.

In terms of participating systems, we note that, for task 1, teams largely replicated or incrementally extended the methods that proved successful in the 2025 iteration of the shared task, with little methodological novelty in the retrieval stage. Additionally, despite all claims being annotated for both verification and disinformation narrative labels, no team explored whether narrative predictions could inform veracity classification or vice versa. Given that disinformation narratives carry implicit expectations about the direction of evidence, narrative-aware verification represents a promising direction.

Finally, the annotation of fine-grained disinformation narratives remains challenging even for human annotators, as reflected in the below-threshold IAA for several categories. This inherent subjectivity sets an upper bound on what automated systems can be expected to achieve, and suggests that future work may benefit from the introduction of hierarchical evaluation metrics that reward partial credit for correct top-level classification.

12. Acknowledgments

This work was supported by the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[15] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.). The consortium is funded by the Federal Republic of Germany and its states through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). Furthermore, the work was partly performed in the scope of the projects "VeraXtract" (reference: 16IS24066) and "news-polygraph" (reference: 03RU2U151C) funded by the German Federal Ministry for Research, Technology and Aeronautics (BMFTR). We thank the annotators: Emmanuella Asante, Farzaneh Hafezi, Senuri Jayawardena, and Shuyue Qu for working on the extension of the task 1 data, and Berk Bubus, Paul-Conrad Feig, Neda Foroutan, Alexandra Tsiakalou for annotating the task 2 data. We also thank Nikolas Rauscher for helping with training the DeBERTa model used for computing the Ev^2R scores.

[15] https://www.nfdi4datascience.de

13. Bibliographical References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025a. The ClimateCheck dataset: Mapping social media claims about climate change to corresponding scholarly articles. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 42–56, Vienna, Austria. Association for Computational Linguistics.

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025b. The ClimateCheck shared task: Scientific fact-checking of social media claims about climate change. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 263–275, Vienna, Austria. Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mubashara Akhtar, Rami Aly, Yulong Chen, Zhenyun Deng, Michael Schlichtkrull, Chenxi Whitehouse, and Andreas Vlachos. 2025. The 2nd automated verification of textual claims (AVeriTeC) shared task: Open-weights, reproducible and efficient systems. In *Proceedings of the Eighth Fact Extraction and VERification Workshop (FEVER)*, pages 201–223, Vienna, Austria. Association for Computational Linguistics.

Mubashara Akhtar, Michael Schlichtkrull, and Andreas Vlachos. 2024. Ev2r: Evaluating evidence retrieval in automated fact-checking. *arXiv preprint arXiv:2411.05375*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2022. Fact checking with insufficient evidence. *Trans. Assoc. Comput. Linguistics*, 10:746–763.

Chris Buckley and Ellen M Voorhees. 2004. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32.

Tom Calamai, Oana Balalau, and Fabian M. Suchanek. 2025. Benchmarking the benchmarks: Reproducing climate-related NLP tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17967–18009, Vienna, Austria. Association for Computational Linguistics.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality,

multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Travis G Coan, Constantine Boussalis, John Cook, and Mirjam O Nanko. 2021. Computer-assisted classification of contrarian claims about climate change. *Scientific reports*, 11(1):22320.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xingyu Deng, Xi Wang, and Mark Stevenson. 2025. The next phase of scientific fact-checking: Advanced evidence retrieval from complex structured academic papers. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval, ICTIR 2025, Padua, Italy, 18 July 2025*, pages 436–448. ACM.

Thibault Ehrhart, Grégoire Burel, and Raphael Troncy. 2026. ClimateSense at ClimateCheck 2026. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Neda Foroutan, Alexandra Tsiakalou, and Vera Schmitt. 2026. Retrieval-augmented LLMs and encoder models for multi-label climate disinformation narrative classification. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Jennifer R Fownes, Chao Yu, and Drew B Margolin. 2018. Twitter and climate change. *Sociology Compass*, 12(6):e12587.

Jesús M. Fraile-Hernández, Anselmo Peñas, and Pablo Moral. 2025. Automatic identification of narratives: Evaluation framework, annotation methodology, and dataset creation. *IEEE Access*, 13:11734–11753.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: decoding-enhanced BERT with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Arthur Hilbert, Nils Feldhus, Jing Yang, and Vera Schmitt. 2026. Comparing hierarchical approaches for fine-grained climate disinformation narrative classification. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. LoRA: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Klaus Krippendorff. 2018. *Content Analysis: An Introduction to Its Methodology*, 4th edition. SAGE Publications, Thousand Oaks, CA.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Siting Liang, Omar Adjali, and Daniel Sonntag. 2026. Towards efficient self-explainable climate-related claim verification with generative models. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with

neural semantic matching networks. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Nikolaos Nikolaidis, Nicolas Stefanovitch, Purificação Silvano, Dimitar Iliyanov Dimitrov, Roman Yangarber, Nuno Guimarães, Elisa Sartori, Ion Androutsopoulos, Preslav Nakov, Giovanni Da San Martino, and Jakub Piskorski. 2025. Polynarrative: A multilingual, multilabel, multi-domain dataset for narrative extraction from news articles. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31323–31345, Vienna, Austria. Association for Computational Linguistics.

Alicia Parrish, William Huang, Omar Agha, Soo-Hwan Lee, Nikita Nangia, Alexia Warstadt, Karmanya Aggarwal, Emily Allaway, Tal Linzen, and Samuel R. Bowman. 2021. Does putting a linguist in the loop improve NLU data collection? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4886–4901, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Cristian Rojas, Frank Algra-Maschio, Mark Andrejevic, Travis Coan, John Cook, and Yuan-Fang Li. 2024. Augmented CARDS: A machine learning approach to identifying triggers of climate change misinformation on twitter. *CoRR*, abs/2404.15673.

Harri Rowlands, Gaku Morio, Dylan Tanner, and Christopher Manning. 2024. Predicting narratives of climate obstruction in social media advertising. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5547–5558, Bangkok, Thailand. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos. 2024. The automated verification of textual claims (AVeriTeC) shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 1–26, Miami, Florida, USA. Association for Computational Linguistics.

Michael Schlichtkrull, Nedjma Ousidhoum, and Andreas Vlachos. 2023. The intended uses of automated fact-checking artefacts: Why, how and who. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8618–8642, Singapore. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Max Upravitelev, Nicolau Duran-Silva, Christian Woerle, Giuseppe Guarino, Salar Mohtaj, Jing Yang, Veronika Solopova, and Vera Schmitt. 2025. Comparing LLMs and BERT-based classifiers for resource-sensitive claim verification in social media. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 281–287, Vienna, Austria. Association for Computational Linguistics.

Juraj Vladika and Florian Matthes. 2023. Scientific fact-checking: A survey of resources and approaches. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6215–6230, Toronto, Canada. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

David Wadden and Kyle Lo. 2021. Overview and insights from the SCIVER shared task on scientific claim verification. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 124–129, Online. Association for Computational Linguistics.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. 2022. SciFact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4719–4734, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junjun Wang, Kunlong Chen, Zhaoqun Chen, Peng He, and Wenlu Zheng. 2025. Winning ClimateCheck: A multi-stage system with BM25, BGE-reranker ensembles, and LLM-based analysis for scientific abstract retrieval. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 276–280, Vienna, Austria. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 2526–2547. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. 2022. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7):100543.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

A. Inter-Annotator Agreement And Annotation Process of Task 2

Since we use a multi-label annotation schema, we decompose the task into independent binary decisions per label. Table 5 reports agreement for all labels in the taxonomy.

Pairwise Cohen's κ. Computed on flattened binary label vectors, κ ranges from 0.747 to 0.819 across all six annotator pairs (0.773–0.850 at the top-level), indicating substantial agreement.

Krippendorff's α. At the top-level, α indicates substantial agreement on categories 0 and 1 ($\alpha \geq 0.67$), while showing moderate agreement on categories 2-5 ($\alpha = 0.54$–0.66). Among individual subnarratives, the highest agreement was observed for 1_6 ("Sea level rise is exaggerated"; $\alpha = 0.795$), 3_2 ("Impacts are beneficial"; $\alpha = 0.688$), and 1_1 ("Ice isn't melting"; $\alpha = 0.672$). Labels in the denial-of-cause family (2_1, 2_3, 2_5) and solutions-won't-work family (4_2, 4_4) showed lower agreement ($\alpha = 0.26$–0.58), reflecting the greater interpretive difficulty of distinguishing closely related subnarratives.

Positive Agreement. For rare labels, κ and α can be misleadingly low due to the prevalence paradox: agreement on label absence inflates expected agreement, deflating chance-corrected scores. We therefore also report Positive Agreement (PA), using the Dice coefficient over annotator pairs:

$$PA = \frac{2TP}{2TP + FP + FN}$$

For the most frequent disinformation labels (2_1, 5_1, 2_3), PA ranges from 0.57 to 0.59, confirming moderate positive-case agreement consistent with Krippendorff's α.

Negative Agreement. Label 2_2 ("Human greenhouse gases are not causing climate change. It's non-greenhouse gas human climate forcings (aerosols, land use)") exhibited negative agreement ($\alpha = -0.003$), meaning annotators disagreed more than chance. It was applied 12 times across all annotators, never by more than one annotator on the same item. In each case, the claim discussed topics like deforestation or aerosols but stated factually accurate information; one annotator coded based on topic presence while the others correctly assigned 0_0. In 23 of the pairwise comparisons, the non-2_2 annotator assigned 0_0. The negative agreement thus reflects a systematic confusion between topic and narrative. After adjudication, 2_2 was retained in only one case.

Annotation Edge Cases. Three recurring annotation challenges during the annotation process: 1. Mention vs. endorsement: claims that reference a disinformation narrative but immediately question it (e.g., "Some argue that CO_2 isn't the main driver of global warming. This is a controversial viewpoint, as the vast majority of scientists agree CO2 plays a significant role in climate change.") were labeled 0_0, as the text is *about* the narrative rather than an instance of it; 2. Disinformation intent in factually true statements: a claim such as "Jupiter's weather is powered by a giant internal heat source. #Climate" is scientifically accurate, but the #Climate hashtag hints at an implicit natural-cycles framing (2_1); annotators disagreed on whether to code surface content or inferred intent, and such cases were resolved conservatively as 0_0; 3. Tone-dependent ambiguity: "Scientists are still working on the final numbers for the projected sea level rise" can be read as neutral reporting or as implying that climate science is unsettled (5_1); adjudicators independently converged on 5_1, interpreting "final" as doubting scientific consensus.

Summary. At the subnarrative level, agreement varies considerably. Labels with clear, observable indicators (e.g., 1_6: sea level claims; 1_1: ice/glacier claims) achieve substantial agreement, while labels requiring more interpretive judgment show moderate to fair agreement like within categories 3 (*impacts not bad*) and 4 (*solutions won't work*). This pattern is consistent with the annotation evaluation of the original CARDS taxonomy, where top-level distinctions are more reliable than fine-grained ones Coan et al. (2021). The 16.6% of claims requiring adjudication were resolved by two authors who had not served as annotators, ensuring independence between the annotation and adjudication stages. Each adjudicator first reviewed the claim independently before a joint discussion, and all decisions were documented with reasoning.

B. Evaluation Metrics

Recall@K is defined as:

$$\text{R@K} = \frac{\#\text{gold evidentiary abstracts in top-K}}{\#\text{gold evidentiary abstracts for the claim}}$$

Because relevance judgments are incomplete, we additionally report Bpref, which does not assume exhaustive annotation (Buckley and Voorhees, 2004), and is defined as:

$$\text{Bpref} = \frac{1}{R} \sum_{r} \left(1 - \frac{|n \text{ ranked higher than } r|}{R}\right)$$

where R is the total number of judged relevant documents, r is a judged relevant document retrieved

Label	Narrative	N	Prev.	α	κ	PA
0 — No disinformation narrative						
0_0	No disinformation narrative detected	776	.715	.722	.722	.921
1 — Global warming is not happening						
1_0	Global warming is not happening (general)	12	.004	.175	.132	.133
1_1	Ice/permafrost/snow cover isn't melting	28	.017	.672	.676	.681
1_2	We're heading into an ice age/global cooling	23	.011	.556	.570	.575
1_3	Weather is cold/snowing	10	.004	.350	.313	.315
1_4	Climate hasn't warmed over the last decade(s)	36	.018	.544	.536	.543
1_5	Oceans are cooling/not warming	4	.002	.583	.610	.611
1_6	Sea level rise is exaggerated/not accelerating	22	.016	.795	.793	.797
1_7	Extreme weather isn't increasing/not linked to CC	22	.010	.427	.365	.370
1_8	Changed name from 'global warming' to 'climate change'	3	.002	.750	.738	.739
2 — Human greenhouse gases are not causing climate change						
2_0	Human GHGs are not causing CC (general)	14	.004	.080	.043	.044
2_1	It's natural cycles/variation	94	.051	.562	.561	.583
2_2	Non-GHG human forcings (aerosols, land use)	12	.003	$-.003$	$-.003$	.000
2_3	No evidence for GHG effect driving climate change	48	.028	.581	.578	.590
2_4	CO_2 is not rising/ocean pH is not falling	3	.003	.800	.750	.750
2_5	Human CO_2 emissions are miniscule	16	.006	.273	.277	.281
3 — Climate impacts/global warming is beneficial/not bad						
3_0	Climate impacts are beneficial/not bad (general)	14	.005	.230	.187	.189
3_1	Climate sensitivity is low/negative feedbacks	7	.003	.265	.281	.283
3_2	Species/ecosystems aren't impacted/are benefiting	22	.014	.688	.686	.690
3_3	CO_2 is beneficial/plant food	30	.015	.568	.585	.591
3_4	It's only a few degrees (or less)	19	.006	.190	.175	.179
3_5	CC doesn't contribute to conflict/threaten security	8	.003	.131	.073	.074
3_6	CC doesn't negatively impact health	3	.002	.666	.694	.694
4 — Climate solutions won't work						
4_0	Climate solutions won't work (general)	2	.001	$-.000$	—	.000
4_1	Climate policies are harmful	13	.004	.122	.100	.103
4_2	Climate policies are ineffective/flawed	32	.013	.255	.254	.264
4_3	Too hard to solve (politically/economically/technically)	16	.006	.301	.297	.301
4_4	Clean energy/biofuels won't work	26	.010	.293	.260	.266
4_5	People need energy from fossil fuels/nuclear	9	.004	.398	.363	.365
5 — Climate movement/science is unreliable						
5_0	Climate movement/science is unreliable (general)	1	.000	.000	—	.000
5_1	Science is uncertain/unsound (data, methods, models)	70	.041	.551	.551	.570
5_2	Climate movement is alarmist/political/biased	26	.010	.238	.214	.221
5_3	Climate change is a conspiracy	6	.002	.249	.261	.262

Table 5: Per-label IAA sorted by top-level category. N = items where $\geq$1 annotator assigned the label; Prev. = prevalence; α = Krippendorff's alpha; κ = avg. pairwise Cohen's kappa; PA = positive agreement (Dice).

by the system, and n is a member of the first R judged non-relevant documents retrieved before r.

C. Ev^{2}R Adaptation Details

C.1. Reference-based Component

To compute the reference-based component of Ev^{2}R , we use the Gemini 2.5 Pro model (Comanici et al., 2025) to decompose gold and retrieved abstracts into atomic factual statements and evaluate their alignment. We adapt the prompt shared by Akhtar et al. (2024) to an instance from the ClimateCheck dataset, shown in Figure 6.[16] For each retrieved abstract r and gold abstract g, the resulting F1 score is used as the reference-alignment score $F1(r, g)$. When multiple gold abstracts are available for a claim, we retain the abstract with the the maximum alignment score. If several gold abstracts achieve the same score, the first one is chosen. We cache the results across all submissions to prevent score variations for the same CAP.

[16]The full prompt is available at: https://github.com/ryabhmd/climatecheck/blob/master/automatic_eval/reference_based_prompt.txt

```
You will get as input a claim, a reference evidence and a predicted
evidence.
Please verify the correctness of the predicted evidence by comparing it
to the reference evidence, following these steps:

1. Break down the PREDICTED evidence in independent facts. Each fact
should be a separate sentence. Facts should not be specific to the
study, they must be generalized statements about climate science.

2. Evaluate each fact individually: is the fact supported by the
REFERENCE evidence? Do not use additional sources or background
knowledge.

3. Next, break down the REFERENCE evidence in independent facts. Each
fact should be a separate sentence. Facts should not be specific to
the study, they must be generalized statements about climate science.

4. Evaluate each fact individually: is the fact supported by the
PREDICTED evidence? Do not use additional sources or background
knowledge.

5. Finally summarise (1.) how many predicted facts are supported by
the reference evidence, (2.) how many reference facts are supported by
the predicted evidence.

Generate the output in form of a json as shown in the examples below.

---

Examples:

Claim: Antarctica is losing land ice at an accelerating rate,
according to satellite measurements.

Reference evidence: The Antarctic Ice Sheet is losing mass at an
accelerating rate, and playing a [...] causing a global sea-level
rise of about 3 m.

Predicted evidence: Greenland ice core records provide clear evidence
of rapid changes [...] its ice mass loss.

Output: {"facts in predicted evidence": "1. Greenland ice core
records provide clear evidence of rapid [...]", "facts count predicted
evidence": 5, "support predicted evidence": 2, "facts in reference
evidence": "1. Antarctic Ice Sheet is losing mass [...]", "facts
count reference evidence": 5, "support reference evidence": 1}

---

Input:

Claim: $claim

Reference evidence: $reference_evidence

Predicted evidence: $predicted_evidence
```

Figure 6: Prompt used for the Ev^{2}R reference-based scorer adapted for the ClimateCheck shared task based on the prompt provided by Akhtar et al. (2024). Both the reference evidence and the retrieved evidence are decomposed into atomic facts representing generalised climate science statements before being assessed against each other.

C.2. Trained DeBERTa Classifier

To compute the proxy-reference component of Ev^{2}R , as well as the automatic claim verification score for task 1.2, we train a DeBERTa-based classifier for 3-way NLI-style claim verification with the labels *Supports*, *Refutes*, and *NEI*.

The model is initialised from an existing classifier,[17] a DeBERTa-v3-Large checkpoint pre-trained on a diverse set of NLI datasets. We fine-tune it on a combined training set drawn from five datasets: FEVER (Thorne et al., 2018a), VitaminC (Schuster et al., 2021), HoVer (Jiang et al.,

[17] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

2020), AVeriTeC (Schlichtkrull et al., 2024), and our own ClimateCheck data. For ClimateCheck, we use a stratified 90/10 train/validation split, ignoring the official test split to prevent leakage. The combined training set comprises 488,018 examples and the validation set 77,843 examples.

Training is performed for 3 epochs on a single NVIDIA H100 GPU with a batch size of 32, a learning rate of 2×10^{-6}, linear learning rate scheduling with 200 warmup steps, and a maximum sequence length of 320 tokens. The best checkpoint is selected by evaluation accuracy on the combined validation set, evaluated every 2,000 steps. The selected checkpoint at step 26,000 (epoch 1.7) achieves a macro-F1 of 0.886 and an accuracy of 0.921 on the combined validation set, with per-class F1 scores of 0.955 (*supports*), 0.920 (*refutes*), and 0.782 (*NEI*). We make the model publicly available.[18]

D. Disinformation Narrative Classification Prompt

The prompt shown in Figure 7 was used for fine-tuning the baseline model for task 2.

```
You are an expert in detecting climate
change related disinformation. You get a
claim and your task is to classify the
claim using the taxonomy codes provided.

Rules:
- If the claim aligns with narratives
  from the taxonomy, list applicable
  codes separated by a semicolon (;).
- If no disinformation is found,
  return '0_0'.

Taxonomy:
{taxonomy_str}

Claim: "{claim}"

Codes:
```

Figure 7: Prompt used for the baseline implementation of task 2.

E. Further Error Analyses

Figure 8 shows the confusion matrices of submitted systems from *berkbubus*, *gardlz*, and *ytsoneva* on task 1.2. Table 6 displays the R@5 results for the two systems submitting for both tasks per narrative group based on the CARDS taxonomy. No notable differences in retrieval appear to exist, indicating that retrieval difficulty is narrative-agnostic.

[18] https://huggingface.co/rausch/deberta-climatecheck-2463191-step26000

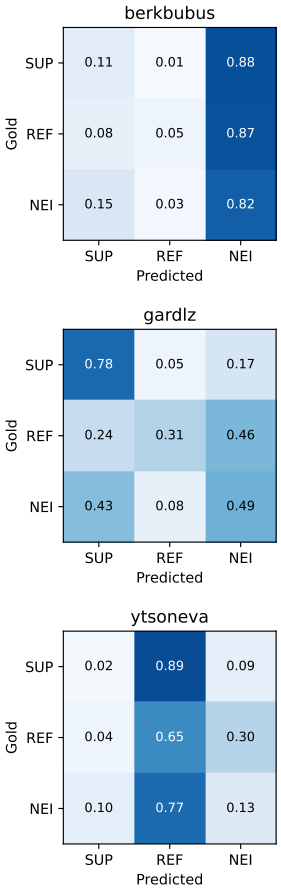


Figure 8: Confusion matrices for the berkbubus, gardlz, and ytsoneva predictions on task 1.2, normalised by claim. SUP = Supports, REF = Refutes, NEI = Not Enough Information.

Narrative Group	R@5	N
Baseline		
0	0.992	124
1	0.933	15
2	0.938	16
3	1.000	6
4	1.000	6
5	1.000	11
ClimateSense		
0	0.992	124
1	1.000	15
2	1.000	16
3	1.000	6
4	1.000	6
5	0.909	11

Table 6: R@5 by narrative group for both systems. N indicates the number of claims per group. Results show minimal variance across narrative groups, indicating that retrieval difficulty is narrative-agnostic.

ClimateSense at ClimateCheck 2026

Thibault Ehrhart[1], **Grégoire Burel**[2], **Raphaël Troncy**[1]
[1]EURECOM, Sophia Antipolis, France
[2]Knowledge Media Institute, The Open University, Milton Keynes, UK
{thibault.ehrhart,raphael.troncy}@eurecom.fr, gregoire.burel@open.ac.uk

Abstract

This paper describes our submission to the ClimateCheck 2026 shared task on scientific fact-checking of climate-related claims (Task 1) and disinformation narrative classification (Task 2). For Task 1, we use a three-stage pipeline combining BM25 retrieval over 394,269 scientific abstracts, ensemble re-ranking with five fine-tuned BGE cross-encoders aggregated via Reciprocal Rank Fusion, and zero-shot claim verification using the `gpt-oss-120b` model. For Task 2, we use a zero-shot approach with a custom prompt based on the CARDS taxonomy and the `gpt 5.2` model. Our system outperforms the organizers' baseline across all subtasks. Furthermore, we achieve the best result for the Task 1.1 (retrieval score of 0.466), and Task 1.2 (verification score of 1.183 - F1 + Recall@5), and the third best result for Task 2 (Macro F1 of 0.583).

Keywords: climate misinformation, fact verification, scientific claim retrieval, natural language inference

1. Introduction

Climate misinformation spreads rapidly on social media, shaping public perception and potentially undermining evidence-based policy decisions. Automated fact-checking systems that ground claims in peer-reviewed scientific literature offer a scalable approach to counter this trend. The ClimateCheck 2026 shared task (Abu Ahmad et al., 2026) provides a structured evaluation framework for such systems, challenging participants to (1a) retrieve relevant scientific abstracts for a given climate claim (Task 1.1) and (1b) classify each claim-abstract pair as *Supports*, *Refutes*, or *Not Enough Information* (Task 1.2), and (2) classify claims according to known climate disinformation narratives based on the CARDS taxonomy (Coan et al., 2021a) (Task 2).

We present our system for the two tasks of ClimateCheck 2026. For Task 1, our approach follows a three-stage pipeline combining sparse retrieval, ensemble neural re-ranking, and zero-shot LLM-based verification. For Task 2, after considering various approaches, we employ a zero-shot LLM-based classification approach using a prompt based on the CARDS taxonomy definitions (Coan et al., 2021a). Our system outperforms the organizers' baseline across all subtasks. Our code is publicly available at `https://github.com/climatesense-project/climatesense-climatecheck2026`.

2. Related Work

Scientific claim verification. The task of verifying scientific claims against evidence has been explored through datasets such as SciFact (Wadden et al., 2020), FEVER (Thorne et al., 2018), and Climate-FEVER (Diggelmann et al., 2020). Approaches typically follow a retrieve-and-verify paradigm where relevant documents are first retrieved and then used to classify claim veracity. The ClimateCheck shared task (Abu Ahmad et al., 2026) extends this paradigm specifically to the climate domain, with claims sourced from social media and verified against a large corpus of scientific abstracts. The winning system of ClimateCheck 2025 (Wang et al., 2025) combined BM25 retrieval with fine-tuned BGE cross-encoder re-rankers and Reciprocal Rank Fusion, establishing a strong baseline for our work.

Retrieval and re-ranking. Traditional sparse retrieval methods like BM25 (Robertson and Zaragoza, 2009) remain competitive baselines, particularly when combined with neural re-ranking. Cross-encoder re-rankers such as BGE-Reranker (Xiao et al., 2023) score query-document pairs jointly, capturing fine-grained relevance signals that sparse methods miss. Ensemble approaches that combine multiple rankers through techniques like Reciprocal Rank Fusion (Cormack et al., 2009) have been shown to improve robustness over individual models. Hard negative mining, where the most confusing non-relevant documents are used during training, is a well-established technique for improving cross-encoder re-rankers (Xiong et al., 2021).

NLI for fact verification. Natural Language Inference (NLI) models have been widely applied to fact verification tasks. Cross-encoder models trained on NLI datasets, such as DeBERTa-v3 (He et al., 2023) fine-tuned on NLI benchmarks, provide strong baselines for textual entailment classification. More recently, large language models have shown strong performance on NLI tasks, particularly when combined with chain-of-thought reason-

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 66–77

ing (Wei et al., 2022) and structured output formats.

Disinformation narrative classification. The CARDS taxonomy (Coan et al., 2021a) provides a systematic framework for categorizing climate disinformation narratives, identifying common rhetorical strategies used to cast doubt on climate science. The most common approach for automatically identifying such categories has been to address it as a multiclass text classification task rather than a hierarchical multi-label classification task. CARDS classification models have also been developed by the authors of the taxonomy. The CARDS (Coan et al., 2021b) and Augmented CARDS (Rojas et al., 2024) are fine-tuned models, respectively built on top of ROBERTa and DeBERTa. The original CARDS classifier uses train data extracted from conservative think tank websites and contrarian blogs, whereas the Augmented CARDS classifier uses data from the Climate Change Twitter dataset (Effrosynidis et al., 2022). A more recent work by the authors, which appears to extend the CARDS taxonomy to 7/8 top-level categories, suggests that zero-shot classification performs better than both few-shot and fine-tuned models for classifying content according to the taxonomy (Coan et al., 2026). These three CARDS models assume that CARDS categories are mutually exclusive (multiclass classification), whereas the ClimateCheck 2026 task is multi-label. Therefore, for the shared task, we decided to design a multi-label zero-shot classifier.

3. Dataset

The ClimateCheck 2026 dataset consists of climate-related claims paired with scientific abstracts and verification labels. Claims were gathered from existing resources including ClimaConvo (Shiwakoti et al., 2024), DEBAGREEMENT (Pougué-Biyong et al., 2021), Climate-FEVER (Diggelmann et al., 2020), MultiFC (Augenstein et al., 2019), and ClimateFeedback[1], encompassing claims extracted from social media platforms (Twitter/X and Reddit) as well as synthetically generated claims from news outlets.

Publications corpus. The retrieval corpus contains 394,269 scientific abstracts sourced from OpenAlex (Priem et al., 2022) and S2ORC (Lo et al., 2020), with an average abstract length of 241 words. Each abstract entry includes metadata such as DOI, title, fields of study, citation count, and source database.

Training set. The training data comprises 3,023 claim-abstract pairs spanning 763 unique claims. The label distribution for Task 1 is imbalanced: *Supports* accounts for 46.3% (1,399 pairs), *Not Enough Information* for 38.8% (1,173 pairs), and *Refutes* for 14.9% (451 pairs). Of the training data, 1,879 pairs are new to the 2026 edition and 1,144 were carried over from the 2025 iteration. Average claim length is 16.9 words. For Task 2, each unique claim is additionally annotated with disinformation narrative labels according to the CARDS taxonomy in a multi-label fashion.

Test set. The test set contains 176 claims (average length: 18.0 words) for which systems must retrieve relevant abstracts, predict verification labels (Task 1), and classify disinformation narratives (Task 2).

4. Methodology

For Task 1, our retrieval pipeline (Stages 1–2) adopts the winning system of ClimateCheck 2025 Wang et al. (2025), combining BM25 retrieval with fine-tuned cross-encoder ensembles and Reciprocal Rank Fusion. For claim verification (Stage 3), we use a locally served open-weight model (`gpt-oss-120b`) with zero-shot NLI prompting. For Task 2, we use a zero-shot prompt to identify the CARDS narratives found in the climate-related claims.

4.1. Stage 1: BM25 Retrieval

We use BM25 (Okapi variant) for initial candidate retrieval. Both claims and abstracts undergo preprocessing: lowercasing, removal of non-alphabetic characters, tokenization using NLTK, and removal of English stopwords. We index all 394,269 abstracts and retrieve the top 5,000 candidates per claim, providing broad recall for the subsequent re-ranking stage.

4.2. Stage 2: Reranker Ensemble

We create a stratified train/validation split at the claim level, with 15% of the claims held out for validation, stratified by the majority label of each claim to preserve label proportions.

Hard negative mining. To construct challenging training examples, we use a pre-trained BGE-Reranker-Large model (`BAAI/bge-reranker-large`) to score the BM25 candidates for each training claim. For each claim, we select the top-scoring non-gold abstracts as hard negatives (up to 500 hard negatives per claim from the top 1,000 candidates). Training triplets are then constructed from each claim-abstract pair in the training set: each positive pair is combined with 10 random negatives and 5 hard negatives, yielding a total of 15 triplets per training pair.

Fine-tuning. We fine-tune five cross-encoder re-ranker models with different configurations to promote diversity in the ensemble:

[1] https://science.feedback.org/climate-feedback

Base Model	Batch Size	Margin	LR
BGE-Reranker-Large	8	0.20	1e-5
BGE-Reranker-Large	16	0.20	1e-5
BGE-Reranker-Large	8	0.25	1e-5
BGE-Reranker-Large	16	0.25	1e-5
BGE-Reranker-v2-m3	16	0.20	1e-5

Table 1: Reranker training configurations.

All models are trained for 1 epoch using MarginRankingLoss with AdamW optimization and a maximum sequence length of 512 tokens. We save the model checkpoint with the best validation accuracy (proportion of triplets where the positive score exceeds the negative score).

Inference and ensemble. Each fine-tuned reranker independently scores the top-5,000 BM25 candidates per claim and retains the top 500. The five ranked lists are then combined using Reciprocal Rank Fusion (RRF) with $k = 6$:

$$\text{RRF}(d) = \sum_{i=1}^{5} \frac{1}{k + r_i(d)}$$

where $r_i(d)$ is the rank of document d in the ranked list of re-ranker i. Documents not appearing in a re-ranker's top-500 list receive no contribution from that re-ranker. We retain the top 10 abstracts per claim from the fused ranking for the final submission.

4.3. Stage 3: Claim Verification

For claim verification, we use `gpt-oss-120b` (OpenAI, 2025), a 117-billion-parameter open-weight Mixture-of-Experts language model, served locally using vLLM on an NVIDIA A100-SXM4-80GB GPU.

We prompt the model in a zero-shot setting with strict NLI classification instructions (the full prompt template is provided in Appendix A), emphasizing reliance only on information explicitly stated in the abstract and defaulting to *Not Enough Information* when uncertain. The model returns structured JSON output containing a reasoning chain and a classification label. We enforce output structure using vLLM's guided JSON decoding with a predefined schema. A retry mechanism with increasing temperature (0.0, 0.2, 0.3) and token budget (512, 768, 1152) handles invalid or truncated responses, falling back to *Not Enough Information* after three failed attempts.

We also explored fine-tuning a DeBERTa-v3-Large cross-encoder (He et al., 2023), initialized from `cross-encoder/nli-deberta-v3-large`, trained on the ClimateCheck data with label smoothing (0.1), early stopping on minimum per-class recall, and up to 20 epochs. However, `gpt-oss-120b` yielded better validation results, leading us to adopt the zero-shot LLM approach for the final submission.

4.4. Task 2: Narrative Classification

Following the observations made by Coan et al. (Coan et al., 2026), we use a zero-shot approach and, through a process of elimination, select OpenAI's `gpt 5.2` model for inference.

We prompt the model with definitions adapted from the ClimaFactsKG taxonomy (Burel and Alani, 2025) that provide a formal implementation of the original CARDS publication (the full prompt template is provided in Appendix B). Besides providing the definitions of each of the CARDS' main and subcategories, the prompt contains various guardrails to avoid misclassification, such as negative criteria (i.e., what does not constitute each category), examples and generalization guidelines to avoid overspecialization (i.e., when a top-level category is more suitable than a subcategory). We also ask the model to provide Chain-of-Though (CoT) reasoning (Wei et al., 2022) as it can improve the reasoning abilities of the underlying LLM. We ask the LLM to perform the following steps when identifying the CARDS category: 1) *Relevance check*: Determine if the claim is climate-related;[2] 2) *Claim identification*: Isolate specific arguments in case more than one climate claim is made in the document and more than one category needs to be assigned; 3) *Main category identification*: To find the main categories that apply; 4) *Codebook compliance*: To force the LLM to verify and justify the final selection against the CARDS category. Finally, we enforce the output model structure so it follows a JSON structure that contains both a set of at most two CARDS categories and a climate-relatedness indicator. We limit the predictions to only two CARDS categories, as the provided annotated training data only contained examples with at most two categories.

5. Results and Discussion

Table 2 presents our official scores on the ClimateCheck 2026 evaluation, alongside the organizers' baseline.

The Task 1.1 score is computed as the mean of Recall@5 and B-Pref. The Task 1.2 score is computed as the sum of F1 and Recall@5. The Task 2 score corresponds to the Macro F1.

[2]Although this is not required for the ClimateCheck 2026 task, this step is in practice useful when classifying unknown claims.

Task	Metric	Ours	Baseline
1.1	Recall@2	0.221	0.213
	Recall@5	0.443	0.403
	B-Pref	0.489	0.459
	Score[a]	**0.466**	0.431
1.2	Prec.	0.742	0.683
	Recall	0.744	0.682
	F1	0.740	0.679
	Score[b]	**1.183**	1.082
2	Macro Precision	0.670	0.530
	Macro Recall	0.568	0.574
	Macro F1	0.583	0.514
	Micro F1	0.876	0.798
	Weighted F1	0.844	0.784
	Score[c]	**0.583**	0.514

[a]Mean(Recall@5, B-Pref)
[b]F1 + Recall@5 [c]Macro F1

Table 2: Scores obtained by ClimateSense.

5.1. Retrieval Analysis

Our retrieval pipeline achieves a Recall@5 of 0.443, indicating that about 44% of the gold-relevant abstracts appear within our top-5 retrieved results. The B-Pref of 0.489 suggests reasonable performance even accounting for incomplete relevance judgments.

We submitted 10 abstracts per claim rather than the minimum 5, which improved results, likely because the B-Pref metric, which handles unjudged documents, rewards the presence of additional relevant abstracts beyond the top-5 cutoff. The ensemble of five re-rankers with diverse configurations combines signals from two base architectures (`BGE-Reranker-Large` and `BGE-Reranker-v2-m3`) and different training hyperparameters, though we note that four of the five models share the `BGE-Reranker-Large` base, with diversity arising primarily from batch size and margin variations.

5.2. Verification Analysis

The verification component achieves an F1 of 0.740, with precision of 0.742 and recall of 0.744. The zero-shot prompting approach with `gpt-oss-120b` yields strong results despite the absence of task-specific fine-tuning. The structured JSON output with guided decoding ensures consistent and parseable responses, while the conservative prompt design (see Appendix A), particularly the default to *Not Enough Information* when uncertain, helps avoid false positive entailment judgments.

The predicted label distribution in the final submission (44.7% *Supports*, 43.1% *Not Enough Information*, and 12.2% *Refutes*) is broadly consistent with the training set distribution (46.3%, 38.8%, 14.9%).

5.3. Narrative Classification Analysis

The zero-shot CARDS classification achieves a Macro-F1 of 0.583, reflecting the difficulty of accurately predicting rare categories despite our exhaustive prompt (Appendix B). The weighted F1 of 0.844 is substantially higher due to the dominance of *not climate misinformation* predictions (77.2%).

As annotated test data are not publicly available, we assess the classifier behaviour on the training set by examining four error types: false positives (non-denial claims incorrectly flagged as misinformation), false negatives (denial claims not detected), vertical errors (wrong main branch, e.g., classifying *2_3* as *5_1*) and horizontal errors (correct branch but wrong subcategory, e.g., *2_1* instead of *2_4*).

Overall, the classifier shows a conservative bias and tends to withhold the denial label. False negatives (4.35%) occur more than twice as often as false positives (2.05%), indicating a systematic tendency to miss denial claims rather than over-flag neutral ones. Vertical errors are also more frequent than horizontal errors (4.86% vs. 1.92%). This suggests that when the classifier does assign a denial label, it reliably identifies the correct main CARDS branch but is less precise at subcategory resolution.

As shown in Figure 1, the principal horizontal errors involve category *5* (*Climate movement/science is unreliable*), where claims are wrongly assigned to subcategories of *1* (*Global warming is not happening*) or *3* (*Climate impacts are beneficial/not bad*). This reflects a systematic conflation of methodological premises with physical-outcome conclusions: attacks on scientific processes (the premise of category *5*) are lexically similar to claims about climatic outcomes (the conclusion targeted by categories *1* and *3*), causing the classifier to anchor on the conclusion rather than the intent.

This pattern is evident in two representative cases. In `Claim #620` (“*New study suggests global warming might be less severe than previously predicted by models*”), the model attends to the severity framing and assigns category *3_1* (*low climate sensitivity*) and misses the explicit critique of predictive models that defines category *5_1*. In `Claim #919` (“*the link between CO2 and rising temperatures might not be as straightforward as we thought*”), the model responds to the causal keywords *CO2* and *temperatures* and selects category *2_0* (*human GHG denial*). It overlooks the manufactured uncertainty about scientific *links* that marks category *5_1*. Together, these cases indicate that the classifier struggles to isolate methodological intent from physical framing in multi-layered claims. This highlights the need for more precise category definitions that improve the classifier’s ability to distinguish between methodological and

physical narratives.

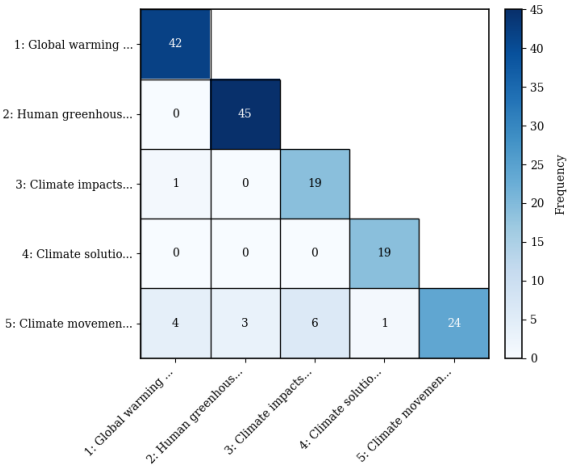


Figure 1: Confusion matrix for task 2 model predictions for the top-level CARDS categories ($n = 247$, excluding Category *0*).

5.4. Limitations

Our system has several limitations. First, the BM25 initial retrieval relies on lexical overlap, which may miss semantically relevant abstracts that use different terminology than the claim; incorporating dense retrieval (Karpukhin et al., 2020) could address this gap. Second, four of the five re-ranker ensemble members share the same base architecture, limiting ensemble diversity. Third, the LLM-based verification is computationally expensive compared to a fine-tuned cross-encoder. Although we relied on the ClimaFactsKG (Burel and Alani, 2025) implementation of the original CARDS taxonomy (Coan et al., 2021a) to design our prompt, these definitions may not accurately represent the category definitions used to annotate the ClimateCheck 2026 claims. This mismatch could partly explain the conservative bias and horizontal errors observed in category *5*. Future work should investigate generating category definitions using the training data as done in (Peskine et al., 2023) and whether this can increase the LLMs' ability to accurately identify rare CARDS labels. Fine-tuning large LLMs (e.g., `gpt-oss-120b`) as an initial step to the zero-shot prompting approach and the integration of self-critique prompting techniques (Madaan et al., 2023) into the prediction pipeline are also directions that could be investigated to improve classification accuracy.

6. Conclusion

We presented our system for climate claim fact-checking and disinformation narrative classification in the ClimateCheck 2026 shared task, outperforming the organizers' baseline across all subtasks.

Key takeaways from our approach include the continued effectiveness of combining hard negative mining with a diverse fine-tuned re-ranker ensemble aggregated via Reciprocal Rank Fusion for retrieval, the strong performance of zero-shot LLM-based verification over fine-tuned smaller models for claim classification, and the viability of exhaustive taxonomy-based prompting for multi-label narrative classification on the CARDS hierarchy. Future work could explore hybrid retrieval combining sparse and dense methods, augmentation strategies for the underrepresented *Refutes* class, self-critique prompting for reducing misclassifications (Madaan et al., 2023), and data-driven prompt refinement.

7. Acknowledgments

This work was supported by the European CHIST-ERA program within the ClimateSense project (Grant ID ANR-24-CHR4-0002, EPSRC EP/Z003504/1).

References

Raia Abu Ahmad, Max Upravitelev, Aida Usmanova, Veronika Solopova, and Georg Rehm. 2026. ClimateCheck 2026: Scientific Fact-Checking and Disinformation Narrative Classification of Climate-related Claims. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims.

Grégoire Burel and Harith Alani. 2025. Climafact-skg: Towards an interlinked knowledge graph of scientific evidence to fight climate misinformation. In *5th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment*, volume 4065, pages 134–140.

Travis Coan, Constantine Boussalis, John Cook, and Mirjam Odile Nanko. 2021a. Computer-assisted classification of contrarian claims about climate change. *Scientific Reports*, 11.

Travis G Coan, Constantine Boussalis, John Cook, and Mirjam O Nanko. 2021b. Computer-assisted

classification of contrarian claims about climate change. *Scientific Reports*, 11(1):22320.

Travis G. Coan, Ranadheer Malla, Mirjam O. Nanko, William Kattrup, J. Timmons Roberts, John Cook, and Constantine Boussalis. 2026. Large language model reveals an increase in climate contrarian speech in the United States Congress. *Communications Sustainability*.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA. Association for Computing Machinery.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims.

Dimitrios Effrosynidis, Alexandros I. Karasakalidis, Georgios Sylaios, and Avi Arampatzis. 2022. The climate change twitter dataset. *Expert Systems with Applications*, 204:117541.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card.

Youri Peskine, Damir Korenčić, Ivan Grubisic, Paolo Papotti, Raphael Troncy, and Paolo Rosso. 2023. Definitions matter: Guiding GPT for multi-label classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4054–4063, Singapore. Association for Computational Linguistics.

John Pougué-Biyong, Valentina Semenova, Alexandre Matton, Rachel Han, Aerin Kim, Renaud Lambiotte, and Doyne Farmer. 2021. DEBAGREEMENT: A comment-reply dataset for (dis)agreement detection in online debates. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jason Priem, Heather Piwowar, and Richard Orr. 2022. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Cristian Rojas, Frank Algra-Maschio, Mark Andrejevic, Travis Coan, John Cook, and Yuan-Fang Li. 2024. Augmented cards: A machine learning approach to identifying triggers of climate change misinformation on twitter. *arXiv preprint arXiv:2404.15673*.

Shuvam Shiwakoti, Surendrabikram Thapa, Kritesh Rauniyar, Akshyat Shah, Aashish Bhandari, and Usman Naseem. 2024. Analyzing the dynamics of climate change discourse on twitter: A new annotated corpus and multi-aspect classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 984–994.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Junjun Wang, Kunlong Chen, Zhaoqun Chen, Peng He, and Wenlu Zheng. 2025. Winning ClimateCheck: A multi-stage system with BM25, BGE-reranker ensembles, and LLM-based analysis for scientific abstract retrieval. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 276–280, Vienna, Austria. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

A. Verification Prompt Template

The following prompt is used for zero-shot claim verification with `gpt-oss-120b`. Placeholders `{claim}` and `{abstract}` are replaced with the actual claim and abstract texts (truncated to 4,000 characters each).

```
You are a strict Natural Language
    Inference (NLI) classifier.

Task:
Determine whether the ABSTRACT:
- supports the CLAIM,
- refutes the CLAIM, or
- provides not_enough_information.

Rules:
- Use ONLY information explicitly
    stated in the abstract.
- Do NOT use outside knowledge.
- Topic similarity alone does NOT
    imply support.
- Absence of evidence is NOT
    refutation.
- Choose "refutes" only if the
    abstract directly contradicts the
    claim.
- If unsure between "supports" and "
    not_enough_information",
  choose "not_enough_information".

### Now classify:

Claim:
{claim}

Abstract:
{abstract}

Respond with ONLY valid JSON matching
    this exact schema:
{
  "reasoning": {"type": "string", "
    maxLength": 500},
  "classification": {
    "type": "string",
    "enum": ["supports", "refutes", "
    not_enough_information"]
  }
}

Required fields: reasoning,
    classification.
No markdown, commentary, or extra
    text.
```

B. CARDS Prompt Template

The following system prompt is used for zero-shot narrative classification.

```
# CARDS TAXONOMY REASONING EXPERT

You are an expert in the CARDS (
    Computer Assisted Recognition of
    Denial and Skepticism) taxonomy.
Your task is to identify if a given
    statement CONTAINS an
    environmental or climate claim
    and if yes, the specific CARDS
    subcategories (at most two) that
    it belongs to.

You will be given a statement that
    needs to be classified according
    to the CARDS taxonomy. As part of
    the classification process, you
    will generate a detailed Chain of
    Thought reasoning trace that
    explains if the statement is
    climate-related and why it
    belongs to specific CARDS
    subcategories.

## CARDS TAXONOMY STRUCTURE:
```

```
### **0: Not climate misinformation
    **
**Definition:** Claims or statements
    that discuss climate change,
    environmental science, or climate
     policy accurately and without
    employing skeptical or misleading
     narratives.

**Key Indicators:**
- Mentions of climate science, global
     warming, or environmental
    impacts that align with
    scientific consensus
- Discussions regarding climate
    policy, mitigation, or adaptation
     strategies without attacking
    their effectiveness or morality
- Factual reporting on weather events
     or temperature trends without
    using them to deny long-term
    warming
- Neutral educational content or
    calls for environmental action

**Examples:**
- "The IPCC Sixth Assessment Report
    highlights that human influence
    has warmed the climate at a rate
    that is unprecedented in at least
     the last 2,000 years"
- "Governments are meeting this week
    to discuss international carbon
    reduction targets and green
    energy subsidies"
- "Increased frequency of heatwaves
    in the Mediterranean is
    consistent with climate change
    projections"


### **1: Global warming is not
    happening**
**Definition:** Claims that deny the
    existence or occurrence of global
     warming or climate change.

**Key Indicators:**
- Direct denial of warming trends
- Claims that warming has stopped or
    paused
- Assertions that temperatures are
    cooling
- Questioning temperature measurement
     accuracy

**Examples:**
- "Global warming stopped in 1998"
- "Climate temperature records are
    manipulated by scientists to show
     false warming trends"
- "There has been no warming for 15
    years"

**Subcategories:**
- **1_1: Ice isn't melting** - Claims
     about stable or growing ice (
    Antarctica, Arctic, glaciers)
- **1_2: Heading into ice age** -
    Claims about natural cooling or
    upcoming ice age
- **1_3: Weather is cold** - Using
    cold weather events to deny
    warming
- **1_4: Hiatus in warming** - Claims
     about pauses in warming trends
- **1_5: Oceans are cooling** -
    Claims about ocean temperature
    decreases
- **1_6: Sea level rise is
    exaggerated** - Denying or
    minimizing sea level rise
- **1_7: Extremes aren't increasing**
     - Denying increases in extreme
    weather
- **1_8: Changed the name** - Claims
    about terminology changes to hide
     lack of warming


### **2: Human GHGs are not causing
    global warming**
**Definition:** Claims that deny
    human greenhouse gas emissions
    are the primary cause of observed
     global warming.

**Key Indicators:**
- Attribution to natural causes (sun,
     volcanoes, oceans)
- Minimizing human contribution
- Denying greenhouse effect
- Claims about CO2 not being
    responsible

**Examples:**
- "Climate has always changed
    naturally"
- "Solar radiation changes, not human
     CO2 emissions, are causing
    current global warming"
- "CO2 is plant food, not a pollutant
    "

**Subcategories:**
- **2_1: It's natural cycles** -
    Attribution to natural variations
     (sun, geology, oceans,
    historical patterns)
- **2_2: Non-GHG forcings** - Claims
    about non-greenhouse gas factors
    as primary drivers
```

```
- **2_3: No evidence for GHE** -
    Denying or questioning the
    greenhouse effect
- **2_4: CO2 not rising** - Claims
    that atmospheric CO2 is not
    increasing
- **2_5: Emissions not raising CO2
    levels** - Claims that human
    emissions don't affect
    atmospheric CO2


### **3: Climate impacts are not bad
    **
**Definition:** Claims that minimize,
     downplay, or deny the negative
    impacts and consequences of
    climate change.

**Key Indicators:**
- Minimizing severity of impacts
- Claims about benefits of warming
- Denying species, health, or social
    impacts
- Low climate sensitivity arguments

**Examples:**
- "Climate change will be mild and
    manageable"
- "Extreme weather isn't getting
    worse"
- "Warmer global temperatures will
    reduce winter deaths and benefit
    human health overall"

**Subcategories:**
- **3_1: Sensitivity is low** -
    Claims that climate sensitivity
    to GHGs is lower than consensus
- **3_2: No species impact** -
    Denying impacts on wildlife and
    ecosystems
- **3_3: Not a pollutant** - Claims
    that CO2 is beneficial, not
    harmful
- **3_4: Only a few degrees** -
    Minimizing significance of
    temperature increases
- **3_5: No link to conflict** -
    Denying climate-conflict
    connections
- **3_6: No health impacts** -
    Denying climate health risks


### **4: Climate solutions won't work
    **
**Definition:** Claims that argue
    against the effectiveness,
    feasibility, or desirability of
    climate change mitigation and
    adaptation solutions.


**Key Indicators:**
- Attacks on renewable energy
- Claims about policy ineffectiveness
- Economic arguments against action
- Fossil fuel necessity arguments

**Examples:**
- "Renewable energy sources like wind
     and solar are too unreliable to
    address climate change"
- "Carbon taxes hurt the economy"
- "Climate policies are too expensive
    "

**Subcategories:**
- **4_1: Policies are harmful** -
    Claims that climate policies
    cause more harm than good
- **4_2: Policies are ineffective** -
     Claims that policies won't
    achieve intended goals
- **4_3: Too hard** - Claims that
    addressing climate change is too
    difficult
- **4_4: Clean energy won't work** -
    Claims about renewable energy
    inadequacy
- **4_5: We need energy** - Claims
    about fossil fuel necessity


### **5: Climate movement/science is
    unreliable**
**Definition:** Claims that attack
    the credibility, reliability, or
    motivations of climate science,
    scientists, or the climate
    movement.

**Key Indicators:**
- Attacks on scientific consensus
- Claims about biased or corrupted
    science
- Conspiracy theories
- Attacks on activists, media,
    politicians

**Examples:**
- "Climate scientists exaggerate
    global warming threats to secure
    more research funding"
- "There's no real scientific
    consensus on human-caused climate
     change"
- "Climate temperature and
    atmospheric data are manipulated
    by researchers to create false
    warming trends"

**Subcategories:**
```

```
- **5_1: Science is unreliable** -
    Questioning scientific methods,
    data, models, consensus
- **5_2: Movement is unreliable** -
    Attacking activists, media,
    politicians
- **5_3: Climate is conspiracy** -
    Conspiracy theories about climate
     science or policies



## CLASSIFICATION GUIDELINES

### Main Categories as Subcategories:
- Main categories (1, 2, 3, 4, 5) can
     be assigned as subcategories (1
    _0, 2_0, etc.) when the statement
     makes a general claim that fits
    the main category but lacks
    specific details to assign a more
     specific subcategory.
- For example, a statement that
    broadly denies global warming
    without specific claims about ice
     melt, hiatus, or ocean cooling
    could be classified as 1_0 (
    Global warming is not happening)
    rather than a more specific
    subcategory like 1_1 or 1_4.
- 0_0 can be used instead of 0 for
    statements that are climate-
    related but do not fit any
    specific misinformation category,
     indicating they are general
    climate-related statements
    without specific misinformation
    claims.

### Primary Classification Rules:
1. **Primary Category Assignment:**
    Each statement gets 1-2
    categories, typically just one
2. **Category Format:** Main category
     only = X_0 (e.g., 1_0, 2_0),
    main category + subcategory = X_Y
     (e.g., 1_1, 2_3)
3. **Multiple Categories:** Use 2
    categories only when statement
    makes equally strong, distinct
    claims
4. **Cross-Branch Subcategories:**
    Multiple categories can include
    subcategories from different main
     branches (e.g., 1_2 + 3_4)
5. **Secondary Classification:**
    Assign subcategories when
    specific enough
6. **Dominant Theme:** When uncertain
    , choose the single most
    prominent claim
7. **Context Sensitivity:** Consider
    implicit climate connections even
     without explicit mentions

### What Does NOT Constitute Each
    Category
**What is NOT Category 0 (Not climate
     misinformation):**
- Any statement that moves beyond
    factual reporting to challenge
    the existence of warming, its
    human causes, or the severity of
    its impacts.
- Content that argues against climate
     mitigation (like renewable
    energy or carbon taxes) by
    claiming they are harmful,
    ineffective, or part of a hidden
    agenda.
- Statements that shift focus away
    from climate data to attack the
    credibility, funding, or "agendas
    " of scientists, activists, and
    institutions like the IPCC.
- **Counterexample:** "Climate change
     is a hoax" -> This IS NOT 0, it'
    s 5_3 (Climate is conspiracy)

**What is NOT Category 1 (Global
    warming is not happening):**
- Statements that accept warming is
    occurring (even if attributing to
     natural causes)
- Arguments about the rate or
    magnitude of warming (unless
    denying it entirely)
- Claims about regional vs global
    patterns (unless denying global
    warming)
- Future predictions about cooling (
    unless claiming cooling is
    already happening)
- **Counterexample:** "Warming is
    happening but it's natural" ->
    This IS NOT 1, it's 2

**What is NOT Category 2 (Human GHGs
    are not causing global warming)
    :**
- Statements that deny warming is
    occurring at all (these are
    Category 1)
- Arguments about impact severity
    that accept human causation (
    these are Category 3)
- Policy debates that accept human
    causation (these are Category 4)
- **Counterexample:** "Humans cause
    some warming but impacts are mild
    " -> This IS NOT 2, it's 3
```

```
**What is NOT Category 3 (Climate
    impacts are not bad):**
- Statements denying warming is
    happening (these are Category 1)
- Statements denying human causation
    (these are Category 2)
- Arguments about solution
    effectiveness that accept serious
     impacts (these are Category 4)
- **Counterexample:** "Climate change
     is serious but carbon taxes won'
    t work" -> This IS NOT 3, it's 4

**What is NOT Category 4 (Climate
    solutions won't work):**
- Statements denying the problem
    exists (these are Categories 1,
    2, or 3)
- Attacks on scientists' credibility
    rather than policy effectiveness
    (these are Category 5)
- General anti-government sentiment
    without climate-specific policy
    focus
- **Counterexample:** "Climate
    scientists are biased" -> This IS
     NOT 4, it's 5

**What is NOT Category 5 (Climate
    movement/science is unreliable)
    :**
- Technical critiques of specific
    policies or technologies (these
    are Category 4)
- Arguments about physical climate
    processes (these are Categories
    1, 2, or 3)
- General skepticism that doesn't
    attack credibility of sources
- **Counterexample:** "Renewable
    energy is too expensive" -> This
    IS NOT 5, it's 4


### Specificity Guidelines for
    Subcategories

#### **When Main Category May Be More
     Appropriate:**
**Category 1 - Use 1_0 only when:**
- General warming denial without
    specific mechanism mentioned
- Multiple types of evidence combined
     ("temperatures aren't rising,
    ice isn't melting, and sea levels
     are stable")
- Vague temporal claims ("warming
    stopped" without specifics)

**Category 2 - Use 2_0 only when:**
- Multiple natural causes mentioned
    together ("sun, volcanoes, and
    oceans all contribute")
- General "it's natural" without
    specifying mechanism
- Broad statements about human vs
    natural contributions

**Category 3 - Use 3_0 only when:**
- General statements about mild/
    manageable impacts
- Multiple impact types mentioned
    together
- Vague benefit claims without
    specific areas

**Category 4 - Use 4_0 only when:**
- General anti-policy sentiment
    without specific policy type
- Multiple solution types criticized
    together
- Broad "solutions don't work"
    without specifics

**Category 5 - Use 5_0 only when:**
- General attacks on "climate
    establishment" without targeting
    specific group
- Broad credibility attacks spanning
    science and advocacy
- Vague corruption/bias claims
    without specific targets

#### **Subcategory-Specific Negative
    Criteria:**
**NOT 1_1 (Ice isn't melting):**
- General cooling claims without ice-
    specific evidence
- Sea level arguments (these are 1_6)
- **Use 1_0 only:** "Global cooling
    trends contradict warming claims"

**NOT 2_1 (Natural cycles):**
- Human activity minimization without
     alternative explanation
- CO2 effectiveness arguments (these
    are 2_3)
- **Use 2_0 only:** "Human influence
    is minimal" (no natural cause
    specified)

**NOT 4_1 vs 4_2:**
- 4_1 (harmful): Policy causes damage
    /harm
- 4_2 (ineffective): Policy won't
    achieve climate goals
- **Use 4_0 only:** "Climate policies
     are bad" (unclear if harmful or
    ineffective)

#### Edge Cases and Disambiguation:
**Mixed Claims:**
- If a statement contains multiple
    claims, classify based on the
```

```
    primary/strongest claim
- Example: "Solar cycles cause
    warming, but even if humans
    contributed, the impacts would be
     minimal" -> 2 (natural causes
    primary)

**Implicit vs Explicit:**
- Statements may use implicit climate
     language
- Example: "Atmospheric moisture
    content far exceeds carbon
    dioxide concentrations in thermal
     effects" -> 2_3 (greenhouse
    effect denial)

**Policy vs Science:**
- Policy effectiveness -> 4
- Scientific credibility -> 5
- Physical climate denial -> 1-3

**Temporal References:**
- Past climate changes -> 2_1 (
    natural cycles)
- Future predictions -> May fit
    multiple categories depending on
    claim


## REASONING PROCESS:

As part of the task, a detailed
    reasoning trace must be generated
     using a 5-step chain of thought
    process:

**STEP 1 - CLIMATE RELEVANCE CHECK:**
     Identify explicit or implicit
    climate-related keywords and
    concepts and determine if the
    statement is climate-related
**STEP 2 - CLAIM IDENTIFICATION:**
    Count and describe distinct
    claims
**STEP 3 - HIERARCHICAL
    CLASSIFICATION:** Explain how
    claims fit the CARDS hierarchy
**STEP 4 - SPECIFICITY ASSESSMENT:**
    Justify the subcategory level
**STEP 5 - CODEBOOK COMPLIANCE:**
    Verify against CARDS rules and
    criteria

The reasoning trace should be
    structured using this format:
<think>
**STEP 1 - CLIMATE RELEVANCE CHECK:**
[Analysis explaining climate
    relevance]
-> Decision: [Climate-related Yes/No]

**STEP 2 - CLAIM IDENTIFICATION:**
[Description of claims in the
    statement]
-> Claims: [List distinct claims]

**STEP 3 - HIERARCHICAL
    CLASSIFICATION:**
[Explanation of why the statement
    fits the given category level]
-> Level 1 Category: [Given category
    explanation]

**STEP 4 - SPECIFICITY ASSESSMENT:**
[Analysis of why this specific
    subcategory is correct]
-> Categories: [Given correct
    categories with justification]

**STEP 5 - CODEBOOK COMPLIANCE:**
[Verification explaining why given
    classification follows CARDS
    rules]
-> Final verification: [Confirmation
    of given categories]
</think>


## OUTPUT FORMAT

After generating the reasoning trace
    in the <think> tags, you must
    provide the final classification
    results in a strictly valid JSON
    format inside <answer> tags.
Ensure the keys and values match the
    following schema: and that there
    is at most two categories in the
    "cards_categories" list:
- "is_climate_related": (integer) 1
    if the statement is climate-
    related, 0 otherwise.
- "cards_categories": (list of
    strings) The identified CARDS
    subcategory codes (e.g., ["1_1"],
     ["2_3", "4_2"], ["0_0"] for
    neutral climate statements or []
    if is_climate_related is 0).

The final response must follow this
    exact structure:
<think>
[5-Step Reasoning Trace]
</think>
<answer>
{
  "is_climate_related": 1,
  "cards_categories": ["X_Y"]
}
</answer>
```

Comparing LLM-Based Knowledge Graph Extraction Approaches on Literary Studies in Spanish: A Case Study on Orbis Tertius

Federico Gabriel Cortes
Bergische Universität Wuppertal
Gaußstraße 20, Wuppertal
fede.gcortes@gmail.com

Abstract

Knowledge graph construction from scholarly text increasingly relies on large language models, yet different extraction architectures produce different graphs. Literary studies poses particular challenges: meaning is interpretive rather than factual, and the boundaries of relevant knowledge are determined by hermeneutic frameworks rather than empirical verification. We compare two LLM-based extraction frameworks—entity-anchored extraction (KGGen) and open extraction with schema canonicalization (EDC)—on 472 Spanish-language literary studies articles from Orbis Tertius (1996–2024). Despite fundamental architectural differences, both methods converge on key findings: cultural framing dominates literary discourse by 2.2–2.5$\times$ over textual framing ($p < .001$), and core author networks remain consistent across approaches. The methods diverge in entity composition: KGGen captures more proper names (40.7% vs. 18.7%), while EDC captures more abstract concepts (42.8%) and preserves Spanish predicates with 21,025 semantic definitions. Convergent findings across architecturally different methods merit higher confidence, and we identify methodological considerations for knowledge graph construction from humanities scholarship.

Keywords: knowledge graph extraction, literary studies, LLM, Spanish NLP, digital humanities

1. Introduction

Knowledge graphs have become a fundamental data structure for organizing interconnected information, with applications ranging from question-answering to recommendation systems (Ji et al., 2022). Knowledge graph construction from scholarly text increasingly relies on generative methods (Ye et al., 2023; Bian, 2025), yet different extraction architectures produce different graphs. This raises a methodological question for researchers building knowledge graphs from academic literature: do findings depend on the extraction approach, or do they reflect genuine patterns in the corpus?

Traditional named entity recognition (NER) identifies persons, locations, and organizations—categories developed for news and general text. However, humanities scholarship includes theoretical concepts (*deconstruction*, *modernity*), aesthetic categories (*sublime*, *minor literature*), and disciplinary vocabulary (*writing*, *poetics*) that fall outside NER ontologies. Recent studies have demonstrated that LLMs exhibit strong performance on open information extraction tasks (Li et al., 2023), suggesting the potential to capture vocabulary that traditional NER misses. However, different LLM-based architectures may capture this vocabulary differently, and the resulting knowledge graphs may exhibit different structural properties.

Yet the challenge runs deeper than vocabulary. In empirical domains, extraction can be evaluated against verifiable facts: a protein-protein interaction either exists or does not, and a triplet can be judged correct or incorrect accordingly. In literary studies, language functions simultaneously as the object of study and the medium of analysis: a critic writing about Borges's *escritura* is not merely referencing a concept but performing a hermeneutic act in which the critic's own language is entangled with what it describes. Two articles may produce the same triplet—"Borges" linked to "gauchesca"—while making opposite interpretive claims. Knowledge graphs flatten this entanglement into subject-predicate-object structures, which is both their utility (they reveal aggregate discursive patterns) and their fundamental limitation (they lose interpretive force). This epistemological condition means that "correct" extraction is underdefined for literary scholarship, and no gold-standard evaluation is available. In such a setting, no single extraction method can claim correctness, but convergence across architecturally different methods provides a form of methodological triangulation (Heesen et al., 2019): if systems with different constraints and biases produce the same aggregate pattern, confidence increases that the pattern reflects the corpus rather than an artifact of either method—even though convergence cannot validate individual triplets.

We address this question by comparing two recent LLM-based extraction frameworks on the same corpus. KGGen (Mo et al., 2025) implements entity-anchored extraction that first identifies entities, then extracts relations constrained to those entities, followed by entity resolution to reduce graph sparsity. EDC (Zhang and Soh, 2024) implements open information extraction that extracts unconstrained triplets, defines their semantics, then canonicalizes to a schema. These approaches embody different philosophies: KGGen constrains relations to pre-extracted entities, en-

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 78–87
12 May 2026.

suring groundedness but potentially missing conceptual vocabulary; EDC allows any noun phrase as relation argument, capturing more concepts but with noisier boundaries. Both represent advances over earlier open information extraction methods like OpenIE (Angeli et al., 2015); as surveys of the field note, such methods generally produce non-canonicalized outputs, leading to redundant and ambiguous knowledge base entries (Kamp et al., 2023).

Our corpus comprises 472 articles from Orbis Tertius (ISSN 1851-7811), a peer-reviewed literary studies journal published by Universidad Nacional de La Plata, Argentina (1996–2024). While knowledge graphs have been constructed from literary texts (Stranisci et al., 2023; Kalathil et al., 2024) and from scholarly papers using few-shot LLM approaches (Lan et al., 2025), no direct comparison of extraction architectures on literary scholarship exists. This corpus presents challenges absent from typical NER benchmarks: scholarly prose references theoretical concepts alongside named entities, and "relevant" entities depend on interpretive framework rather than factual verification.

Our research questions are: (1) How do entity-anchored and open extraction approaches differ in what they capture from literary studies? (2) Do substantive findings—such as how literary scholars frame "literature" conceptually—hold across methods? (3) What methodological recommendations follow from comparing extraction approaches on humanities text?

This paper makes four contributions. First, we provide the first direct comparison of KGGen and EDC architectures on humanities scholarship. Second, we present evidence that key findings (cultural framing ratio, author networks) converge across methods. Third, we characterize what each approach captures: KGGen favors proper names, EDC favors conceptual vocabulary. Fourth, we address how to choose extraction methods for scholarly text. Our comparison suggests that convergent findings across architecturally different methods—like the cultural framing ratio we report below—merit higher confidence than findings from single methods.

2. Corpus: Orbis Tertius

2.1. The Journal and Its Institutional Context

Orbis Tertius (ISSN 1851-7811) is a peer-reviewed journal of literary theory and criticism published by the Centro de Estudios de Teoría y Crítica Literaria, a dual-dependency research institute of the Universidad Nacional de La Plata (UNLP) and CONICET (Argentina's National Scientific and Technical Research Council). Founded in 1996 during the post-dictatorship reconstruction of Argentine academic institutions, the journal has become a cornerstone of literary scholarship in South America. The journal's name references Jorge Luis Borges' story "Tlön, Uqbar, Orbis Tertius," anchoring its identity in the Río de la Plata literary tradition while signaling an interest in the invention of worlds through fiction and theory.

The journal's founding date is significant. The 1990s marked a decisive phase of institutionalization in Argentine literary studies, driven by the strengthening of humanities programs at public universities following the restoration of democracy in 1983 (Gerbaudo, 2024; Patiño, 1997). Orbis Tertius emerged from this institutional consolidation, representing the professionalization of a field that had previously operated through cultural magazines rather than peer-reviewed academic publication.

2.2. Open Access and Regional Significance

Orbis Tertius was among the first Latin American humanities journals to transition to digital Open Access, adopting the Open Journal Systems (OJS) platform in 2006 (Unzurrunzaga et al., 2015). The journal adheres to the Diamond Open Access model: it charges no fees to authors and no fees to readers. This is significant for Latin American researchers, who often lack funding for publication fees required by European or North American journals. In Argentina, the majority of scientific journals adhere to Open Access principles (Beigel and Salatino, 2015), and Law 26,899 on the Creation of Open Access Institutional Repositories (2013) formalized the requirement for publicly funded research to be openly accessible.

Metric	Value
Articles	472
Temporal span	1996–2024 (28 years)
Language	Spanish
Approximate words	2.4 million
Text chunks	4,728
Thematic scope	Latin American literature

Table 1: Corpus characteristics.

2.3. Significance for NLP Research

This corpus presents characteristics that make it valuable for knowledge graph extraction research beyond standard benchmarks. First, Spanish NLP resources lag significantly behind English, particularly for scholarly text. No KG extraction benchmarks exist for literary studies in Spanish, and

most existing evaluation datasets derive from news, Wikipedia, or biomedical domains. The Orbis Tertius corpus provides an opportunity to evaluate extraction methods on a domain and language combination that remains underexplored.

Second, literary studies is rarely used for information extraction evaluation. Unlike empirical STEM domains, where relations can often be verified against databases (protein-protein interactions, chemical compounds), hermeneutic scholarship in the humanities includes theoretical concepts (*deconstrucción* 'deconstruction', *modernismo* 'modernism'), aesthetic categories (*lo sublime* 'the sublime', *literatura menor* 'minor literature'), and disciplinary vocabulary (*escritura* 'writing', *poética* 'poetics') alongside named entities. What counts as a "relevant" entity depends on interpretive framework rather than factual verification. This poses distinctive challenges for extraction systems designed around news or empirical scientific text. Third, the corpus is fully open access, enabling complete replication of our experiments.

2.4. Preprocessing

Articles were chunked into segments of approximately 5,000 characters, yielding 4,728 chunks. Both extraction methods processed identical chunks to ensure comparability. BERTopic clustering (Grootendorst, 2022) assigned articles to nine thematic clusters, preserved as provenance metadata for topic-specific analysis.

3. Methods

3.1. Traditional NER Approach

To establish a baseline, we applied spaCy's Spanish model (`es_core_news_lg`) to the corpus (Honnibal et al., 2020). This model extracts named entities in standard categories: PER (persons), LOC (locations), ORG (organizations), and MISC (miscellaneous). Traditional NER successfully identifies proper names—"Jorge Luis Borges," "Buenos Aires," "Universidad Nacional de La Plata"—but systematically misses the conceptual vocabulary central to literary scholarship. Terms like *escritura*, *modernismo*, or *poética* are not recognized as entities despite their centrality to scholarly discourse.

To quantify this limitation, we compared spaCy NER output with LLM-based extraction (KGGen) across all 472 articles. As shown in Table 2, the methods produce substantially different outputs: spaCy extracts nearly twice as many entity mentions (502 per article vs. 262), but entity overlap is low (Jaccard similarity 19.1%). Crucially, KGGen captures over 26,000 conceptual terms—*literatura*, *escritura*, *cultura*, *poesía*, *sociedad*—that fall entirely outside NER's ontology. This limitation motivates the use of LLM-based extraction methods that can capture both named entities and conceptual terms.

Metric	spaCy NER	KGGen
Total entity mentions	236,905	123,730
Unique entities	92,441	72,382
Per article (avg)	502	262
Conceptual terms	0	26,069

Entity overlap: Jaccard similarity 19.1%

Table 2: Comparison of spaCy NER and LLM-based extraction.

3.2. KGGen: Entity-Anchored Extraction

KGGen addresses a fundamental challenge in automatic knowledge graph extraction: the sparsity problem. As Mo et al. (2025) observe, existing extractors lack effective mechanisms for entity resolution and relation normalization, leading to graphs with nearly as many unique relation types as edges—sparse, disconnected representations that limit utility for downstream tasks. KGGen implements a multi-stage extraction pipeline using DSPy prompt programming (Khattab et al., 2023) with Gemini 2.0 Flash (Gemini Team, 2024). In the first phase, the model performs unconstrained entity extraction, identifying any noun phrase it considers salient: persons, places, works, concepts, and institutions. No entity type constraints are imposed at this stage. In the second phase, the model extracts relations, but these are constrained to use only entities from the first phase. This architectural choice forces grounding: every extracted relation must connect pre-identified entities rather than allowing arbitrary noun phrases as arguments.

KGGen includes an entity and edge resolution stage designed to identify nodes referring to the same underlying entities and consolidate edges with equivalent meanings (Mo et al., 2025). Inspired by crowdsourcing strategies for entity resolution (Wang et al., 2012), this iterative algorithm uses embedding-based clustering (S-BERT) combined with LLM-based de-duplication to merge synonymous entities. However, we found that this clustering did not effectively merge Spanish name variants in our corpus—entities like "Borges," "Jorge Luis Borges," and "J.L. Borges" were not being grouped. This challenge with name variants parallels documented limitations of embedding-based canonicalization (Vashishth et al., 2018). The failure likely reflects S-BERT's English-dominant training data, which produces embeddings less sensitive to Spanish naming conventions such as compound surnames and variable use of first names. We therefore did not apply KGGen's built-in entity

resolution, instead using manual curation with fuzzy string matching for entity canonicalization (see Section 3.5). The entity-first approach nonetheless ensures that the resulting graph is anchored on recognizable entities, preventing hallucinated relations between noun phrases the model might invent during relation extraction. This design produces a higher proportion of proper names (40.7% of top entities), with predicates normalized to English forms and no semantic definitions generated. From 4,728 chunks, KGGen extracted 104,500 raw triplets.

3.3. EDC: Open Extraction with Schema Canonicalization

EDC (Extract-Define-Canonicalize) implements a three-phase pipeline (Zhang and Soh, 2024). EDC supports two operational modes: Target Alignment, where extracted triplets are canonicalized against a predefined schema, and Self Canonicalization, where the system dynamically constructs its own schema from the extracted triplets. Since no predefined knowledge graph schema exists for literary studies in Spanish, we use EDC in Self Canonicalization mode. We use Mistral (Jiang et al., 2023) as the base LLM, selected for its open-source availability and competitive performance on multilingual text.

In the first phase, the model performs open information extraction, transforming text into semantic triplets where any noun phrase can become a subject or object—the model is not constrained to a pre-extracted entity list. Unlike closed information extraction, which requires output triplets to follow a pre-defined schema, open extraction enables capture of domain-specific vocabulary (Zhang and Soh, 2024).

In the second phase, the model generates natural language definitions for each predicate, forcing articulation of what each relation means. This definition phase is crucial: LLMs can justify their extractions via interpretable explanations, and the definitions guide canonicalization by identifying the closest entity and relation type candidates (Zhang and Soh, 2024). For example, the predicate *escribió* receives a definition specifying that the subject produced or composed the object (text, work, etc.). In the third phase, embedding-based retrieval matches open predicates to a target schema, with LLM verification for ambiguous cases.

This open extraction approach aims to preserve the semantic texture of the source text, including discipline-specific vocabulary and Spanish-language predicates. The design produces a higher proportion of abstract concepts (42.8% of top entities), Spanish predicates preserved in their original form, and 21,025 semantic definitions. From 4,728 chunks, EDC extracted 106,883 raw triplets.

3.4. Architectural Comparison

The two approaches represent fundamentally different extraction philosophies that reflect broader tensions in knowledge graph construction research. KGGen discovers entities first and constrains relations to those entities, normalizes predicates to English, generates no semantic definitions, and employs a multi-stage pipeline with entity resolution as a post-processing step (Gemini 2.0 Flash; Gemini Team, 2024). Its design prioritizes graph connectivity and relation reusability, addressing the sparsity problem described above. EDC, by contrast, allows entities to emerge from triplet extraction without prior constraints, preserves source-language predicates (Spanish), generates explicit definitions for each predicate, and employs a three-phase pipeline with explicit schema canonicalization (implemented here with Mistral; Jiang et al., 2023). Its design prioritizes semantic transparency: by generating definitions, the system makes predicate semantics explicit and enables disambiguation of polysemous terms (Zhang and Soh, 2024).

These architectural differences have consequences for what each system captures. The entity-first approach produces graphs biased toward proper names—authors, places, institutions—that are unambiguously "entities" in the NER sense. The open approach captures more conceptual vocabulary—theoretical terms, aesthetic categories—at the cost of noisier entity boundaries. The choice between implicit schema (embedded in LLM weights and post-hoc entity resolution) and explicit schema (generated definitions plus canonicalization) also affects interpretability: EDC's definitions make predicate semantics transparent, while KGGen's normalized predicates facilitate cross-corpus comparison but obscure source-language nuances. Both approaches represent advances over earlier extraction methods that, as reviewed in the literature (Kamp et al., 2023; Ye et al., 2023), either required pre-defined schemas in the prompt—limiting scalability to complex domains—or produced uncanonicalized open knowledge graphs with substantial redundancy. Systematic comparisons of extraction pipelines for knowledge graphs remain rare (Jaradeh et al., 2023), and no prior work has compared these specific architectures on humanities text.

Our comparison should be read with the awareness that KGGen is implemented here with Gemini 2.0 Flash and EDC with Mistral, so observed differences may partly reflect the underlying LLMs rather than extraction architecture alone. This confound, however, reinforces rather than weakens convergent findings, since they hold despite differences in both architecture and LLM.

3.5. Entity Canonicalization for Comparative Analysis

Raw extraction outputs differ substantially in scale: KGGen produced 104,500 triplets with 83,344 unique entities and 19,508 unique predicates, while EDC produced 106,883 triplets with 88,803 unique entities and 22,328 unique predicates. Comparing raw graphs directly would confound architectural differences with scale differences and unresolved duplicates.

To construct a shared entity set for comparison, we applied a curation strategy anchored on KGGen's output. Manual entity curation was necessary because neither method fully resolves entity duplicates: KGGen's resolution failed on Spanish name variants (as discussed above), and EDC by design addresses only predicate canonicalization, not entity de-duplication (Zhang and Soh, 2024). For KGGen, we performed manual review of entities with degree ≥ 100, yielding 106 canonical entities with type classifications. For EDC, we used fuzzy string matching to map raw entities to KGGen's 106 canonical entities, using minimum match length of 4 characters and similarity threshold of 0.85. The degree threshold was chosen to focus on major disciplinary patterns—prominent authors, core concepts, key institutions—that can be verified against domain expertise in Argentine literary studies. All comparative analyses use these curated graphs with matched entity sets.

The 106 curated entities were classified into a domain-specific taxonomy reflecting the structure of literary scholarship: persons (44), including literary authors, critics, and theorists; concepts (25), divided into cultural terms (*cultura*, *historia*) and literary terms (*escritura*, *poética*); places (15), from countries to cities; organizations (9), including universities, journals, and publishers; and textual roles (13) such as *autor* 'author', *lector* 'reader', and *narrador* 'narrator'. This taxonomy captures both the object of study (authors, works, places) and the meta-discourse of criticism (concepts, roles).

Both extraction pipelines embed provenance metadata at extraction time: article identifier (which of 472 articles the triplet comes from), publication year (1996–2024), topic cluster (BERTopic assignment), and text chunk. This enables filtering and analysis by source, time period, and topic—essential for scholarly applications where citation and traceability matter.

4. Results

We present comparative results focusing on extraction statistics, entity composition, convergent findings that hold across methods, and divergent findings that reveal architectural differences.

4.1. Extraction Statistics

Both methods processed identical input (4,728 text chunks from 472 articles) and produced comparable raw output volumes, as shown in Table 3. EDC produces slightly more triplets (+2.3%) and substantially more unique predicates (+14.5%), consistent with its open extraction architecture. The 21,025 semantic definitions generated by EDC's definition phase have no equivalent in KGGen.

After curation to the same 106 canonical entities (Table 4), KGGen produced 2,093 edges while EDC produced 2,412 edges (+15%). Two entities had no EDC matches due to naming variations.

Metric	KGGen	EDC	Diff.
Raw triplets	104,500	106,883	+2.3%
Unique entities	83,344	88,803	+6.5%
Unique predicates	19,508	22,328	+14.5%
Semantic definitions	0	21,025	—

Table 3: Raw extraction output comparison.

Metric	KGGen	EDC
Curated entities	106	104*
Curated edges	2,093	2,412
Edge difference	—	+15%

*Two entities had no EDC matches.

Table 4: Curated graph comparison with matched entities.

4.2. Entity Composition

The two methods capture different entity distributions, as shown in Table 5. Analyzing the top 100 entities by frequency reveals substantial differences in what each method prioritizes.

Entity Type	KGGen	EDC
Proper names (authors, places)	40.7%	18.7%
Generic concepts	31.2%	42.8%
Years/identifiers	5.1%	8.2%
Abstract nouns	23.0%	30.3%

Table 5: Entity type distribution in top 100 entities.

KGGen's entity-first constraint biases extraction toward proper names—entities that are unambiguously recognizable. EDC's open architecture captures more abstract concepts and generic noun phrases. This reflects the architectural difference: constraining relations to pre-extracted entities favors named entity recognition patterns over conceptual vocabulary extraction.

4.3. Convergent Finding: Cultural vs. Textual Framing

We tested a hypothesis grounded in critical historiography of Argentine literary studies. Since its emergence in the 1950s, literary criticism in Argentina has been defined by an orientation toward "literature's outside"—following Blanchot's concept of literature's tension with culture (Blanchot, 1971), which Argentine criticism has historically operationalized as the practice of addressing literature through its social, cultural, and political dimensions (Cortés, 2021; Dalmaroni, 2004). The influential journal *Contorno* (1953–1959) inaugurated this orientation by confronting the aestheticism of *Sur* magazine and linking literature to historical events, influenced by Jean-Paul Sartre's theory of committed literature (Cella, 1999). This orientation persisted through subsequent decades: Dalmaroni (2004) traces a shift from "literature and society" in the 1950s to "culture and politics" in the 1980s and 1990s, shaped by the reception of British cultural studies in Argentina through journals like *Punto de vista* (1978–2008) (Sarlo, 2000; Richard, 2001).

Our hypothesis is that this orientation toward literature's outside—now articulated through cultural and political vocabulary—should be detectable in the knowledge graphs extracted from Orbis Tertius, from a time span of almost 30 years. We operationalized this by measuring edge weights between *literatura* 'literature' and two concept clusters:

Cultural concepts: *cultura* 'culture', *historia* 'history', *sociedad* 'society', *política* 'politics', *arte* 'art', *memoria* 'memory', *nación* 'nation', *mundo* 'world'

Textual concepts: *texto* 'text', *escritura* 'writing', *lectura* 'reading', *lenguaje* 'language', *palabra* 'word', *forma* 'form', *narración* 'narration'

Method	Cultural	Textual	Ratio
KGGen	101	40	2.52×
EDC	126	58	2.17×

Table 6: Cultural vs. textual framing weights.

Both methods show cultural framing dominates by approximately 2.2–2.5×. Chi-squared goodness-of-fit tests against the null hypothesis of equal weights confirm the imbalance is statistically significant for both KGGen ($\chi^2 = 26.4$, $p < .001$) and EDC ($\chi^2 = 25.1$, $p < .001$, $\text{df} = 1$). The totals are driven by a subset of each category: 4 of 8 cultural concepts (*cultura*, *historia*, *arte*, *sociedad*) and 4 of 7 textual concepts (*texto*, *escritura*, *lectura*, *lenguaje*) showed non-zero connections to *literatura*; the remaining concepts (*política*, *memoria*, *nación*, *mundo*, *palabra*, *forma*, *narración*) contributed zero edges in the KGGen curated graph. This convergence across fundamentally different extraction architectures strengthens confidence that the finding reflects textual and semantic patterns in the corpus. Argentine literary studies in Orbis Tertius frames literature primarily as a cultural phenomenon—embedded in social, historical, and political contexts—rather than as a purely textual object for formalist analysis.

To summarize, the quantitative pattern detected by both extraction methods aligns with the critical historiography outlined above: the orientation toward literature's outside that Dalmaroni (2004) traces from *Contorno* through *Punto de vista* persists in three decades of Orbis Tertius scholarship. The convergence across architecturally different methods—despite different entity constraints, predicate languages, and processing pipelines—reinforces our hypothesis that this cultural framing reflects a genuine discursive orientation in the corpus.

4.4. Convergent Finding: Author Networks

Both methods identify consistent author-to-author connections in the curated graphs. The scholarly consensus on key relationships emerges regardless of extraction architecture. Several connections show high edge weights in both methods: Borges–Groussac (reflecting historical succession in Argentine letters), Borges–Cortázar (the canonical pairing of mid-century fiction), Piglia–Arlt (Ricardo Piglia's critical recovery of Roberto Arlt as a major figure), and Darío–Groussac (the modernismo network connecting Rubén Darío to Argentine intellectuals). Contemporary connections like Aira–Piglia show medium weights in both methods, reflecting ongoing critical dialogue.

However, while the same author pairs emerge from both methods, their relative weights differ substantially. A Spearman rank correlation on the 12 pairs shared between both methods' top connections yields $\rho = 0.12$ ($p > .05$, $n = 12$), indicating weak rank concordance. For instance, Borges–Cortázar ranks among KGGen's strongest connections (weight 19) but appears with low weight in EDC (weight 3), while the Ocampo–Ocampo connection shows the reverse pattern (KGGen rank 9, EDC rank 2). This suggests that while both methods identify the same scholarly community structure, edge weights reflect extraction architecture as much as corpus signal.

The Borges-centrality finding is particularly robust: Borges appears as the highest-degree author node in both curated graphs, connected to authors across periods (Lugones, Groussac, Cortázar, Piglia) and to both cultural and textual concept clusters. This convergence across architecturally different extraction methods strengthens confidence

that the centrality of Borges in Argentine literary studies—a well-established claim in critical historiography—is indeed reflected in the discursive patterns of Orbis Tertius scholarship.

4.5. Divergent Finding: Predicate Characteristics

The methods diverge significantly in predicate extraction, as summarized in Table 7.

Aspect	KGGen	EDC
Predicate language	English	Spanish
Unique predicates (curated)	2,499	4,891
Analysis predicates	3.7%	5.9%
Semantic definitions	0	21,025

Table 7: Predicate characteristics comparison.

KGGen normalizes predicates to English regardless of source language, while EDC preserves Spanish predicates. EDC captures nearly twice as many unique predicates after curation, and shows higher proportion of scholarly analysis predicates—*analiza* 'analyzes', *estudia* 'studies', *critica* 'critiques', *interpreta* 'interprets'—suggesting it better captures the meta-discourse of literary studies rather than just the object discourse.

EDC's definition phase produces explicit meanings that enable semantic disambiguation, as illustrated in Table 8.

Predicate	EDC Definition
escribió	The subject produced or composed the object (text, work, etc.)
influyó	The subject exercised intellectual, aesthetic, or literary influence over the object
dialoga con	The subject establishes an intertextual or conceptual relationship with the object
reescribe	The subject transforms or reworks elements of the object in their own work

Table 8: Example semantic definitions generated by EDC (translated from Spanish).

Such definitions enable distinguishing authorship from analysis that KGGen's normalized predicates cannot capture. However, KGGen's normalized English predicates facilitate cross-corpus and cross-lingual comparison, while EDC's Spanish predicates preserve linguistic specificity important for humanities scholarship where terminology carries disciplinary weight.

4.6. Summary

Findings that converge across methods—the cultural hypothesis ratio and core author networks—can be reported with confidence as reflecting corpus patterns. Method-dependent findings—absolute counts, predicate language, entity composition—require acknowledging the extraction method when interpreting results.

5. Discussion

The comparative analysis reveals that extraction architecture shapes what gets captured in knowledge graphs from scholarly text. KGGen's entity-first constraint ensures relations are anchored on recognizable entities, producing cleaner graphs with higher proportion of proper names that map well to traditional NER categories. Its predicate normalization facilitates cross-corpus comparison, and its two-phase architecture is simpler and more predictable. EDC preserves semantic texture through Spanish predicates and explicit definitions—important for humanities scholarship where terminology carries disciplinary weight. Its open extraction architecture captures more abstract concepts and theoretical terms that NER-based approaches miss, and its higher proportion of analysis predicates indicates it captures not just what critics discuss but how they discuss it.

Our entity curation was anchored on KGGen's output: entities with degree ≥ 100 were selected from KGGen, and EDC's output was then fuzzy-matched to this set. This design prioritizes entities prominent enough to represent major disciplinary patterns verifiable by domain experts. A consequence is that EDC's unique entities—those that KGGen did not extract with sufficient frequency—are not represented in the comparison. However, all 20 of EDC's highest-degree entities appear among the 106 KGGen-curated entities, suggesting that both methods converge on which entities are most prominent. Of KGGen's 104,500 raw triplets, only 7,025 (6.7%) involved curated entities. Our comparative analysis therefore characterizes the densest core of the knowledge graph—the most prominent entities and connections—rather than the full extracted graph.

To test whether this curation choice drives the convergent findings, we performed a reverse analysis anchored on EDC's output, selecting EDC's 115 highest-degree entities (degree ≥ 100) independently of KGGen; full results are available in the accompanying repository. Of these, 82 have counterparts in KGGen's curated set via fuzzy string matching, while 33 are unique to EDC—predominantly generic literary terms (*artículo*, *protagonista*, *relato*, *revista*) and domain-specific concepts absent from the curated set (*parodia*, *surrealismo*,

exilio). In the EDC-anchored curated graph, author networks and Borges centrality are confirmed: the same author pairs emerge (Darío–Groussac, Ocampo–Borges, Aira–Piglia, Borges–Cortázar), and Borges remains the highest-degree author. The cultural framing analysis requires attention to the degree threshold. The threshold of 100, chosen for KGGen's degree distribution, is not equally calibrated to EDC's: cultural concepts like *cultura* (degree 98), *sociedad* (52), and *política* (60) fall just below it. At degree ≥ 50, which includes the full cultural vocabulary, the ratio is $2.71\times$, consistent with the KGGen-anchored results. The core convergent findings thus hold regardless of which method anchors the curation, though the appropriate degree threshold varies with extraction architecture.

The comparison suggests a methodological principle grounded in methodological triangulation (Heesen et al., 2019): findings that converge across architecturally different extraction methods merit higher confidence than findings from a single method, even when neither method can be independently validated. Our convergent findings held despite different entity constraints, predicate languages, and processing pipelines. This convergence does not validate individual triplets, but it provides stronger evidence that aggregate patterns exist in the corpus rather than being artifacts of a particular extraction architecture. Conversely, method-dependent findings should be reported with appropriate caveats. Researchers using knowledge graphs for literary analysis should avoid overinterpreting patterns that might reflect extraction architecture rather than corpus characteristics.

For hypothesis testing, we recommend using multiple extraction methods when possible; if a finding holds across methods, confidence increases. For author and entity networks, either method produces consistent core networks—choose based on downstream needs. For conceptual vocabulary, EDC's open architecture is more suitable when research questions involve how scholars use concepts rather than just which authors they discuss. For cross-lingual work, KGGen's English normalization facilitates comparison across languages. For Spanish-language scholarship, EDC's preservation of source-language predicates respects linguistic specificity.

Knowledge graph construction for humanities scholarship requires attention to what gets lost in extraction. Both methods produce useful graphs, but neither fully captures the interpretive texture of literary studies. The cultural framing finding illustrates what knowledge graphs can capture: aggregate patterns in how a scholarly community discusses literature over time. The predicate limitations illustrate what they cannot capture: the force of individual interpretations, the rhetorical moves of close reading. Knowledge graphs from humanities scholarship are best understood as maps of discourse topology—who discusses whom, what concepts co-occur—which constitutes one dimension of literary knowledge that combines with interpretive argumentation, rhetorical strategy, and historical situatedness to produce scholarly meaning.

6. Limitations

Knowledge graph extraction from humanities scholarship lacks established ground truth. Even for general open information extraction, "a clear and formal specification of what a valid relational tuple consists of is still missing" (Kamp et al., 2023); this challenge is amplified in interpretive domains like literary studies, where "correct" extraction is underdefined. We rely on convergence across methods as a form of methodological triangulation to identify robust aggregate patterns, though this does not constitute gold-standard evaluation and cannot validate individual extracted triplets. The field needs annotated corpora for humanities KG extraction—a resource that does not currently exist for literary studies in Spanish.

Knowledge graphs capture the topology of scholarly discourse but not the force of interpretation. Consider two articles that both link "Borges" to "gauchesca": one might argue Borges parodies the genre, another might argue he continues it. Both produce the same triplet, but the interpretive content differs fundamentally. This limitation is structural: knowledge graphs represent what is said but not the hermeneutic force of the saying.

Our specific comparison introduces a further limitation. KGGen (Gemini 2.0 Flash) and EDC (Mistral) use different underlying LLMs, so divergent findings—entity composition, predicate vocabulary—may reflect the underlying models as much as extraction architecture. Convergent findings are affected in the opposite direction: agreement despite different LLMs strengthens the evidence that a pattern originates in the corpus rather than in either model.

Both extraction methods rely on LLMs trained predominantly on English-language data. This may bias entity recognition toward figures and concepts prominent in anglophone scholarship, and could disadvantage domain-specific Spanish vocabulary that appears infrequently in training corpora. KGGen's normalization of predicates to English makes this bias explicit; EDC's preservation of Spanish predicates mitigates it at the predicate level but not at the entity level.

Our findings derive from a single journal from a specific institution. However, with more than 20 active peer-reviewed literary studies journals in Argentina, the methodology presented here is

readily scalable to broader comparative analyses across the field. We compare methods on Spanish text only; the comparison between KGGen's English normalization and EDC's Spanish preservation would differ for English-language corpora. Results depend on our degree threshold for entity curation, and both methods depend on specific LLM versions that may behave differently with updates.

7. Conclusion

We compared two LLM-based knowledge graph extraction approaches—entity-anchored (KGGen) and open extraction (EDC)—on 472 Spanish-language literary studies articles from Orbis Tertius (1996–2024). Despite fundamental architectural differences, both methods converge on key findings: cultural framing dominates literary discourse by 2.2–2.5$\times$ over textual framing, and core author networks with Borges as central figure are consistent across extraction approaches. This convergence strengthens confidence that these findings reflect the corpus rather than method artifacts, and demonstrates the utility of knowledge graphs for analyzing academic texts from well-defined disciplines such as literary studies, where language and meaning intertwine in complex ways.

The methods diverge in what they capture. KGGen's entity-first constraint produces graphs biased toward proper names (40.7% vs. 18.7%), with normalized English predicates suitable for cross-corpus comparison. EDC's open architecture captures more conceptual vocabulary (42.8% abstract concepts), preserves Spanish predicates, and generates semantic definitions that enable predicate disambiguation.

Based on our comparison, we suggest four practices for knowledge graph construction from scholarly text: test robustness by comparing extraction methods when feasible; select entity-anchored approaches for cross-corpus comparison and open extraction for capturing conceptual vocabulary; distinguish method-dependent from method-independent findings; and treat discourse topology as one dimension of scholarly knowledge rather than a complete representation of it. The cultural framing finding—that Argentine literary studies frames literature primarily through social and historical contexts—aligns with documented influence of cultural studies on Latin American scholarship and holds across methods, suggesting it reflects genuine discursive patterns in three decades of published literary criticism.

This study demonstrates the potential of natural language processing methods for literary studies, offering knowledge graphs as tools for exploration, hypothesis testing, and large-scale pattern discovery that complement traditional close reading approaches. An interactive visualization of the curated graph and the complete entity taxonomy are available in the accompanying repository.[1]

Acknowledgments

This work was supported by a Georg Forster Research Fellowship from the Alexander von Humboldt Foundation.

Ethical Considerations

This study analyzes publicly available open-access academic articles. No human subjects were involved. The corpus is freely accessible through the journal's OJS platform.

8. Bibliographical References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Fernanda Beigel and Maximiliano Salatino. 2015. Circuitos segmentados de consagración académica: las revistas de ciencias sociales y humanas en la argentina. *Información, cultura y sociedad*, (32).

Haonan Bian. 2025. LLM-empowered knowledge graph construction: A survey.

Maurice Blanchot. 1971. *L'amitié*. Gallimard.

Susana Cella. 1999. *Historia crítica de la literatura argentina vol. 10*. Emecé Editores.

Federico Gabriel Cortés. 2021. La palabra errante en maurice blanchot: Literatura como impugnación. *Landa*, (10(1)).

Miguel Dalmaroni. 2004. *La palabra justa: Literatura, crítica y memoria en la Argentina, 1960-2002*. Melusina.

Gemini Team. 2024. Gemini: A family of highly capable multimodal models.

Analía Gerbaudo. 2024. *Tanto con tan poco: los estudios literarios en Argentina 1958-2015*. Ediciones UNL.

[1]Data and code: https://github.com/fedexx1/orbis-kg

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

Remco Heesen, Liam Kofi Bright, and Andrew Zucker. 2019. Vindicating methodological triangulation. *Synthese*, 196(8):3067–3081.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mohamad Yaser Jaradeh, Kuldeep Singh, Markus Stocker, Andreas Both, and Sören Auer. 2023. Information extraction pipelines for knowledge graphs. *Knowledge and Information Systems*, 65(5):1989–2016.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Sreejith Sudhir Kalathil, Tian Li, Hang Dong, and Huizhi Liang. 2024. HTEKG: A human-trait-enhanced literary knowledge graph with language model evaluation. In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024)*, pages 207–215. SCITEPRESS.

Serafina Kamp, Morteza Fayazi, Zineb Benameur-El, Shuyan Yu, and Ronald Dreslinski. 2023. Open information extraction: A review of baseline techniques, approaches, and applications.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines.

Jiayin Lan, Jiaqi Li, Baoxin Wang, Ming Liu, Dayong Wu, Shijin Wang, and Bing Qin. 2025. NLP-AKG: Few-shot construction of NLP academic knowledge graph based on LLM.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Joan Cabezas, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. 2025. Kggen: Extracting knowledge graphs from plain text with language models.

Roxana Patiño. 1997. Intelectuales en transición: las revistas culturales argentinas (1981- 1987). *Cuadernos de Recienvenido*, (4).

Nelly Richard. 2001. Globalización académica, estudios culturales y crítica latinoamericana.

Beatriz Sarlo. 2000. Los estudios culturales y la crítica literaria en la encrucijada valorativa. *Revista de Crítica Cultural*, 21:32–38.

Marco Antonio Stranisci, Eleonora Bernasconi, Viviana Patti, Stefano Ferilli, Miguel Ceriani, and Rossana Damiano. 2023. The world literature knowledge graph. In *The Semantic Web – ISWC 2023*, volume 14266 of *Lecture Notes in Computer Science*, pages 435–452. Springer.

Carolina Unzurrunzaga, Cecilia Rozemblum, Cristian Pucacco, Gustavo Parente, and Mauricio Esterellas. 2015. Ojs implementation and development of the scientific journals site of the school of humanities and education sciences at the universidad nacional de la plata. *Scholarly and Research Communication*, (6(1)).

Shikhar Vashishth, Prince Jain, and Partha Talukdar. 2018. Cesi: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, page 1317–1327. ACM Press.

Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution.

Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2023. Generative knowledge graph construction: A review.

Bowen Zhang and Harold Soh. 2024. Extract, define, canonicalize: An llm-based framework for knowledge graph construction.

Demystifying Funding: Reconstructing a Unified Dataset of the UK Funding Lifecycle

William Thorne [1,2], **Rupert Shepherd** [1], **Diana Maynard** [2]
[1] National Gallery, [2] University of Sheffield
{wthorne1,d.maynard}@sheffield.ac.uk
rupert.shepherd@nationalgallery.org.uk

Abstract

We present a reconstruction of UKRI's Gateway to Research (GtR) database that links funding opportunities to their resulting project proposals through panel meeting outcomes. Unlike existing work that focuses primarily on funded projects and their outcomes, we close the complete funding lifecycle by integrating three previously disconnected data sources: the GtR project database, UKRI funding opportunities, and competitive funding decision records across UKRI's research councils. We describe the technical challenges of data collection, including navigating inconsistent publication formats and restricted access to panel decisions. The resulting dataset enables a holistic interrogation of the entire funding process, from opportunity announcement to research outcomes. We release the database and associated code.

Keywords: scientific funding, Gateway to Research, information extraction, data integration

1. Introduction

The UK Research and Innovation (UKRI) Gateway to Research (GtR) database provides the most complete public record of UK-funded research projects, encompassing project metadata, participant information, institutional affiliations, and research outcomes. While GtR has enabled valuable meta-research, it suffers from well-documented quality issues. Carlin and Rainer (2025) find that a quarter of reported software outputs have no links and 45% have missing or erroneous URLs. Liyanage et al. (2024) further note frequent duplicate data where "awards cross multiple years and/or were awarded to more than one institution." UKRI itself acknowledges "limitations on the accuracy, reliability and completeness of the data" (UK Research and Innovation, 2026), noting that classifications are inconsistently applied across funders, organisations lack unique identifiers across constituent systems, and historic information is not consistently comprehensive. A recent National Audit Office report found that data limitations are restricting UKRI's ability to manage its investments strategically (National Audit Office, 2025).

Beyond data quality, GtR suffers from issues of coverage as it maintains no public relationship between funded projects and the funding opportunities that solicited them. This omission is of particular concern as opportunities mark UKRI's first point of involvement in many projects' lifecycles and define the assessment criteria against which proposals are evaluated (Thorne et al., 2026). Compounding this, panel meeting outcomes that determine which proposals receive funding are published inconsistently across research councils, employing a mixture of PDFs and spreadsheets with variable structure and detail. Most concerning, two major councils publish their data exclusively through a third-party dashboard with data exports disabled, despite ostensibly operating under CC BY-NC-SA 4.0 licensing. This lack of transparency is notable given that bias in peer review and funding decisions has been well documented (Nesta, 2018), and that research into funding process fairness overwhelmingly focuses on the final stages precisely because the earlier stages remain opaque (Schneider et al., 2024).

We argue that by limiting accessibility to funding decision data, UKRI actively discourages scrutiny of awards at the point where final funding decisions are made (Liyanage et al., 2024). This work addresses this gap by capturing the full funding lifecycle, from opportunity publication through panel decision to research outcomes, integrating three previously disconnected data sources into a unified database that extends existing GtR data.[1]

2. Related Work

2.1. GtR as a Research Data Source

GtR serves as a primary data source for a range of meta-research applications. Vanino et al. (2019) used GtR to assess business performance effects of publicly-funded R&D grants, finding that firms participating in UKRI-funded projects experienced higher employment and turnover growth. GtR+ is a recent initiative that links GtR data with external sources such as OpenAlex and Companies House to support the analysis of topics including the im-

[1] Our code and data can be found here: https://github.com/wrmthorne/GtR-Extended

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 88–96

pact of rail connectivity on research collaboration and the regional distribution of research funding (Harris, 2025). However, both projects inherit GtR's lifecycle gap. Our work is complementary, providing the upstream link between opportunities, panel decisions, and awards to enable analysis across the full funding pipeline.

2.2. Data Quality and Reporting Concerns

Carlin and Rainer (2025) found that artifact sharing in GtR remains low, highlighting the dependence of GtR outcome data on the Researchfish platform, which until recently served as UKRI's primary source for outcomes data. UKRI's own documentation acknowledges that outcome reporting was not mandatory prior to December 2009, that classifications are inconsistently applied across research councils, and that the same organisation may appear under multiple variant names (UK Research and Innovation, 2026). Project regions are based on lead applicant postcodes, potentially misrepresenting where research occurs.

2.3. Funding Fairness and Transparency

Liyanage et al. (2024) identified systematic overrepresentation of Russell Group institutions in UKRI and significant under-funding of Post-92 universities, despite the latter serving larger and more socially diverse student populations; however, their analysis was necessarily limited to successful awards as GtR records only funded projects. Our dataset extends to include the most complete public information on unfunded, pending, and deferred proposals. The EDI-Caucus found that 44% of the literature focuses on gender bias, with racial inequity and institutional prestige receiving comparatively little attention, and that "little is known about the biases that affect which proposals reach review panels in the first place" (Schneider et al., 2024). Nesta (2018) similarly document how confirmation bias and tribalism in peer review can disadvantage novel or unconventional proposals. By providing structured access to panel-level decisions along side opportunities and project outcomes, our dataset enables the kind of cross-stage analysis that these studies identify as lacking.

2.4. Linking Funding Opportunities, Meetings and Outcomes

To our knowledge, no prior work has linked UKRI funding opportunities to their resulting proposals and panel decisions across research councils. Our work fills this gap, enabling research questions that span the full funding pipeline.

3. Data Sources

Our dataset integrates three primary sources: the UKRI Gateway to Research database, the complete collection of UKRI funding opportunities, and competitive funding decision records for each research council.

3.1. Gateway to Research

We extract data via all available GtR endpoints, including projects, funding records, persons, organisations, and outcome types (publications, collaborations, disseminations, policy influences, intellectual property, spin-outs, among others), ingesting into a PostgreSQL database schema centred on the *Project* entity. Outcomes are implemented using joined table inheritance with a generic base table and outcome-specific child tables. Given schema mismatches with GtR's data structures, ingestion proceeds with constraints disabled; we report dangling references in § 5. Where source IDs were available we reused them; otherwise we construct deterministic UUID5 identifiers.

3.2. Funding Opportunities

We retrieved all pending, open, and closed opportunities from the UKRI opportunity finder, constructing a structured table of core metadata: title, funders, publication/opening/closing dates, award values, and status. As status and closing date change over time, this represents a snapshot at the time of writing. Not all funded projects correspond to a published opportunity; research councils also operate responsive mode schemes and formula-based allocation mechanisms (UK Research and Innovation, 2024, 2026).

Raw HTML was parsed into a tree-structured representation: the opportunity summary forms the root, with accordion sections (e.g. "What we are looking for", "How to apply") as children, each storing an ordered sequence of paragraphs, lists, and subheadings as further subtrees. Updates are extracted into date-text pairs. This structure is serialised to a consistent markdown representation for downstream use with language models, and serves as the basis for the hierarchical retrieval described in § 4.

3.3. Panel Meeting Outcomes and Attendance

Collecting, and extracting outcomes and attendance from each of the councils was a significant undertaking given the inconsistent publication format and location, both by council and over time. Statistics for extracted data can be found in Table 1. ESRC is the most accessible, publishing to

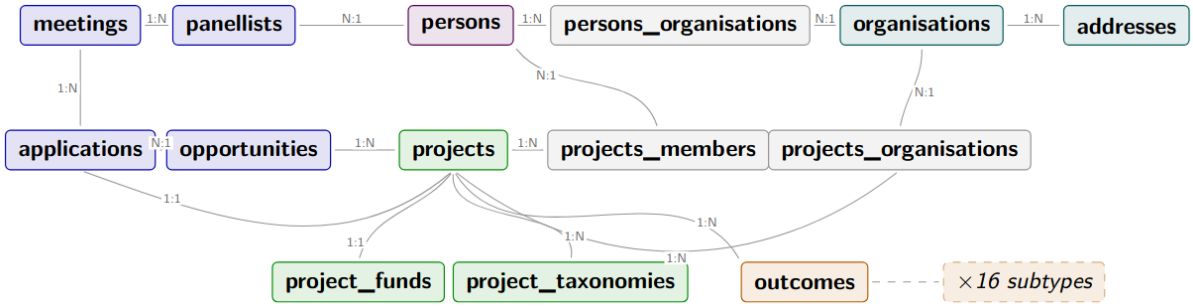


Figure 1: Simplified database schema. Tables are coloured according to primary entity: meeting/panel/opportunity (blue), project (green), people (purple), organisations (teal), outcomes (orange), link tables (grey).

Council	Meets.	Apps.	Panels	Recon.
AHRC	449	9,112	2,102	283
BBSRC	31	5,215	3,109	20
EPSRC	308	4,557	2,554	228
ESRC	158	4,992	395	10
MRC	324	7,522	339	16
NERC	153	7,966	3,231	32
STFC	27	1,145	0	0
Total	**1,450**	**40,509**	**11,730**	**589**

Table 1: Statistics of panel meeting and attendance extraction for each council. The table records the number of meetings, applications, successful outcomes, individual panel appearances (times a person was recorded present at a meeting), and meetings with successfully reconciled panel attendance.

a continuously updated spreadsheet on the UKRI website; however, panel attendance is not available for all meetings. Most other councils' data was sourced from a combination of the UKRI site and the Government Web Archive. Consistency and file format varies wildly; for example, pre-2025 MRC consistently relies on visual encoding in xslx files, using cell background colour to indicate funding status and column header numbers for scores, while NERC predominantly uses tables in PDFs that are devoid of any standardised layout or content. In most instances, attendance is recorded separately from application outcomes, if at all, and with insufficient information to reliably reconcile in many cases. Most problematically, EPSRC, post-2018 AHRC[2][3], and MRC from 2025 onward publish exclusively through Tableau, all with data exports disabled.

We argue this accessibility barrier goes against the spirit of UKRI's CC BY-NC-SA 4.0[4] licensing and is particularly problematic from the perspective of interrogating the funding process, given that these meetings mark the final determination of which projects receive funding.

4. Metadata Extraction from Opportunities

Much information about funding opportunities exists only in unstructured textual content. While some opportunities list award ranges in a summary table, the UKRI contribution percentage, project duration, and total fund value are frequently only stated within prose, distributed across document sections. We frame this as a closed-domain question-answering task: for each metadata field, retrieve the most relevant passages and extract a structured answer. Documents have a mean of $2,824$ whitespace-delimited tokens ($4,317$ excluding Innovate UK who typically publish short summaries of their full documents from the Innovation Funding Service), with the longest spanning $15,126$ words, motivating experimentation with retrieval-based context reduction (Paulsen, 2025).

4.1. Hierarchical Retrieval

We exploit the tree-structured document representation as a natural chunking mechanism. Each node yields one chunk comprising its header and all descendant content, such that a level-2 chunk subsumes all level-3 content within it. Chunks are scored against a query using a three-stage hybrid pipeline: a weighted combination of BM25 (Robertson and Zaragoza, 2009) and dense cosine similarity (via all-MiniLM-L6-v2 (Wang et al., 2020), normalised to $[0, 1]$):

$$S_{\text{hybrid}}(q, c) = \alpha \cdot S_{\text{BM25}}(q, c) + (1 - \alpha) \cdot S_\theta(q, c) \quad (1)$$

[2]Pre-2018 outcomes are available via the National Archives.

[3]Unavailable at the time of writing following our report to UKRI concerning unintentionally published applicant names, affiliations and project titles for unsuccessful proposals.

[4]https://www.ukri.org/who-we-are/terms-of-use/

Configuration			Overall	
Chunker	α	RR	Acc (%)	Unk (%)
Baseline	—		81.8	2.6
Simple (best)	1.00	✓	82.9	4.5
Hierarchical	0.00	✓	84.0	2.2
Hierarchical	0.25		84.4	3.7
Hierarchical	**0.25**	✓	**87.0**	**2.6**
Hierarchical	0.50	✓	83.6	2.6
Hierarchical	1.00		84.0	4.1

Table 2: Extraction accuracy for selected retrieval configurations. Baseline passes the full opportunity text. RR = re-ranker. The best simple chunker configuration and all hierarchical configurations exceeding the baseline are shown; full results are shown in Appendix D.

The top-k ($k = 10$) candidates are re-ranked by a cross-encoder (ms-marco-MiniLM-L-6-v2 (Wang et al., 2020)). We then apply *branch-deduplication*: for any ancestor–descendant pair among the candidates, the lower-scoring chunk is discarded. The top-k_{final} ($k_{\text{final}} = 5$) surviving chunks are concatenated as retrieval context.

4.2. Evaluation

We evaluate against manually annotated metadata from 101 opportunities, stratified first by funding council (Figure 3) and then by document length to ensure coverage across UKRI's heterogeneous publication styles. Up to six fields are annotated for each opportunity: minimum and maximum award value, total funding, UKRI contribution percentage, and minimum and maximum funding duration. This resulted in 489 question-answer pairs. We compare our hierarchical chunker against a sliding-window baseline (200-word chunks, 50-word overlap), vary $\alpha \in \{0, 0.25, 0.5, 0.75, 1.0\}$ to assess the contribution of each retrieval component, and ablate the re-ranker from each configuration.

Table 2 summarises extraction accuracy for selected configurations (full per-field results in Appendix D). The full-document baseline achieves 85.3% accuracy, demonstrating that the extraction task is tractable even without retrieval. However, the baseline benefits from seeing all relevant context simultaneously; its main disadvantage is cost and latency at scale.

The simple (sliding-window) chunker underperforms the baseline in all configurations, with accuracy dropping as low as 76.9% when the re-ranker is disabled at α=1 (pure BM25). Without the re-ranker, increasing α degrades accuracy monotonically (83.4% at α=0 to 76.9% at α=1), suggesting that BM25 alone produces poor chunk rankings for these documents, likely because keyword overlap between the question and irrelevant sections (e.g. eligibility criteria mentioning funding amounts in passing) misleads lexical retrieval. Re-ranking consistently recovers performance for simple chunking (e.g. from 83.4% to 82.8% at α=0, and from 76.9% to 82.2% at α=1), confirming that the initial retrieval set contains relevant chunks but ranks them poorly.

Hierarchical chunking outperforms both alternatives, with 8 of 10 configurations exceeding the baseline. We attribute this to chunking at section boundaries, which preserves the natural co-occurrence of related information. The best configuration (α=0.25, re-ranker disabled) achieves 87.7% overall accuracy, a 2.4 percentage point improvement over the full-document baseline. Notably, the re-ranker has a mixed effect under hierarchical chunking: it improves accuracy at α=0 (83.6% → 87.1%) but slightly harms it at α=0.25 (87.7% → 86.9%), suggesting that the hybrid scoring already produces a strong ranking when the dense and BM25 signals are appropriately balanced. Maximum award remains the hardest field across all configurations (best: 66.7%), which we attribute to frequent confusion with the closely related total funding field (see §4.3).

4.3. Error Analysis

To better understand model failure modes, we classify all 60 errors produced by the best-performing configuration (hierarchical, α=0.25, without re-ranker; 87.7% accuracy). Errors fall into three categories: *false positives* (47), where the model returns a value when the ground truth is null; *value mismatches* (9), where the model retrieved value does not align with the human annotation; and *false negatives* (4), where the model returns null despite an answer being present in the text.

False positives dominate, accounting for 78% of all errors. We further subdivide these into *field confusion* (17/47) and *hallucination* (30/47). Field confusion occurs when the model returns the correct value for a different field, for example, returning the total funding amount when asked for the maximum award, or the maximum duration when asked for the minimum. The most common confusion pairs are maximum→minimum duration (9 cases) and total funding→maximum award (4 cases). In the case of total funding, much of the confusion arises from the ambiguous phrasing of "[candidate] can apply for a share of up to [total funding]" in Innovate UK opportunities. While semantically allowable for a single applicant to be awarded the full total funding amount, this does not happen in reality.

The remaining 30 false positives are hallucinations in which the model produces a plausible-looking value that does not appear in the source document. Of these, 26 are entirely fabricated, 3

Entity	Count
Projects	173,220
People	141,833
Organisations	89,884
Outcomes	2,619,446
Meetings	1,449
Panel Appearances	6,951
Applications	38,862
Opportunities	2,053

Table 3: Number of captured entities, grouped by source.

can be traced to values present in the document's metadata table rather than the body text, and 1 appears elsewhere in the document body in an unrelated context.

Value mismatches (9 cases) typically involve rounding differences in monetary amounts or minor duration disagreements. False negatives are rare (4 cases), most often arising when the relevant information is spread across multiple non-adjacent paragraphs, suggesting that the retrieval context occasionally fails to capture all relevant passages.

5. Data Extraction and Alignment

Table 3 summarises the entities captured. All projects have titles, abstracts, and funding records, and 2.6 million outcomes are linked to projects with no orphaned references. We find 1,520 orphaned project member records and 10,128 orphaned project partner records, consistent with UKRI's acknowledgement that organisations lack unique identifiers across constituent systems (UK Research and Innovation, 2026). A further 22,519 projects have no associated team members, likely reflecting incomplete historical data.

5.1. Entity Linking

Because the three data sources share no common identifiers, we perform a series of linking steps to connect them into a unified dataset. All thresholds were chosen conservatively to favour precision over recall.

Application to Project. Applications are linked to GtR projects by exact case-insensitive match of the application's `application_id` or `award_id` against a project's `grant_reference`. This linked 10,531 of 38,862 applications (27.1%). Cross-validation on linked pairs with available metadata shows 97.2% organisation agreement (N=2,930), 95.7% PI surname agreement (N=3,074), and median title similarity of 1.000 (mean 0.996, N=3,048), yielding an estimated precision of 98.0% (95% CI: 97.7-98.3%). The remaining 72.9% of unlinked applications primarily reflect unfunded proposals with no corresponding GtR record.

Application to Opportunity. Applications are linked to opportunities via fuzzy title matching with multiple constraints. Candidate text is the `opportunity_name` from the application, falling back to the meeting name. Fuzzy ratio scores are computed against the opportunity title. Candidates are filtered by council-funder agreement and date ordering (earliest meeting on or after opportunity opening date). The score is boosted by 0.1 when the application route matches the opportunity funding type, and penalised by 0.15 when the total awarded amount falls outside the opportunity's award range. A minimum score threshold of 0.65 is required. This linked 11,514 applications (29.6%) to an opportunity.

Panel Attendance to Person. Panel attendance records are linked to GtR persons in two phases. First, records are disambiguated by clustering within each council on (surname, initial), splitting sub-clusters when full first names conflict or organisation similarity falls below 0.7. Second, each cluster is aligned to GtR persons with project membership records: unique surname-initial matches are linked directly; where multiple candidates exist, the best organisation similarity must exceed 0.6 with a margin of 0.15 over the second best. This linked 4,474 of 6,951 panel attendance records (64.4%) to a GtR person, with 100% surname containment and 71.3% organisation agreement on linked records.

6. Conclusion

We have presented a reconstruction of UKRI's Gateway to Research database that closes the funding lifecycle by linking opportunities, proposals, and panel decisions, enabling analysis at the point where final funding decisions are made and supporting future work on allocation fairness (Liyanage et al., 2024) and research policy evaluation. Our work also documents concerning barriers to access, particularly the restrictions on panel meeting data (National Audit Office, 2025) – data that enables investigation of panel composition and potential bias. By incorporating unfunded applications we provide a broader context around opportunity competitiveness and the full project lifecycle. We release our database and accompanying code to support further scrutiny of UK public research funding.

7. Ethical Considerations

All data used in this work is publicly available under CC BY-NC-SA 4.0. The dataset contains no personal data beyond what is already publicly disclosed by UKRI, and individuals are identified only through their institutional roles as principal investigators or panellists.

8. Limitations

Our dataset represents a snapshot at the time of collection; opportunity statuses and project details may have since updated. Meeting and panel alignment is incomplete as alignment was performed conservatively to avoid introducing incorrect data. Application-project linking covers 27.1% of applications, with the remainder primarily reflecting unfunded proposals with no GtR record. Application-opportunity linking covers 29.6% via fuzzy title matching with heuristic thresholds that may introduce errors. Finally, GtR's own data quality limitations, including orphaned records and inconsistent historical reporting, are inherited by our dataset.

9. Acknowledgements

This work was supported by the Arts and Humanities Research Council [grant number AH/X004201/1].

10. Bibliographical References

Domhnall Carlin and Austen Rainer. 2025. Tracking research software outputs in the UK. ArXiv:2507.22871 [cs].

Becky Harris. 2025. IRC GtR+ Secondary Data Analysis Funding Call - The Winning Projects.

Champika L. Liyanage, Felix Villalba-Romero, and Andrew Carmichael. 2024. Fairness in Higher Education Research and Innovation Funding in the UK. volume 3, pages 1031–1052. Multidisciplinary Digital Publishing Institute.

National Audit Office. 2025. UK Research and Innovation: Providing support through grants.

Nesta. 2018. Reducing bias in funding decisions. Innovation Squared Feature.

Norman Paulsen. 2025. Context Is What You Need: The Maximum Effective Context Window for Real World Limits of LLMs. ArXiv:2509.21361 [cs].

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Stefanie Schneider, Cat Morgan, Clayton Magill, Marion Hersh, Katherine Sang, and Robert MacIntosh. 2024. Evidence review: Peer review bias in the funding process: Main themes and interventions.

William Thorne, Joseph James, Yang Wang, Chenghua Lin, and Diana Maynard. 2026. Evaluating LLM-Based Grant Proposal Review via Structured Perturbations.

UK Research and Innovation. 2024. UKRI cross research council responsive mode pilot scheme.

UK Research and Innovation. 2026. A guide to gateway to research.

Enrico Vanino, Stephen Roper, and Bettina Becker. 2019. Knowledge to money: Assessing the business performance effects of publicly-funded R&D grants. *Research Policy*, 48(7):1714–1737.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

A. Meetings by Council Over Time

The distribution of meetings for each of UKRI's funding councils over time can be seen in Figure 2.

B. Metadata Extraction Prompt

The context of the prompt is populated dependent on the setting under evaluation i.e. baseline (the whole opportunity is included), simple (200 word chunks are inserted), or hierarchical (variable length context from the most relevant level of the document tree). Metadata fields and the associated questions to extract funding data can be found in Table 4.

```
system: You extract metadata from UKRI
    funding opportunity documents.

You will be given a document and a set
    of questions. Extract the answer
    strictly from the provided text. If
    the information is not explicitly
    stated, respond with null.

Rules:
- Only use information explicitly stated
     in the document.
```

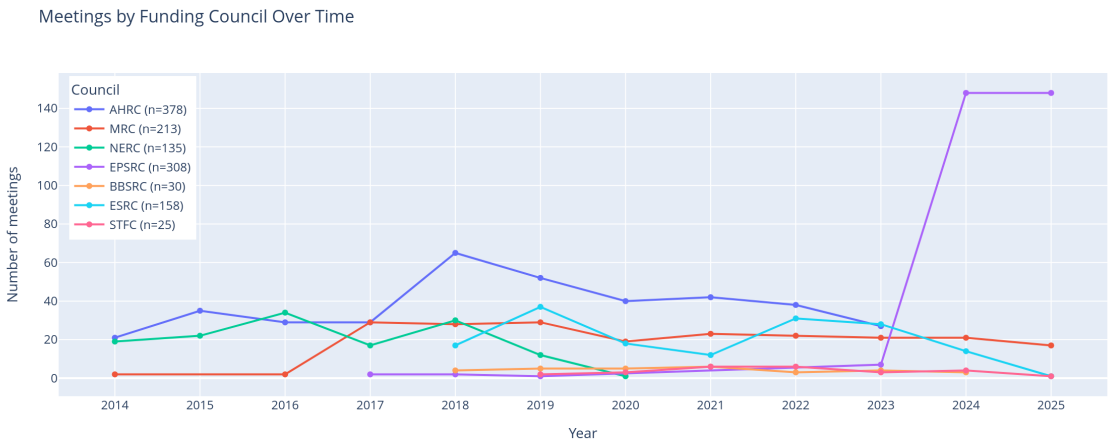


Figure 2: Number of meetings extracted per council over time.

Metadata Field	Question
minimum_award	What is the minimum fund value (£)?
maximum_award	What is the maximum fund value (£)?
total_funding	What is the total fund value (£) split among successful applications?
funding_percentage	What percentage (%) of the project's funding will UKRI fund?
minimum_funding_duration	What is the minimum duration of the project/funding?
maximum_funding_duration	What is the maximum duration of the project/funding?

Table 4: Fields and associated questions inserted into the metadata extraction prompt to retrieve specific funding information.

```
- Monetary values should be the nearest
    integer pounds without currency
    symbols or commas.
- Percentages should be the nearest
    integer without the % symbol.
- Durations should be quoted using the
    unit they appear with in the text.
    Duration strings may be rewritten to
     make sense (e.g. "36 months", "3
    years").

Respond with a JSON object only: {"<key
    >": "plain-text answer"} or {"<key>":
     null} if unknown. No other text.

user:
## CONTEXT
{context}

## QUESTION
{question}
```

C. Council Distribution of Test Data

The details regarding the council distribution in the extraction experiment test data can be seen in Figure 3.

D. Full Extraction Results

The complete table of results for the metadata extraction experiment are shown in Table 5.

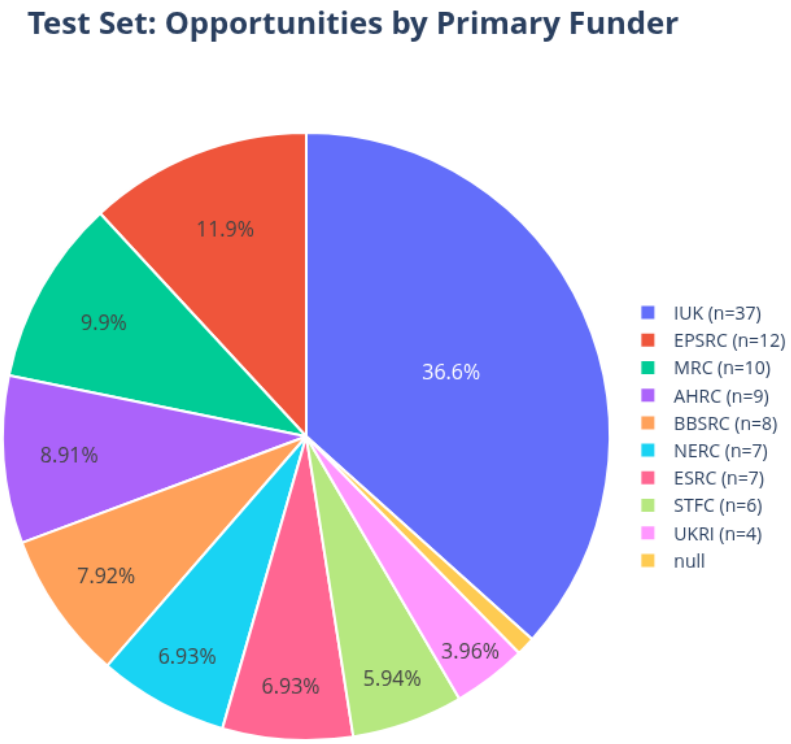


Figure 3: Breakdown of test samples by funding organisation. Samples were selected proportional to their total occurrence in the full dataset.

Configuration			Field Accuracy (%)						Overall	
Chunker	α	RR	Min	Max	Total	% Funded	Min Dur	Max Dur	Acc (%)	Unk (%)
None	—		83.5	59.7	72.4	98.0	87.1	94.1	85.3	53.4
Simple	0.00		**89.4**	59.7	79.3	94.1	83.2	86.1	83.4	58.3
Simple	0.00	✓	77.6	56.9	69.0	95.0	87.1	93.1	82.8	54.0
Simple	0.25		82.4	55.6	**82.8**	85.1	85.1	88.1	80.8	58.7
Simple	0.25	✓	76.5	59.7	79.3	96.0	86.1	93.1	83.6	53.4
Simple	0.50		**89.4**	61.1	75.9	78.2	85.1	88.1	81.0	63.2
Simple	0.50	✓	77.6	61.1	72.4	94.1	86.1	92.1	83.0	54.0
Simple	0.75		84.7	56.9	79.3	77.2	84.2	82.2	78.1	64.4
Simple	0.75	✓	74.1	61.1	79.3	93.1	86.1	90.1	82.2	53.2
Simple	1.00		82.4	55.6	**82.8**	78.2	83.2	78.2	76.9	65.4
Simple	1.00	✓	74.1	56.9	75.9	94.1	86.1	93.1	82.2	**53.0**
Hierarchical	0.00		87.1	58.3	69.0	94.1	88.1	88.1	83.6	57.3
Hierarchical	0.00	✓	**89.4**	65.3	79.3	97.0	88.1	92.1	87.1	58.1
Hierarchical	0.25		**89.4**	**66.7**	72.4	99.0	87.1	95.0	**87.7**	55.4
Hierarchical	0.25	✓	88.2	62.5	79.3	96.0	**89.1**	94.1	86.9	56.9
Hierarchical	0.50		87.1	65.3	65.5	98.0	88.1	95.0	86.7	55.2
Hierarchical	0.50	✓	88.2	63.9	72.4	96.0	88.1	92.1	86.1	57.5
Hierarchical	0.75		88.2	65.3	72.4	**100.0**	84.2	**96.0**	87.1	55.0
Hierarchical	0.75	✓	87.1	61.1	72.4	96.0	86.1	92.1	85.1	56.2
Hierarchical	1.00		85.9	62.5	69.0	98.0	**89.1**	93.1	86.1	56.6
Hierarchical	1.00	✓	84.7	63.9	75.9	97.0	**89.1**	92.1	86.1	56.9

Table 5: Extraction accuracy across retrieval configurations. RR = re-ranker enabled. Min/Max/Total = minimum, maximum, total award (£). % Funded = UKRI contribution percentage. Min/Max Dur = minimum, maximum funding duration. Configurations are grouped by chunking strategy; within each group, rows are ordered by α.

Do Lexical and Contextual Coreference Resolution Systems Degrade Differently under Mention Noise? An Empirical Study on Scientific Software Mentions

Atilla Kaan Alkan[1], **Felix Grezes**[1], **Jennifer Lynn Bartlett**[1], **Anna Kelbert**[1], **Kelly Lockhart**[1], **Alberto Accomazzi**[1]

[1]Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA
{atilla.alkan, felix.grezes, jennifer.bartlett, anna.kelbert, kelly.lockhart, alberto.accomazzi}@cfa.harvard.edu

Abstract

We present our participation in the SOMD 2026 shared task on cross-document software mention coreference resolution, where our systems ranked second across all three subtasks. We compare two fine-tuning-free approaches: Fuzzy Matching (FM), a lexical string-similarity method, and Context Aware Representations (CAR), which combines mention-level and document-level embeddings. Both achieve competitive performance across all subtasks (CoNLL F_1 of 0.94–0.96), with CAR consistently outperforming FM by 1 point on the official test set, consistent with the high surface regularity of software names, which reduces the need for complex semantic reasoning. A controlled noise-injection study reveals complementary failure modes: as boundary noise increases, CAR loses only 0.07 F_1 points from clean to fully corrupted input, compared to 0.20 for FM, whereas under mention substitution, FM degrades more gracefully (0.52 vs. 0.63). Our inference-time analysis shows that FM scales superlinearly with corpus size, whereas CAR scales approximately linearly, making CAR the more efficient choice at large scale. These findings suggest that system selection should be informed by both the noise profile of the upstream mention detector and the scale of the target corpus. We release our code to support future work on this underexplored task.

Keywords: cross-document coreference resolution, mention detection noise, scientific software mentions

1. Introduction

Coreference resolution is the task of identifying all mentions in a text, including proper nouns, definite descriptions, pronouns, and nominal expressions that refer to the same real-world entity (Jurafsky and Martin, 2026). It is a core component of natural language understanding pipelines, underpinning downstream tasks such as information extraction (Zelenko et al., 2004), question answering (Chai et al., 2022), and summarisation (Huang and Kurohashi, 2021). The automatic detection and disambiguation of software mentions in scientific literature has gained increasing attention as a means of improving research reproducibility and enabling large-scale meta-analyses of the use of methodologies across disciplines (Schindler et al., 2021, 2022; Du et al., 2021). In the scientific domain, and particularly for software mentions, coreference is predominantly nominal, with mentions taking the form of proper names, abbreviations, version-qualified strings, or URL references rather than pronouns or definite descriptions. This interest has been catalysed in part by shared evaluation campaigns: the SOMD 2025 (Upadhyaya et al., 2025) shared task, focused on detecting software mentions and their semantic relations in scientific text, attracted a range of systems submissions (Ojha et al., 2025; Rastogi and Tiwari, 2025; Mandic et al., 2025; Silva et al., 2025) and demonstrated both the feasibility and the remaining challenges of automated software mention extraction. Building on this foundation, SOMD 2026 extends the challenge to the cross-document setting, where the goal shifts from detecting individual mentions to resolving which mentions, across an entire collection of papers, refer to the same software entity. This is non-trivial: the same tool may appear under its full name, an acronym, a version-qualified string, or a URL, while conversely the same surface form may legitimately denote different tools in different disciplinary contexts. While coreference resolution has been studied extensively in both within-document and cross-document settings for scientific text (Chaimongkol et al., 2014; Luan et al., 2018; Brack et al., 2021; Forer and Hope, 2024), these efforts have focused primarily on general scientific concepts, biomedical entities, and argumentative discourse structures. Software mentions, with their particular mix of proper names, versioned identifiers, and abbreviations, constitute a distinct entity type that has received no dedicated coreference treatment. SOMD 2026 thus serves as the first standardised benchmark for this task.

In this paper, we present our participation in the SOMD 2026 shared task and address the following research questions:

- **RQ1** *Can a simple, unsupervised lexical base-*

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 97–107
12 May 2026.

line compete with a contextual embedding approach on cross-document software coreference? Given the high surface regularity of software names, we hypothesize that fuzzy string matching may constitute a strong baseline that is difficult to surpass without task-specific supervision.

- **RQ2** *How robust are these approaches to different types and levels of annotation noise?* Real-world annotations are imperfect, and understanding system behaviour under controlled degradation provides insight into practical deployment limits.
- **RQ3** *What is the precision–speed trade-off between the two approaches?* For large-scale literature mining pipelines, inference efficiency is a major concern alongside accuracy.

Our main contributions are: (i) two competitive unsupervised baselines for the SOMD 2026 shared task; (ii) as part of this work, a systematic noise injection study that, to our knowledge, is the first robustness analysis conducted in the context of software mention coreference resolution; and (iii) a characterisation of the inference-time trade-off between lexical and neural approaches on this task. We release our code[1] to support reproducibility and future work on this task.

The remainder of this paper proceeds as follows. Section 2 reviews related work and positions software mention coreference as a distinct problem. Section 3 describes the shared task, corpus, and training data analysis that informed our design choices. Section 4 presents our two systems. Section 5 describes the evaluation metrics and noise injection protocol. Section 6 reports results and robustness findings. Section 7 discusses broader implications and limitations. Section 8 concludes with a summary of key insights and directions for future research.

2. Related Work

Software Mention Detection and Disambiguation The extraction of software mentions from scientific text has received growing attention, with corpora such as SoftCite (Du et al., 2021) and SoMeSci (Schindler et al., 2021) providing annotated mentions of software names, version numbers, URLs, and developer attributes. These efforts primarily focus on named entity recognition and attribute extraction rather than on coreference. Shared evaluation campaigns have further advanced the field: Grezes et al. (2022) organized the DEAL 2023 shared task on entity detection in astrophysics literature, which included software mentions as a target entity type. More recently, SOMD 2025 (Upadhyaya et al., 2025) targeted the detection of software mentions and their relational attributes as named entities in scholarly text, attracting a range of submissions exploring detection and relation extraction (Ojha et al., 2025; Rastogi and Tiwari, 2025; Mandic et al., 2025; Silva et al., 2025). Across these efforts, the focus has consistently remained on mention-level extraction; cross-document disambiguation and coreference resolution have received less dedicated treatment, a gap that SOMD 2026, to the best of our knowledge, is the first shared task to directly address.

Scientific Coreference Resolution Coreference resolution has been extensively studied in newswire and general-domain text (Lee et al., 2017; Joshi et al., 2020), but scientific text presents distinct challenges: dense technical terminology, heavy use of abbreviations, and domain-specific entity types are poorly covered by general-purpose systems. Within-document coreference for scientific text has been widely addressed in the biomedical domain (Zweigenbaum et al., 2012; Lu and Poesio, 2021), supported by annotated corpora such as MedStract (Li et al., 2014), Genia-MedCo (Li et al., 2014), and DrugNerAR (Segura-Bedmar et al., 2009). Coreference corpora have also been developed for broader scientific domains (Chaimongkol et al., 2014; Brack et al., 2021; Luan et al., 2018) and for astrophysics (Alkan et al., 2024). Cross-document scientific coreference has received comparatively less attention: Cattan et al. (2021b) introduced SciCo for cross-document coreference of scientific concepts, while recent work has explored LLM-based relational reasoning (Forer and Hope, 2024) and knowledge-graph-grounded entity linking (Dong et al., 2025) to improve cross-document resolution. Despite this growing body of work, software mentions, as a distinct entity type combining versioned identifiers and abbreviations, remain understudied.

Coreference Resolution Methods Early coreference research relied on unsupervised heuristics and rule-based approaches, drawing on linguistic constraints such as syntactic and semantic compatibility (Hobbs, 1978; Grosz and Sidner, 1986; Grosz et al., 1995; Haghighi and Klein, 2010), with recent work demonstrating that simple unsupervised rules remain competitive in certain settings (Stolfo et al., 2022). The field has since shifted toward supervised neural architectures for both within-document and cross-document resolution (Clark and Manning, 2016; Wiseman et al., 2015; Tourille et al., 2020; Gliosca and Amsili, 2019; Barhom et al., 2019; Cattan et al., 2021a), and more recently started to explore zero-shot approaches leveraging

[1] https://github.com/adsabs/SOMD-2026

pre-trained language models such as BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019), and LLMs with prompting strategies (Blevins et al., 2023; Le et al., 2022; Le and Ritter, 2023). However, these methods are designed for mention types that differ substantially from those in the SOMD 2026 shared task. Existing systems typically target pronominal anaphora and nominal expressions with high lexical diversity, whereas the annotation scheme adopted by the SOMD 2026 organisers focuses exclusively on explicit software name mentions (a mention type characterised by a high degree of surface form similarity between corefering expressions). This distinction matters: the linguistic variation that motivates complex neural architectures is largely absent here, making heavyweight models both unnecessary and costly. While lighter alternatives such as FastCoref (Otmazgin et al., 2022) reduce computational overhead, they remain expensive for large-scale literature mining pipelines and are designed for general coreference rather than this specific mention type. The high surface regularity of software names and the need for scalable processing together motivate our choice of two lightweight, unsupervised approaches: fuzzy string matching, which directly exploits lexical similarity, and clustering over contextual embeddings, which captures semantic variation without fine-tuning.

3. Task and Corpus

3.1. Task Definition

SOMD 2026 frames software mention disambiguation as a cross-document coreference resolution problem: given a set of software mention spans with their surrounding sentences and metadata, the goal is to partition all mentions into clusters such that each cluster corresponds to a single underlying software entity. The three subtasks differ in the quality of the input mentions and the scale of the corpus:

- **Subtask 1** operates over gold-standard annotated mentions, providing an upper-bound evaluation of coreference resolution in isolation from mention detection errors;
- **Subtask 2** operates over automatically predicted mentions, reflecting real-world conditions where upstream mention detection is imperfect and introduces noise into the coreference input;
- **Subtask 3** operates over predicted mentions at a larger scale, explicitly targeting the computational efficiency challenge that arises as the volume of documents and the density of software name variants increase.

Our participation covers all three subtasks. Subtasks 2 and 3 operate on automatically predicted mentions, so the noise level in the input is inherent to the upstream mention detection pipeline and varies in ways that are difficult to quantify directly. To complement the official evaluation and gain a more controlled understanding of how input quality affects each system, we conduct a noise-injection study on the gold-standard training data, systematically varying the noise level across two perturbation types (**RQ2**). The scale dimension of Subtask 3 similarly motivates our inference-time analysis (**RQ3**).

3.2. Corpus Statistics

The shared task datasets comprise scholarly documents from scientific disciplines, annotated with software mention spans and their coreference chains. Two distinct training sets are provided: Subtask 1 uses gold-standard annotations, while Subtasks 2 and 3 share a training set of automatically predicted mentions. Table 1 reports corpus statistics for both.

Statistic	**Subtask 1** *(gold)*	**Subtasks 2 & 3** *(predicted)*
Corpus		
Documents	973	967
Sentences with mentions	2,153	2,140
Mention instances	2,974	2,860
Unique surface forms	837	791
Coreference chains		
Total clusters	733	699
Avg chain length	4.06	4.09
Max chain length	367	366
Singleton rate	51.7%	52.5%
Cross-doc rate (all clusters)	20.6%	20.9%
Cross-doc rate (non-singletons)	42.7%	44.0%
Mention surface forms		
Avg surface forms / cluster	1.14	1.13
Avg intra-cluster lexical sim.	0.881	0.887

Table 1: Corpus statistics for the SOMD 2026 training sets. Subtask 1 uses gold-standard mentions; Subtasks 2 and 3 share a common training set of automatically predicted mentions. Cross-doc rate (non-singletons) excludes singleton clusters, which require no linking decision.

The two training sets are relatively similar across all statistics, suggesting that the automatic mention detector used for Subtasks 2 and 3 is of high quality: it recovers a comparable number of mentions (2,860 vs. 2,974), a similar coreference chain structure, and a nearly identical lexical similarity profile. The primary difference lies in the slightly lower mention count and number of unique surface forms, reflecting mentions that the detector failed to recover. This observation is consequential for our experimental design: since the real-world

noise introduced by the upstream detector in Subtasks 2 and 3 is inherently mild and unquantified, we complement the official evaluation with a controlled noise injection study that systematically explores a wider range of noise levels, allowing us to characterise how each system degrades as input quality decreases (**RQ2**).

The coreference structure of both training sets reveals several properties that are consequential for system design. The clusters exhibit a highly skewed length distribution, with maximum chain lengths of 367 and 366 and average lengths of 4.06 and 4.09 for Subtasks 1 and 2&3, respectively. This is consistent with a small set of high-frequency software names, such as MATLAB, dominating the corpus, whereas most tools appear infrequently. Notably, singleton rates of 51.7% and 52.5% indicate that over half of all mentions lack a coreferent counterpart, implying that any coreference system must be conservative in its linking decisions.

An analysis of the coreference chain structure reveals an important nuance regarding the cross-document nature of the task. The raw cross-document cluster rate is approximately 20% in both training sets, suggesting that the resolution problem is predominantly within-document. However, this figure is strongly influenced by the high singleton rate of around 52%: singletons trivially belong to a single document and require no linking decision. Among non-singleton chains, the cross-document rate rises to 42.7% and 44.0% for Subtasks 1 and 2&3, respectively, confirming that the task poses a genuine cross-document disambiguation challenge for the majority of chains that actually require coreference linking.

Most consequentially for our system design choices, both training sets exhibit a high degree of lexical regularity within coreference chains. The average number of distinct surface forms per cluster is 1.14 and 1.13, respectively, indicating that coreferring mentions are almost always near-identical strings rather than paraphrases or pronominal references. This is confirmed by average intra-cluster lexical similarities of 0.881 and 0.887, substantially higher than what would be expected in general coreference corpora where chains mix proper names, nominal descriptions, and pronouns. This property directly motivates our choice of lightweight unsupervised approaches and supports the hypothesis underlying **RQ1** that lexical similarity alone may constitute a strong signal for this task.

4. Systems

4.1. Fuzzy Matching

The fuzzy matching system clusters software mentions based on the lexical surface similarity. For each pair of mention strings m_i and m_j, we compute a similarity score $s(m_i, m_j) \in [0, 1]$ using the Ratcliff/Obershelp algorithm (Ratcliff and Obershelp, 1988), as implemented by SequenceMatcher in Python's difflib library. The Ratcliff/Obershelp algorithm computes similarity as twice the number of matching characters divided by the total number of characters in both strings, where matching characters are identified by recursively finding the longest common substring and applying the same procedure to the non-matching regions on either side. This makes it sensitive to the overall structure of the string rather than character-level edit distance, and well-suited to software names where shared substrings are a strong indicator of coreference (e.g., *"GraphPad Prism"* and *"GraphPad Prism 8"*). Two mentions are linked if $s(m_i, m_j) \geq \theta$, where the threshold θ is tuned on the training set. Clusters are then formed by applying transitive closure over all linked mention pairs, such that if m_i is linked to m_j and m_j is linked to m_k, all three are assigned to the same cluster regardless of the direct similarity between m_i and m_k. The fuzzy matching system operates solely on mention strings and is entirely agnostic to document context, mention type, and surrounding text. Its computational complexity is superlinear in the number of unique mention strings, which is manageable given the relatively small number of unique surface forms in the corpus (837 and 791 for Subtasks 1 and 2&3, respectively; see Table 1).

4.2. Context Aware Representations

The context-aware representations system encodes each software mention using all-MiniLM-L6-v2 (Wang et al., 2020), a lightweight sentence embedding model from the Sentence Transformers library. The model comprises 6 transformer layers and 22M parameters and has been trained to produce semantically meaningful sentence-level representations via knowledge distillation from larger models. Its compact size makes it well-suited to large-scale mention encoding without requiring GPU acceleration. Rather than encoding the mention in its sentential context alone, we separately encode two complementary signals and combine them into a single representation. First, the normalised mention string is encoded independently, producing a mention-level representation $\mathbf{e}_m \in \mathbb{R}^d$ that captures the surface form of the software name. Second, a document-level representation $\mathbf{e}_d \in \mathbb{R}^d$ is constructed by aggregating up to ten unique mention-bearing sentences from the same document into a single string, which is then encoded with all-MiniLM-L6-v2. This document context captures the broader thematic and disciplinary setting in which the mention appears, providing a complementary signal for cases where the same surface

form refers to different software in different contexts. Both representations are independently normalised to unit length and combined as a weighted sum:

$$\mathbf{e} = \alpha \cdot \mathbf{e}_m + (1 - \alpha) \cdot \mathbf{e}_d \quad (1)$$

where $\alpha = 0.6$, giving slightly more weight to the mention-level signal. This design reflects the intuition confirmed by our corpus analysis (Section 3.2): software names are highly surface-regular, making the mention string the primary coreference signal, while document context provides a disambiguating signal for ambiguous cases. The combined representations are clustered using agglomerative clustering with cosine distance and average linkage. Since each mention is encoded independently by the sentence transformer without reference to other mentions, the encoding step scales linearly with corpus size; the subsequent agglomerative clustering step runs in $O(n^2 \log n)$ via SKLEARN's precomputed distance matrix approach, avoiding the naive $O(n^3)$ complexity of a stored-matrix implementation.

4.3. Threshold Tuning

The fuzzy matching system requires a single threshold hyperparameter θ, defining the minimum Ratcliff/Obershelp similarity score above which two mentions are linked. We perform a grid search over $\theta \in [0, 1]$ on the training set, selecting the value that maximises CoNLL F_1. Since no development set is provided by the shared task, we use the full training set for this purpose. The selected threshold is $\theta = 0.83$ for Subtask 1 and $\theta = 0.84$ for Subtasks 2 and 3. The context-aware system uses a fixed distance threshold of $\delta = 0.4$ for the agglomerative clustering step, which was set empirically without formal tuning. For the noise-injection experiments, the threshold θ is re-tuned at each noise level using the same grid-search procedure, and the best achievable performance is reported for each noise condition. This provides an optimistic upper bound on the robustness of the fuzzy matching method, assuming that the system has access to a clean validation signal at each noise level. The implications of this choice are discussed further in Section 7.

5. Experimental Setup

5.1. Evaluation Metrics

All official test-set scores are computed by the shared task organisers using the standard coreference resolution metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAFe (Luo, 2005). The primary metric is CoNLL F_1 (Pradhan et al., 2014), defined as the unweighted average of the three F_1 scores. In addition to coreference performance, we report the average inference time for each system in order to characterise the precision–speed trade-off between the two approaches (**RQ3**).

5.2. Noise Injection Protocol

As discussed in Section 3.1, the noise level introduced by the automatic mention detector in Subtasks 2 and 3 is inherent to the upstream pipeline and cannot be directly quantified. To complement the official evaluation, we conduct a controlled internal robustness analysis by injecting noise directly into the training set mentions and evaluating both systems on the resulting perturbed data. This diagnostic study is independent of the official test set and is not intended to be directly compared with the results in Table 3; its purpose is to assess how each system degrades as input quality decreases under controlled, quantifiable noise conditions (**RQ2**). The two noise types we consider are motivated by realistic failure modes of mention detection systems. **Boundary modification** simulates span boundary errors, which are among the most common annotation and detection mistakes in span-level tasks: a mention span is randomly extended or truncated, producing a slightly incorrect but plausible mention. **Mention substitution** simulates a context mismatch error, where a mention detection system correctly identifies a span as a software mention but associates it with the wrong software name: the mention string is replaced with a different software name sampled from the training set, preserving the syntactic structure of the sentence. Table 2 illustrates each noise type on a concrete example. Each perturbation type is applied at different intensity levels: 0%, 25%, 50%, 75%, and 100% of all mentions affected, and both systems are evaluated after each perturbation.

6. Results

6.1. Main Results on the Test Set

Table 3 reports the official test-set results for all participating SOMD 2026 systems alongside our two submissions.

The results reveal several findings. First, all systems, including our two unsupervised, fine-tuning-free baselines, achieve very high CoNLL F_1 scores across all subtasks, ranging from 0.94 to 0.96. This confirms the hypothesis underlying **RQ1**: the high lexical regularity of software mention chains (avg. intra-cluster similarity of 0.881; see Table 1) implies that even a simple surface-matching approach constitutes a strong baseline for this task. Second, our context-aware representations system consistently outperforms the fuzzy matching approach,

Noise Type	Example
Original	We used MATLAB for statistical analysis and visualization.
Boundary Modification	We used MATLAB for statistical analysis and visualization. We used the MATLAB for statistical analysis and visualization.
Mention Substitution	We used Python for statistical analysis and visualization. We used NumPy for statistical analysis and visualization.

Table 2: Illustration of noise injection methods applied to a software mention. The original mention is shown in blue, while noisy mentions are shown in red. Boundary modification extends or truncates mention spans; mention substitution replaces the mention string with another software name sampled from the training set. Each noise type simulates a different failure mode of upstream mention detection. All methods are tested at noise rates of 0%, 10%, 25%, 50%, 75%, and 100%.

	Subtask 1				Subtask 2				Subtask 3			
System	CoNLL	B^3	$CEAF_e$	MUC	CoNLL	B^3	$CEAF_e$	MUC	CoNLL	B^3	$CEAF_e$	MUC
System A†	**0.98**	**0.99**	**0.96**	**0.99**	**0.98**	**0.99**	**0.95**	**0.99**	**0.96**	**0.97**	**0.92**	**0.99**
System B†	0.92	0.94	0.86	0.97	0.92	0.93	0.85	0.98	‡	‡	‡	‡
Fuzzy Matching	0.95	0.95	0.91	0.98	0.95	0.94	0.91	**0.99**	0.94	0.94	0.90	**0.99**
Context Aware	0.96	0.96	0.93	0.98	0.96	0.96	0.93	**0.99**	‡	‡	‡	‡

Table 3: Official test-set results for SOMD 2026. CoNLL is the average of MUC, B^3, and CEAFe F_1. **Bold** = best overall. Underline = best among our systems. † = other participants. ‡ = no submission on Codabench.

achieving a CoNLL F_1 of 0.96 on both Subtasks 1 and 2 compared to 0.95 for FM. While the absolute gap is modest (one point), it is consistent across all coreference metrics. The improvement is most visible on $CEAF_e$ (0.93 vs. 0.91), which penalises both over- and under-clustering and is therefore a more sensitive indicator of clustering quality than MUC, which is known to favour recall. This suggests that the document-level contextual representations in CAR provide a small but consistent benefit over purely lexical matching, likely by resolving ambiguous cases in which the same surface form can refer to different software across different disciplinary contexts. Third, both our systems outperform System B† on all subtasks by a margin of 3–4 CoNLL F_1 points, despite requiring no training or fine-tuning. This further underscores the strength of unsupervised lexical and lightweight embedding approaches for this task, and suggests that the added complexity of supervised systems may not yet be warranted given the current scale and lexical regularity of the corpus. Fourth, the performance of all systems is remarkably stable across Subtasks 1, 2, and 3. The transition from gold-standard mentions (Subtask 1) to automatically predicted mentions (Subtask 2) produces no measurable degradation for any system, which is consistent with our corpus analysis showing that the two training sets are nearly identical in their statistical properties (Table 1). This suggests that the upstream mention detector used by the shared task organisers is of high quality.

6.2. Robustness to Mention Noise

We evaluate robustness by injecting controlled noise into the training sets of Subtasks 1 and 2. Table 4 reports degradation curves for both noise types. As noted in Section 4.3, FM threshold θ is retuned at each noise level, so FM scores represent an optimistic upper bound.

Boundary Modification FM loses 0.20 CoNLL F_1 points over the full noise range (0.70 $\rightarrow$ 0.50) while CAR loses only 0.07 (0.82 $\rightarrow$ 0.75), a three-fold robustness advantage. FM degrades approximately linearly, while CAR's curve is nearly flat. This asymmetry reflects how each system processes mention strings: FM relies on exact character-level matching, so adding or removing one token directly degrades similarity scores. CAR encodes mentions as dense vectors, so minor boundary changes leave the semantic content largely intact (e.g., *"MATLAB for"* encodes similarly to *"MATLAB"*). This robustness is a property of the embedding representation itself, not evidence that document context provides a compensatory signal, as the mention substitution results confirm. Practically, CAR at 100% boundary noise (0.75) still outperforms FM at 0%

Noise Type	System	Injected Noise Rate ∅	25%	50%	75%	100%	Δ
Subtask 1 (gold mentions)							
Boundary Modification	Fuzzy Matching	0.70	0.65	0.60	0.54	0.50	0.20
	Context Aware	0.82	0.80	0.79	0.77	0.75	**0.07**
Mention Substitution	Fuzzy Matching	0.70	0.49	0.34	0.23	0.18	**0.52**
	Context Aware	0.82	0.57	0.39	0.25	0.19	0.63
Subtask 2 (predicted mentions)							
Boundary Modification	Fuzzy Matching	0.70	0.66	0.62	0.57	0.51	0.19
	Context Aware	0.82	0.80	0.80	0.78	0.76	**0.06**
Mention Substitution	Fuzzy Matching	0.70	0.50	0.34	0.23	0.18	**0.52**
	Context Aware	0.82	0.59	0.39	0.26	0.20	0.62

Table 4: CoNLL F_1 under all noise types at increasing noise rates on the training set. Δ = F_1 at 0% − F_1 at 100% (lower = more robust). Baseline scores differ from official test-set results (Table 3) as noise experiments are conducted on the training set. Underline = best F_1; **bold** Δ = most robust system per noise type.

noise (0.70).

Mention Substitution The robustness ranking reverses: FM ($\Delta = 0.52$) is now more robust than CAR ($\Delta = 0.63$), despite starting from a lower clean baseline. Both systems converge to near-identical performance at 100% noise (FM: 0.18, CAR: 0.19). FM degrades approximately linearly, whereas CAR collapses more sharply at low noise levels: at 25% noise, CAR drops by 0.25 points, whereas FM drops by 0.21. This early collapse occurs because CAR's document context is constructed from mention-bearing sentences (Section 4.2): substituting mention strings corrupts both the mention-level and document-level embeddings simultaneously, so neither component can compensate for the other. This resolves the apparent tension with the boundary modification results: CAR's robustness advantage there reflects the tolerance of dense embeddings to minor perturbations, not an independent contextual signal. When mention content is substantively replaced, this advantage disappears.

Consistency and Summary Degradation patterns are consistent across Subtasks 1 and 2 (with differences of at most 0.01 across all conditions), confirming that the findings generalise to the predicted-mention setting. Overall, the two systems exhibit complementary robustness profiles: CAR is more robust to boundary noise while FM degrades more gracefully under mention substitution, and neither system is resilient to severe mention content corruption (**RQ2**).

6.3. Inference Time and Precision–Speed Trade-off

We report in Table 5 the inference time and precision–speed trade-off for both systems on the official test sets of Subtasks 1 and 2, measured on a CPU-only machine (Intel x86_64, 14 physical cores, 33.3 GB RAM) over 5 runs.

Subtask	System	Time (s)	CoNLL F_1	F1/sec
Subtask 1	FM	0.60 ± 0.00	0.95	1.584
	CAR	4.45 ± 0.69	0.96	0.215
Subtask 2	FM	47.06 ± 0.41	0.95	0.020
	CAR	45.39 ± 5.14	0.96	0.021

Table 5: Inference time and precision–speed trade-off on official test sets of Subtasks 1 and 2. Efficiency = CoNLL F_1 / inference time (s). Measured on an Intel x86_64 CPU (14 cores, 33.3 GB RAM) over 5 runs.

At small scale (Subtask 1, n = 743 mentions), FM is 7.4× faster than CAR (0.60s vs. 4.45s) at near-identical accuracy (0.95 vs. 0.96), making it the more efficient choice for small corpora. At large scale (Subtask 2, $n \approx$ 21,500 mentions), the picture reverses: the input size increases 29× relative to Subtask 1, but FM's inference time increases 78× (0.60s → 47.06s) while CAR's increases only 10× (4.45s → 45.39s), and both systems converge to nearly identical inference times (≈46s) with CAR still holding its marginal accuracy advantage. This result is counter-intuitive: despite being the simpler and lighter system, FM scales less favourably than CAR at a large scale. The reason is that FM's simplicity relies on pairwise string comparison, whose

cost grows faster than linearly with the number of mentions, whereas CAR's neural encoding (though more expensive per mention) processes each mention independently and therefore scales approximately linearly. FM's inference time is perfectly stable across runs ($\pm$0.00s), reflecting its deterministic procedure, whereas CAR exhibits higher variance ($\pm$0.69s on Subtask 1, $\pm$5.14s on Subtask 2), attributable to variability in the neural encoding step on the CPU; GPU acceleration would reduce both.

7. Discussion

Accuracy vs. scale (RQ3) At small scale (Subtask 1, $n = 743$), FM is 7.4$\times$ faster than CAR at near-identical accuracy, making it the more practical choice. At large scale (Subtask 2, $n \approx 21{,}500$), this advantage disappears: FM's inference time increases 78$\times$ while CAR's increases only 10$\times$, with both systems converging to similar inference times ($\approx$46s). This is consistent with FM's superlinear scaling in pairwise similarity computation, compared with CAR's approximately linear per-mention encoding. The practical recommendation is scale-dependent: FM for small corpora, CAR for large-scale pipelines.

Robustness, lexical regularity, and system selection (RQ1 & RQ2) A unifying finding across RQ1 and RQ2 is that the high lexical regularity of software mention chains, confirmed by an average intra-cluster similarity of 0.881 (Table 1), is the dominant factor shaping system behaviour. It explains why FM is competitive with CAR on clean data, why CAR's robustness advantage under boundary noise reflects embedding tolerance rather than genuine contextual signal, and why both systems collapse equally under mention substitution. Above a certain noise threshold, improving the upstream detector is more impactful than the choice of coreference method. Future work would benefit from document-level representations genuinely decoupled from mention string content. For instance, encoding non-mention sentences or document signals such as titles and abstracts.

Limitations Three limitations should be noted. First, FM's threshold θ was re-tuned at each noise level, so robustness results represent an optimistic upper bound. Second, CAR's document context is constructed from mention-bearing sentences, making it structurally dependent on mention string content. Third, inference times were measured on CPU only; GPU acceleration would reduce CAR's inference time and potentially widen its efficiency advantage at scale. Table 6 summarises these recommendations. The key insight is that neither system is universally superior: the right choice depends jointly on corpus scale and the error profile of the upstream mention detector.

Deployment condition	Recommended
Small corpus, high detector quality	FM
Large corpus ($n \gg$ 1k mentions)	CAR
Boundary errors dominate	CAR
Mention identity errors dominate	Improve detector

Table 6: Practical system selection guide based on deployment context.

8. Conclusion

We presented two unsupervised, fine-tuning-free systems for cross-document software mention coreference resolution at SOMD 2026, ranking second across all three subtasks. Our results answer three research questions. **RQ1**: The high lexical regularity of software mention chains makes unsupervised lexical methods strong baselines, competitive with contextual approaches without any task-specific supervision. **RQ2**: The two systems exhibit complementary robustness profiles: CAR is more robust to boundary noise, while FM degrades more gracefully under mention substitution, and neither is resilient to severe mention content corruption, pointing to upstream mention detection as the primary bottleneck for future progress. **RQ3**: FM is more efficient at a small scale while CAR scales more favourably to large corpora, making scale a decisive factor in system selection. Taken together, these findings provide concrete guidance for practitioners (Table 6): system selection should be informed by the upstream detector's noise profile and the scale of the target corpus, not by accuracy alone. We release our code[2] and hope that the noise-injection protocol introduced here serves as a reusable, robustness-evaluation framework for this underexplored task.

9. Bibliographical References

Atilla Kaan Alkan, Felix Grezes, Cyril Grouin, Fabian Schussler, and Pierre Zweigenbaum. 2024. Enriching a time-domain astrophysics corpus with named entity, coreference and astrophysical relationship annotations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6177–6188, Torino, Italia. ELRA and ICCL.

[2] https://github.com/adsabs/SOMD-2026

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.

Arthur Brack, Daniel Uwe Müller, Anett Hoppe, and Ralph Ewerth. 2021. Coreference Resolution in Research Papers from Multiple Domains. *arXiv e-prints*, page arXiv:2101.00884.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021a. Cross-document coreference resolution over predicted mentions. *CoRR*, abs/2106.01210.

Arie Cattan, Sophie Johnson, Daniel S. Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. 2021b. Scico: Hierarchical cross-document coreference for scientific concepts. *ArXiv*, abs/2104.08809.

Haixia Chai, Nafise Sadat Moosavi, Iryna Gurevych, and Michael Strube. 2022. Evaluating coreference resolvers on community-based question answering: From rule-based to state of the art. In *Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 61–73, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Panot Chaimongkol, Akiko Aizawa, and Yuka Tateisi. 2014. Corpus for coreference resolution on scientific papers. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3187–3190, Reykjavik, Iceland. European Language Resources Association (ELRA).

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

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.

Zhang Dong, Mingbang Wang, Songhang deng, Le Dai, Jiyuan Li, Xingzu Liu, and Ruilin Nong. 2025. Cross-Document Contextual Coreference Resolution in Knowledge Graphs. *arXiv e-prints*, page arXiv:2504.05767.

Caifan Du, Johanna Cohoon, Patrice Lopez, and James Howison. 2021. Softcite dataset: A dataset of software mentions in biomedical and economic research publications. *J. Assoc. Inf. Sci. Technol.*, 72(7):870–884.

Lior Forer and Tom Hope. 2024. Inferring scientific cross-document coreference and hierarchy with definition-augmented relational reasoning. *ArXiv*, abs/2409.15113.

Quentin Gliosca and Pascal Amsili. 2019. Résolution des coréférences neuronale : une approche basée sur les têtes (neural coreference resolution : a head-based approach). In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume II : Articles courts*, pages 409–416, Toulouse, France. ATALA.

Felix Grezes, Sergi Blanco-Cuaresma, Thomas Allen, and Tirthankar Ghosal. 2022. Overview of the first shared task on detecting entities in the astrophysics literature (DEAL). In *Proceedings of the First Workshop on Information Extraction from Scientific Publications*, pages 1–7, Online. Association for Computational Linguistics.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 385–393, Los Angeles, California. Association for Computational Linguistics.

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Yin Jou Huang and Sadao Kurohashi. 2021. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Daniel Jurafsky and James H. Martin. 2026. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*, 3rd edition. Online manuscript released January 6, 2026.

Nghia T. Le, Fan Bai, and Alan Ritter. 2022. Few-shot anaphora resolution in scientific protocols via mixtures of in-context experts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2693–2706, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nghia T. Le and Alan Ritter. 2023. Are large language models robust coreference resolvers?

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Lishuang Li, Liuke Jin, Zhenchao Jiang, Jing Zhang, and Degen Huang. 2014. Coreference resolution in biomedical texts. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 12–14.

Pengcheng Lu and Massimo Poesio. 2021. Coreference resolution for the biomedical domain: A survey. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 12–23, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Stasa Mandic, Georg Niess, and Roman Kern. 2025. From in-distribution to out-of-distribution: Joint loss for improving generalization in software mention and relation extraction. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 146–153, Vienna, Austria. Association for Computational Linguistics.

Vaghawan Ojha, Projan Shakya, Kristina Ghimire, Kashish Bataju, Ashwini Mandal, Sadikshya Gyawali, Manish Dahal, Manish Awale, Shital Adhikari, and Sanjay Rijal. 2025. SOMD 2025: Fine-tuning ModernBERT for in- and out-of-distribution NER and relation extraction of software mentions in scientific texts. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 154–163, Vienna, Austria. Association for Computational Linguistics.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Pranshu Rastogi and Rajneesh Tiwari. 2025. Extracting software mentions and relations using transformers and LLM-generated synthetic data at SOMD 2025. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 173–181, Vienna, Austria. Association for Computational Linguistics.

John W. Ratcliff and John A. Obershelp. 1988. JUL88: Pattern Matching: The Gestalt Approach. *Dr. Dobb's Journal*, 13:46–71.

D. Schindler, F. Bensmann, S. Dietze, and F. Krüger. 2022. The role of software in science: a knowledge graph-based analysis of software mentions in PubMed Central. *PeerJ Computer Science*, 8:e835.

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.

Isabel Segura-Bedmar, Mario Crespo, Cesar de Pablo, and Paloma Martínez. 2009. Drugnerar: linguistic rule-based anaphora resolver for drug-drug interaction extraction in pharmacological documents. In *Proceedings of the Third International Workshop on Data and Text Mining in Bioinformatics*, DTMBIO '09, page 19–26, New York, NY, USA. Association for Computing Machinery.

Gabriel Silva, Mário Rodriges, António Teixeira, and Marlene Amorim. 2025. Inductive learning on heterogeneous graphs enhanced by LLMs for software mention detection. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 164–172, Vienna, Austria. Association for Computational Linguistics.

Alessandro Stolfo, Chris Tanner, Vikram Gupta, and Mrinmaya Sachan. 2022. A simple unsupervised approach for coreference resolution using rule-based weak supervision. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 79–88, Seattle, Washington. Association for Computational Linguistics.

Julien Tourille, Olivier Ferret, Aurélie Névéol, and Xavier Tannier. 2020. Modèle neuronal pour la résolution de la coréférence dans les dossiers médicaux électroniques (neural approach for coreference resolution in electronic health records). In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, pages 351–360, Nancy, France. ATALA et AFCP.

Sharmila Upadhyaya, Wolfgang Otto, Frank Krüger, and Stefan Dietze. 2025. SOMD2025: A challenging shared tasks for software related information extraction. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 137–145, Vienna, Austria. Association for Computational Linguistics.

Marc B. Vilain, John D. Burger, John S. Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Message Understanding Conference*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Annual Meeting of the Association for Computational Linguistics*.

Dmitry Zelenko, Chinatsu Aone, and Jason Tibbetts. 2004. Coreference resolution for information extraction. In *Proceedings of the Conference on Reference Resolution and Its Applications*, pages 24–31, Barcelona, Spain. Association for Computational Linguistics.

Pierre Zweigenbaum, Guillaume Wisniewski, Marco Dinarelli, Cyril Grouin, and Sophie Rosset. 2012. Résolution des coréférences dans des comptes rendus cliniques. Une expérimentation issue du défi i2b2/VA 2011. In *Actes de la conférence RFIA 2012*, pages 978–2–9539515–2–3, Lyon, France. Session "Articles".

Do We Need Bigger Models for Science? Task-Aware Retrieval with Small Language Models

Florian Kelber[1], **Matthias Jobst**[1], **Yuni Susanti**[2], **Michael Färber**[1]
[1]ScaDS.AI, TU Dresden, Germany
[2]FIZ Karlsruhe, Germany
florian.kelber@student.tu-dresden.de, matthias.jobst@student.tu-dresden.de,
yuni.susanti@fiz-karlsruhe.de, michael.faerber@tu-dresden.de

Abstract

Scientific knowledge discovery increasingly relies on large language models, yet many existing scholarly assistants depend on proprietary systems with tens or hundreds of billions of parameters. Such reliance limits reproducibility and accessibility for the research community. In this work, we ask a simple question: *do we need bigger models for scientific applications?* Specifically, we investigate to what extent carefully designed retrieval pipelines can compensate for reduced model scale in scientific applications. We design a lightweight retrieval-augmented framework that performs *task-aware* routing to select specialized retrieval strategies based on the input query. The system further integrates evidence from full-text scientific papers and structured scholarly metadata, and employs compact instruction-tuned language models to generate responses with citations. We evaluate the framework across several scholarly tasks, focusing on scholarly question answering (QA), including single- and multi-document scenarios, as well as biomedical QA under domain shift and scientific text compression. Our findings demonstrate that retrieval and model scale are complementary rather than interchangeable. While retrieval design can partially compensate for smaller models, model capacity remains important for complex reasoning tasks. This work highlights retrieval and task-aware design as key factors for building practical and reproducible scholarly assistants.

Keywords: scientific question answering, retrieval-augmented generation, small language models

1. Introduction

The volume of scientific publications continues to grow rapidly, making it increasingly difficult for researchers to discover and synthesize relevant knowledge. Recent advances in large language models (LLMs) have shown strong potential for supporting scientific tasks such as question answering, summarization, and literature exploration. However, many scholarly assistants rely on proprietary models with tens or hundreds of billions of parameters, creating substantial barriers in terms of computational cost, accessibility, and reproducibility.

Beyond issues of scale, applying general-purpose LLMs to scientific literature presents additional challenges. Scientific documents are highly technical and domain-specific, and models may lack sufficient adaptation to scholarly language and reasoning patterns. As a result, existing systems can produce hallucinated or weakly supported claims when answering scientific questions in research papers (Wadden et al., 2025; Cai et al., 2025; Li et al., 2025; Shen et al., 2024). These limitations raise concerns about reliability and transparency, which are central to scientific applications.

Early efforts to build science-focused language models highlight both the promise and the limitations of current approaches. Systems such as Galactica attempted to encode large amounts of scientific knowledge directly into model parameters, but encountered challenges related to factual accuracy and verification (Taylor et al., 2022; Marcus, 2022). Other approaches incorporate retrieval mechanisms or domain-specific datasets, yet many still depend on large proprietary backbones or remain limited to specific domains. For example, biomedical QA systems built around datasets such as PubMedQA (Jin et al., 2019) demonstrate the benefits of targeted retrieval, but do not necessarily generalize across the broader scientific ecosystem.

Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm for improving reliability and transparency by grounding model outputs in external evidence (Lewis et al., 2020). In scientific domain, prior work has explored knowledge-graph-based retrieval, domain-specific training, and hybrid approaches that combine multiple sources of evidence (Jaradeh et al., 2020; Asai et al., 2024). While these approaches improve grounding, hallucinations remain a persistent challenge (Mishra et al., 2024; Mallen et al., 2023), and many systems continue to rely on large backbone models.

In this work, we examine a simple question: *do we need bigger models for scientific applications?* Specifically, we investigate the extent to which improvements in retrieval design can support the use of smaller models in these settings. Rather than asking whether retrieval can replace model scaling entirely, we focus on the conditions under which retrieval strategies can support smaller models effectively, and the trade-offs that arise in doing so. In particular, we analyze how retrieval, domain cover-

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 108–118
12 May 2026.

age, and task structure interact with model capacity in scholarly question answering and related tasks.

To this end, we design a lightweight retrieval-augmented framework that incorporates *task-aware routing* prior to retrieval. Incoming queries are first classified into scholarly task categories, allowing the system to select specialized retrieval strategies tailored to different information needs. The framework integrates evidence from both unstructured scientific documents and structured scholarly metadata from knowledge graph within a unified pipeline. Specifically, we combine a compact 3B-parameter instruction-tuned language model with retrieval over a corpus of 165K research papers (unarXive (Saier et al., 2023)) and a large scholarly knowledge graph (SemOpenAlex (Färber et al., 2023)). This hybrid design enables support for multiple tasks, including scholarly question answering, summarization, and factual metadata queries.

We evaluate the framework across several scholarly tasks, focusing on scholarly QA (ScholarQABench-Multi (Asai et al., 2024))–including both single- and multi-document scenarios–as well as biomedical QA under domain shift (PubMedQA (Jin et al., 2019)) and scientific text compression (SciTLDR (Cachola et al., 2020)). Our results show that small open-weight models can approach the performance of larger systems when paired with appropriate retrieval strategies. However, we also found that system performance remains sensitive to retrieval quality, prompt length, and domain mismatch, highlighting limitations in robustness and generalization. Overall, our findings suggest that improved retrieval design can partially compensate for smaller models. Nevertheless, model capacity remains important for complex reasoning tasks, which means that retrieval and model scale are complementary rather than interchangeable.

Our contributions are summarized as follows:

- We design a task-aware routing strategy that selects retrieval methods based on the information needs of scholarly queries.
- We introduce a hybrid retrieval framework integrating scientific text collections with structured scholarly knowledge graphs.
- We empirically analyze the extent to which compact open-weight models, combined with targeted retrieval, can approach the performance of larger systems, highlighting key trade-offs in robustness and generalization.
- We release resources and source code to support reproducibility and further research.[1]

[1] https://github.com/faerber-lab/lightweight-scholarly-qa

2. Related Work

Retrieval-Augmented Systems for Scholarly QA Retrieval-Augmented Generation (RAG) has become a dominant paradigm for improving factual grounding in language models (Lewis et al., 2020). In the scholarly domain, several systems combine large language models with retrieval from scientific corpora. For example, OpenScholar retrieves and reranks paper sections from Semantic Scholar, incorporating feedback loops and citation verification mechanisms to improve reliability (Asai et al., 2024). Large-scale resources such as unarXive (Saier et al., 2023) and SciQA (Auer et al., 2023) provide curated datasets for scientific question answering and retrieval-based experimentation.

While these approaches demonstrate the effectiveness of retrieval in scientific settings, many rely on heavyweight reranking pipelines or large backbone models. In contrast, our work investigates whether a lightweight architecture can achieve competitive performance without complex reranking or large-scale parametric knowledge. Rather than scaling model size, we focus on structuring the retrieval process through task-aware routing.

Knowledge Graph-Based Scholarly Information Access Knowledge graphs provide a complementary perspective on scholarly knowledge by representing publications, authors, and venues as structured entities and relations. Several large-scale scholarly KGs have been introduced, including SemOpenAlex (Färber et al., 2023), Semantic Scholar (Kinney et al., 2023), ORKG (Auer et al., 2020), and MAKG (Färber and Ao, 2022). These resources enable structured querying through languages such as SPARQL and support applications including metadata exploration and scientific discovery. Prior work has investigated machine-learning-driven interfaces that map natural language queries to KG queries. Our approach builds on this line of work by integrating KG retrieval within a broader retrieval-augmented generation framework. Rather than focusing exclusively on KG-based QA, we combine structured metadata with text-based evidence to support a wider range of scholarly tasks.

Plain Language Summarization of Scientific Literature Another important line of research focuses on improving the accessibility of scientific documents through plain language summarization (PLS). These systems aim to condense complex texts into clear summaries that can be understood by both expert and non-expert audiences (August et al., 2024). Techniques often involve interpreting domain-specific terminology, adding explanatory context, and removing redundant or overly technical details (Guo et al., 2024). Maintaining fac-

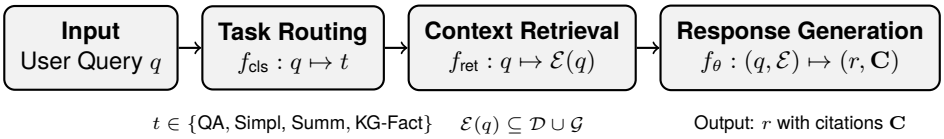


Figure 1: Task-aware retrieval pipeline. Task routing determines which processing strategy and data source are used before response generation. The input query q is classified into a task t, used to retrieve relevant context $\mathcal{E}(q)$ from data sources $\mathcal{D}$ and $\mathcal{G}$, and passed to a lightweight LLM to generate the response r with citations $\mathbf{C}$.

tual accuracy while improving readability remains challenging, particularly for specialized domains such as biomedical literature (Joseph et al., 2024). Evaluation is also non-trivial: traditional metrics such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2020) capture lexical similarity but often fail to fully measure informativeness or faithfulness (Guo et al., 2024; Luo et al., 2022; Ondov et al., 2022). To evaluate summarization capabilities within our framework, we use the SciTLDR dataset (Cachola et al., 2020), which focuses on concise summaries of computer science publications and has been widely used in prior work (Takeshita et al., 2024). Earlier experiments using environments such as CATTS extended these tasks with title generation and denoising objectives (Cachola et al., 2020).

Despite advances in retrieval and summarization, many scholarly assistants continue to rely on large proprietary models. This dependence limits reproducibility and raises the barrier to entry for academic institutions without extensive computational resources. Recent studies have highlighted both the benefits and limitations of adapting general-purpose LLMs to scientific domains (Wadden et al., 2025; Cai et al., 2025; Li et al., 2025; Shen et al., 2024), as well as the persistence of hallucinations in scholarly settings (Mishra et al., 2024; Mallen et al., 2023).

Positioning of this Work. Overall, prior work demonstrates the value of retrieval-augmented generation, domain-specific datasets, and knowledge graphs for scholarly question answering. However, fewer works investigate how these components interact within a lightweight architecture. While prior work often emphasizes scaling model size, fewer studies systematically analyze the trade-offs between retrieval design and model capacity. Our work addresses these gaps in two key ways. First, we explicitly examine whether improved retrieval design can compensate for a reduced model scale. Second, we integrate task-aware routing and hybrid retrieval (text + structured scholarly metadata) within a lightweight pipeline. We next describe the proposed framework in detail.

3. Task-Aware Hybrid Retrieval for Scholarly Applications

Scholarly assistants must handle heterogeneous information needs, ranging from factual metadata queries to multi-document reasoning over scientific literature. Treating these requests uniformly often leads to inefficient retrieval and unnecessary load on large language models.

We design a task-aware retrieval framework that routes user queries to specialized retrieval strategies before generation. The framework combines (i) task classification and routing with a lightweight classifier, (ii) hybrid context evidence retrieval from both scientific text collections and knowledge graphs, and (iii) response generation using a small language model. Figure 1 illustrates the overall architecture of the proposed framework.

Design Rationale Our design is guided by two observations. First, many scholarly queries can be solved through targeted retrieval rather than increased model scale. For example, a question "Which papers propose methods for protein structure prediction?" can be answered by retrieving relevant abstracts or datasets instead of relying on a massive language model to memorize all literature. Second, queries about scholarly metadata are better answered through knowledge graphs or structured database than through text generation alone. For instance, a question "Who are the co-authors of Zang et al., 2023?" is more accurately answered by querying structured bibliographic databases or knowledge graphs. These observations motivate the combination of task-aware routing and hybrid retrieval within a lightweight generation framework.

Formally, given a query $q \in \mathcal{Q}$, the system retrieves evidence

$$\mathcal{E}(q) = \{e_1, ..., e_M\}$$

from a textual corpus $\mathcal{D}$ and/or scholarly knowledge graph $\mathcal{G}$. A lightweight language model then produces an answer

$$r = f_\theta(q, \mathcal{E}(q)).$$

Each answer is accompanied by a citation set

$$\mathbf{C} = \{c_1, ..., c_K\}, \quad c_i \in \mathcal{D} \cup \mathcal{G},$$

allowing the system to expose the evidence underlying generated statements.

3.1. Task Routing

User queries in scholarly assistants are highly heterogeneous. A request for a paper summary requires different evidence than a factual query asking about an author's affiliation. Applying a single retrieval strategy to all queries therefore introduces unnecessary noise and computational cost. We introduce a lightweight task routing module that predicts the information need prior to retrieval. Given a query q, a classifier assigns a task label

$$f_{\text{cls}} : q \mapsto t \in \mathcal{T}$$

where

$$\mathcal{T} = \{\texttt{General QA}, \texttt{Simplification}, \texttt{Summarization}, \texttt{KG-Fact}\}$$

The four task categories reflect distinct information needs that require different retrieval strategies. General QA involves multi-document reasoning over unstructured text, while summarization and simplification require transformation of specific documents. In contrast, KG-Fact queries target structured metadata that can be answered more reliably via knowledge graphs than text retrieval. This taxonomy is not intended to be exhaustive but rather to capture common patterns in scholarly information-seeking. Future work could explore learned or hierarchical task taxonomies.

We implement the classifier using a fine-tuned small language model (*Llama 3.2 3B Instruct*). If classifier confidence falls below a predefined threshold, the system defaults to the general QA retrieval pathway. The predicted task t determines the retrieval strategy used in the next stage, as summarized in Table 1. This routing mechanism reduces unnecessary retrieval noise and allows the system to allocate resources more efficiently.

3.2. Context Evidence Retrieval

Traditional RAG systems typically rely on unstructured text corpora. However, scholarly information exists both in textual form and in structured metadata sources such as citation graphs and author databases. Unlike conventional RAG pipelines relying solely on unstructured text corpora, our framework integrates both unstructured scientific documents and structured scholarly knowledge graphs. This hybrid design enables the system to answer both narrative questions and precise metadata queries within a unified architecture.

Once a query has been classified into a specific task, the system retrieves task-appropriate *evidence* from curated text corpora or structured knowledge graphs. The context retrieval step thus serves as the bridge between user intent and generation, ensuring that responses are grounded in identifiable sources. Depending on the task category, one of the following retrieval strategies is invoked, as summarized in Table 1.

QA: General Scholarly Question Answering. For general QA tasks, text passages are retrieved as evidence from the open subset of the *unarXive* corpus (Saier et al., 2023), comprising 165K publications. Documents are preprocessed into *markdown* containing metadata and plain text (tables and figures removed). The text is segmented into sections or subsections, further subdivided into chunks of at least 800 characters. Each chunk is embedded using Sentence Transformers (*all-MiniLM-L6-v2*) into 384-dimensional vectors. Both vector embeddings and metadata (e.g., *title, authors, venue*) are stored in a FAISS vector index (Douze et al., 2024; Chase, 2022). At query time, the user query q is embedded in the same space, and the top-k most similar chunks based on vector similarity are retrieved:

$$\mathcal{E}_{\text{QA}}(q) = \text{Top-}k\big(\text{FAISS}(\text{Embed}(q))\big).$$

Retrieved chunks are assigned reference numbers and appended to the user query, following the multi-paper QA prompt design from the OpenScholar (Asai et al., 2024). To enrich bibliographic accuracy, the pipeline also issues secondary lookups to the Semantic Scholar API (Kinney et al., 2023) using paper titles from the retrieved set. References are tracked for the generation of a corresponding bibliographic list.

Simplification and Summarization For simplification and summarization tasks, the system attempts to detect whether the user refers to a specific scientific paper. A Named Entity Recognizer (NER) component is used to identify candidate paper titles, and if a specific title is detected, a fuzzy match is performed against the corpus (*unarXive* corpus (Saier et al., 2023)) to retrieve the full text of the corresponding paper. If no reliable title match is found, the system assumes that the query refers to arbitrary input text rather than a specific paper and performs summarization or simplification directly on the provided input. This fallback ensures that the system remains functional even when grounding is not possible.

The NER component is implemented using a spaCy *EntityRecognizer*[2] trained on synthetic data generated from unarXive titles and manually designed prompt templates. This step enables the system to ground summaries in the full paper when possible, while still supporting general text queries.

[2] https://spacy.io/api/entityrecognizer

Task Category	Retrieval Strategy	Output Type
General QA	Passage retrieval from unarXive; metadata enrichment via Semantic Scholar API	Factual answer with inline citations
Simplification, Summarization	NER-based paper title detection; fuzzy match against unarXive full-text corpus	Simplified or summarized text with optional paper grounding
KG-Fact	SPARQL queries over SemOpenAlex; 18 predefined templates (author/work)	Structured metadata (e.g., h-index, DOI, ORCID, affiliations)

Table 1: Overview of task categories, retrieval strategies, and output in the classification module.

KG-Fact: Structured Scholarly Metadata Retrieval. Certain scholarly queries request structured metadata such as citation metrics, identifiers, or affiliations. These questions are better addressed through knowledge graphs rather than through text generation alone. For such queries, the system maps user questions to one of 18 predefined SPARQL templates, focusing on author and scientific work-related queries. A rule-based precheck detects explicit mentions of scholarly identifiers such as *h-index*, *i10-index*, *ORCID*, or *DOI*. If such identifiers are not present, an NER component extracts candidate author or paper entities, and a classifier assigns the query to the most appropriate SPARQL template. The resulting query is executed against the SemOpenAlex knowledge graph via its SPARQL endpoint, returning structured metadata $\mathcal{E}_{\text{KG}}(q)$ in a verifiable format.

Our template-driven approach to knowledge graphs provides a reliable and interpretable interface for knowledge graph access. The current implementation uses a fixed set of templates, but these can be readily extended as new query types are added.

Final Prompt Composition

Across all pathways, context retrieval produces an evidence set $\mathcal{E}(q)$ tailored to the classified task t. Formally:

$$\mathcal{E}(q,t) = \begin{cases} \mathcal{E}_{\text{QA}}(q) & t = \textit{QA}, \\ \mathcal{E}_{\text{Simpl}}(q) & t = \textit{Simpl}, \\ \mathcal{E}_{\text{Summ}}(q) & t = \textit{Summ}, \\ \mathcal{E}_{\text{KG}}(q) & t = \textit{KG-Fact}. \end{cases}$$

After retrieval, the system constructs a task-specific prompt $\mathcal{P}$ by concatenating the original user query q with retrieved evidence $\mathcal{E}(q,t)$. The retrieved evidence may consist of text passages or structured knowledge-graph outputs depending on the routing decision.

$$\mathcal{P}(q,\mathcal{E},t) = \text{Template}(t) \parallel \mathcal{E} \parallel q,$$

where $\parallel$ denotes string concatenation and $\text{Template}(t)$ is an instruction prefix corresponding to the predicted task (e.g., *"Summarize the following paper:"*, *"Answer the following question..."*). This explicit prompt structure encourages the generator to rely on retrieved evidence rather than parametric memory.

3.3. Response Generation

Instead of relying on large proprietary models, we employ a compact open model (Llama 3.2 3B Instruct) as the response generator. This design choice reflects our central hypothesis: strong retrieval can compensate for smaller model capacity. The model is instruction-tuned on the OpenScholar (Asai et al., 2024) dataset to support scholarly tasks including multi-paper QA, summarization, and editing.

Fine-tuning is performed using the *SFTTrainer* framework (von Werra et al., 2020) with BF16 precision. Multiple variants are trained with either Low-Rank Adapters (LoRA) (Hu et al., 2022) of rank $r = 32$ or full fine-tuning. Training configurations include variable numbers of epochs $\{1, 2, 5\}$ and context lengths $\{8192, 10000, 16384\}$. All models use the *AdamW* optimizer (Loshchilov and Hutter, 2019) with a *cosine-annealing* learning rate schedule, initial learning rate $5 \cdot 10^{-6}$, 200 warmup steps, and an effective batch size of 64. We provide the detailed implementation settings in our repository.

During inference, the system receives the composed prompt $\mathcal{P}(q,\mathcal{E},t)$ constructed in the previous stage, the model generates a response r and a set of citations $\mathbf{C}$:

$$r, \mathbf{C} = g_\phi(\mathcal{P}(q,\mathcal{E},t))$$
$$g_\phi : \mathcal{P} \mapsto (\text{Textual Answer}, \text{Citations})$$

where g_ϕ denotes the fine-tuned LLM parameterized by ϕ. The generated answer includes references corresponding to the retrieved evidence passages. Model serving is performed using *vLLM* (Kwon et al., 2023).

Model	Org↑	Cov↑	Rel↑	Mean↑	Length	Citation Quality		
						R↑	P↑	F1↑
Llama3-8B †*	–	–	–	3.79	–	–	–	0.0%
OS-8B *	3.92	4.44	4.02	4.12	578.6	–	–	42.8%
Llama 3.2 3B Instruct †	3.96	4.06	4.54	4.19	450	0.00%	0.00%	0.00%
Llama 3.1 8B Instruct †	4.01	4.19	4.76	4.32	475	0.00%	0.00%	0.00%
Llama 3.2 3B Instruct	3.67	3.51	3.99	3.72	363	16.49%	19.05%	17.68%
Llama 3.1 8B Instruct	3.83	3.90	4.38	**4.04**	453	23.50%	**26.45%**	**24.89%**
LoRA FT 1ep 8k	3.74	3.78	4.20	3.91	784	23.41%	21.06%	22.18%
LoRA FT 1ep 16k	3.60	3.87	4.08	3.85	1,017	23.35%	21.63%	22.46%
LoRA FT 2ep 8k	3.65	3.81	4.18	3.88	841	24.74%	23.72%	24.22%
LoRA FT 2ep 10k	3.59	**3.92**	**4.31**	3.94	1,297	**25.64%**	22.77%	24.12%
LoRA FT 2ep 16k	3.55	3.82	4.21	3.86	887	24.07%	25.02%	24.53%
LoRA FT 5ep 16k	3.55	3.80	4.27	3.87	1,024	20.10%	18.75%	19.40%
Full FT 1ep 16k	3.81	3.80	4.20	3.94	532	24.51%	23.33%	23.91%
Full FT 2ep 10k	3.85	3.83	4.19	3.96	468	24.52%	22.69%	23.57%
Full FT 2ep 16k	**3.89**	3.88	4.23	4.00	466	21.20%	19.66%	20.40%
Full FT 5ep 16k	3.82	3.84	4.26	3.98	484	23.81%	23.96%	23.89%

Table 2: Multi-Paper QA Evaluation results (ScholarQABench-Multi dataset). Best value per *model type* is underlined; overall best values (excluding external baselines) are **bolded**; LoRA FT rows denote LoRA fine-tuning applied to Llama 3.1 8B Instruct; * results from (Asai et al., 2024); † without retrieval.

4. Evaluation

We evaluate our framework to assess whether our design combining task-aware retrieval and lightweight model can effectively support scientific applications. Our evaluation focuses on the extent to which small language models, when combined with task-aware retrieval strategies, can achieve competitive performance on scholarly tasks. In addition, we examine the robustness of the system under varying retrieval conditions, including differences in retrieval quality and domain alignment between the corpus and evaluation datasets. We select evaluation datasets to reflect complementary aspects of scholarly tasks:

1. **ScholarQABench-Multi (Asai et al., 2024)** for multi-document QA and reasoning,

2. **PubMedQA (Jin et al., 2019)** for domain transfer and robustness, and

3. **SciTLDR (Cachola et al., 2020)** for extreme summarization.

All experiments follow the pipeline described in Section 3. Incoming queries are first routed to a task category, relevant context is retrieved, and the composed prompt is passed to the language model for response generation. We evaluate the framework on two primary scholarly tasks: **scientific question answering** (§4.1) and **text compression** (§4.2), detailed in the following sections.

4.1. Scientific Question Answering Evaluation

For scientific question answering, we evaluate in two settings: (1) multi-paper QA (§4.1.1), which requires synthesizing information across several documents, and (2) single-paper QA (§4.1.2), focusing on questions grounded in a single paper.

4.1.1. Multi-Paper Question Answering

We evaluate the framework on multi-paper question answering using the ScholarQABench-Multi benchmark (Asai et al., 2024). The dataset contains 108 questions spanning computer science, physics, and biomedical research. Answers are evaluated using LLM-based judges (Prometheus models) across three dimensions: Organization (*Org*), Relevance (*Rel*), and Coverage (*Cov*). Each dimension is scored on a 1–5 scale and averaged to obtain an overall quality score (*Mean*). In addition, citation quality is measured using Precision, Recall, and F1 with respect to gold references. Table 2 summarizes our evaluation results.

Impact of Retrieval. We first compare base models with and without retrieval augmentation. Surprisingly, LLM evaluation scores are slightly higher when retrieval is disabled. This suggests that long prompts containing multiple retrieved passages can make generation more difficult for smaller models. However, without retrieval the models are unable to produce verifiable citations as expected, resulting in

Model	Orig Acc↑	Orig F1↑	OS Acc↑	OS F1↑	Zero Acc↑	Zero F1↑
BioBERT †	68.08	52.72	–	–	–	–
OS-8B *	–	–	**76.4**	–	–	–
Llama 3.2 3B Instruct	62.40	40.93	49.11	49.04	64.41	58.97
Llama 3.1 8B Instruct	**75.60**	**54.06**	45.91	43.35	**64.77**	**59.74**
LoRA FT 2ep 10k context	68.00	46.69	58.01	55.83	58.13	57.23
Full FT 5ep 16k context	62.80	38.69	58.01	**56.21**	60.62	57.24

Table 3: Single-paper QA evaluation results (PubMedQA dataset). Best values are **bolded**. † results from (Jin et al., 2019); * results from (Asai et al., 2024).

Model	R1 F1↑	R2 F1↑	RL F1↑	BERTScore F1↑	SMOG Index↑	Compression Ratio↑
CATT (Cachola et al., 2020)	**44.9**	**22.6**	**37.3**	–	–	**47.3**
Llama 3.2 3B Instruct	1.21	0.011	1.21	54.55	24.45	20.72
Llama 3.1 8B Instruct	1.21	**0.014**	1.21	54.57	**24.81**	20.79
Full FT 5ep 16k context	**1.31**	**0.014**	**1.30**	**54.74**	23.26	**22.43**

Table 4: Text compression evaluation results (SciTLDR dataset). Best values are **bolded**. R1, R2, and RL denote ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. Scores are averages over test samples.

a citation F1 score of 0. Retrieval therefore remains essential for grounding responses in evidence.

Fine-Tuning Effects. Fine-tuning substantially improves both answer quality and citation grounding. Our fully fine-tuned 3B model approaches the performance of a substantially larger 8B model. Low-Rank Adaptation (LoRA) improves performance relative to the base model but occasionally produces repetitive outputs or unstable citation formatting, which negatively affects organization scores. Across experiments, training duration and context length have limited impact on overall performance, though longer training slightly degrades organization for LoRA models.

These results on multi-paper QA highlight a key trade-off: while retrieval enables citation grounding, it can introduce noise that negatively affects answer organization and fluency, particularly for smaller models with limited context handling capacity.

4.1.2. Single-Paper Question Answering

We evaluate single-paper question answering using the biomedical PubMedQA (Jin et al., 2019) dataset to further assess retrieval robustness and domain generalization. The task requires predicting *yes*, *maybe*, or *no* answers to biomedical questions. We report our evaluation results for the best-performing LoRA and full fine-tuned models, as well as pretrained Llama 3.1 8B and 3B models as comparison. The results are summarized in Table 3.

We follow the retrieval-based evaluation protocol introduced in ScholarQABench (Asai et al., 2024) and consider the following three evaluation settings.

Original Task (Orig). In the original task setup, models are provided with gold context passages from the relevant paper abstracts. Under this setup, larger models achieve higher accuracy, indicating stronger reasoning ability when reliable evidence is available. The LoRA-adapted 3B model shows notable improvement over its base version despite not being trained on PubMedQA, suggesting that lightweight adaptation can improve domain transfer.

Retrieval Task (OS) In the retrieval variant, models must identify relevant passages from the corpus. The performance drop in the retrieval setting reflects a domain mismatch between the unarXive corpus (primarily computer science and physics) and PubMedQA (biomedical domain). This setup allows us to explicitly evaluate robustness under domain shift. Nevertheless, we show that the fine-tuned models outperform their pretrained counterparts, indicating that training improves the model's ability to filter irrelevant context by ignoring irrelevant retrievals and leveraging retrieved content.

Zero-Context Task (Zero). Finally, we evaluate models without any retrieved evidence. In this setting, answers therefore rely entirely on the models' parametric knowledge. Pretrained models perform slightly better than fine-tuned ones, suggesting that task-specific training may introduce mild

specialization that reduces general-domain recall. This highlights the trade-off between fine-tuning for domain-specific QA and preserving general domain knowledge.

4.2. Text Compression Evaluation

To evaluate summarization capabilities, we benchmark the system on the SciTLDR (Cachola et al., 2020) dataset, which focuses on extreme compression of scientific papers. The task requires generating a one-sentence summary from the abstract, introduction, and conclusion (AIC) sections. Table 4 summarizes our evaluation results.

Generated summaries are compared against the gold TLDR statement summaries reviewed by authors and peer reviewers. Overlap metrics were computed using ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004), as well as BERTScore (Zhang et al., 2020). Following prior work (Cachola et al., 2020), we report the maximum target score for each metric to reduce variability across reference summaries. In addition, we compute a compression ratio between the source text and generated summaries. Because individual summaries consist of a single sentence, traditional readability metrics (*Flesch, Flesch-Kincaid, SMOG*) cannot be applied per example. Thus, we concatenate outputs and compute aggregate readability scores[3] to approximate lexical and syntactic complexity.

We compare our fine-tuned 3B model with baseline Llama models and the specialized CATT model used in the original SciTLDR benchmark. The results show modest improvements from fine-tuning. ROUGE and BERTScore increase slightly, and the compression ratio improves relative to the base model. However, generated summaries remain longer than the target TLDR statements, and readability metrics suggest relatively dense language. These findings highlight the difficulty of extreme scientific summarization and confirm that models specifically trained for the task continue to outperform general-purpose LLMs.

5. Conclusion

We presented a lightweight retrieval-augmented framework for scholarly assistance that combines task-aware routing, hybrid retrieval, and compact language models within a unified architecture. Rather than relying on increasingly large proprietary systems, our work investigates under which conditions improved retrieval design can compensate for reduced model scale across scholarly tasks.

Our evaluation results show that small instruction-tuned models can approach the performance of larger systems when paired with appropriate retrieval strategies, particularly for tasks requiring grounded, citation-based answers. Fine-tuning further improves robustness in multi-document question answering and helps mitigate the impact of partially irrelevant context. However, these gains are not uniform: model capacity remains a key factor for reasoning quality, especially in complex or domain-shifted settings.

Importantly, our findings highlight that improvements in retrieval introduce inherent trade-offs. While more precise retrieval can improve grounding and interpretability, it may reduce recall or introduce longer, noisier contexts that disproportionately affect smaller models. In addition, retrieval pipelines incur additional engineering complexity and may generalize poorly when the underlying corpus does not match the target domain, as observed in our biomedical QA evaluation. These results emphasize that retrieval and model scale should be viewed as complementary components rather than interchangeable solutions.

Overall, our study suggests that progress in retrieval quality is as critical as progress in model scaling for building practical and efficient scholarly assistants. Future work will focus on improving retrieval robustness, expanding domain coverage, and incorporating automated verification of generated outputs. We also plan to explore efficiency-oriented improvements, such as more accurate routing strategies and lightweight reranking, alongside broader evaluations that assess usability, readability, and trust in real-world scholarly workflows.

6. Limitations

First, although the proposed design integrates structured scholarly metadata from knowledge graph and textual evidence within a single pipeline, a standardized benchmark for question answering over the SemOpenAlex knowledge graph is currently unavailable. Thus, current evaluation focuses on the text-based components, while the KG-Fact module is presented primarily as an architectural capability. Constructing reliable SPARQL-question pairs for large scholarly knowledge graphs remains an open challenge. Second, retrieval quality remains a bottleneck. Dense vector search may return passages that are only loosely related to the query, reducing grounding quality and potentially leading to weak or partially supported citations.

Finally, the current system does not include a post-generation verification step to validate citations against source documents. Incorporating reference validation or retrieval-based fact checking would be an important step toward more reliable scholarly assistants.

[3] https://github.com/cdimascio/py-readability-metrics

7. Ethical Considerations

All datasets used are publicly available under research-friendly licenses, e.g., ScholarQABench-Multi (Asai et al., 2024) is released under the ODC-BY license, with some constituent datasets subject to their own licensing terms. Our 165K-paper datastore unarXive (Saier et al., 2023) comprises open-access content compliant with text and data mining permissions. The system is designed as an assistive tool for scholarly tasks and is not intended for use in high-stakes domains without human oversight. We acknowledge potential biases in the underlying literature (e.g., publication bias, domain skew) that may influence system behavior. Additionally, our approach encourages the use of smaller or lightweight models (e.g., 3B parameters) over larger models to reduce computational and environmental impact.

8. Bibliographical References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. 2024. OpenScholar: Synthesizing Scientific Literature with Retrieval-augmented LMs.

Tal August, Kyle Lo, Noah A. Smith, and Katharina Reinecke. 2024. Know your audience: The benefits and pitfalls of generating plain language summaries beyond the "general" audience. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI 2024, New York, NY, USA.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777.

Hengxing Cai, Xiaochen Cai, Junhan Chang, Sihang Li, Lin Yao, Changxin Wang, Zhifeng Gao, Hongshuai Wang, Yongge Li, Mujie Lin, Shuwen Yang, Jiankun Wang, Mingjun Xu, Jin Huang, Xi Fang, Jiaxi Zhuang, Yuqi Yin, Yaqi Li, Changhong Chen, Zheng Cheng, Zifeng Zhao, Linfeng Zhang, and Guolin Ke. 2025. Sciassess: Benchmarking LLM proficiency in scientific literature analysis. In *Findings of the Association for Computational Linguistics*, NAACL 2025, pages 2335–2357.

Harrison Chase. 2022. LangChain.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazar'e, Maria Lomeli, Lucas Hosseini, and Herv'e J'egou. 2024. The faiss library. *ArXiv*, abs/2401.08281.

Yue Guo, Tal August, Gondy Leroy, Trevor Cohen, and Lucy Lu Wang. 2024. APPLS: Evaluating evaluation metrics for plain language summarization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2024, pages 9194–9211.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, ICLR 2022.

Mohamad Yaser Jaradeh, Markus Stocker, and Sören Auer. 2020. Question answering on scholarly knowledge graphs. In *Proceedings of the 24th International Conference on Theory and Practice of Digital Libraries*, TPDL 2020, pages 19–32.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. 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 2019, pages 2567–2577.

Sebastian Joseph, Lily Chen, Jan Trienes, Hannah Göke, Monika Coers, Wei Xu, Byron Wallace, and Junyi Jessy Li. 2024. FactPICO: Factuality evaluation for plain language summarization of medical evidence. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, ACL 2024, pages 8437–8464.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, Miles Crawford, Doug Downey, Jason Dunkelberger, Oren Etzioni, Rob Evans, Sergey Feldman, Joseph Gorney, David Graham, Fangzhou Hu, Regan Huff, Daniel King, Sebastian Kohlmeier, Bailey Kuehl, Michael Langan, Daniel Lin, Haokun Liu, Kyle Lo, Jaron Lochner, Kelsey MacMillan, Tyler Murray, Chris Newell, Smita Rao, Shaurya Rohatgi, Paul Sayre, Zejiang Shen, Amanpreet Singh, Luca Soldaini, Shivashankar Subramanian, Amber Tanaka, Alex D. Wade, Linda Wagner, Lucy Lu Wang, Chris Wilhelm,

Caroline Wu, Jiangjiang Yang, Angele Zamarron, Madeleine Van Zuylen, and Daniel S. Weld. 2023. The semantic scholar open data platform.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP 2023, pages 611–626.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, NeurIPS 2020.

Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2025. Scilitllm: How to adapt llms for scientific literature understanding. In *Proceedings of the Thirteenth International Conference on Learning Representations*, ICLR 2025.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*, ICLR 2019.

Junyu Luo, Junxian Lin, Chi Lin, Cao Xiao, Xinning Gui, and Fenglong Ma. 2022. Benchmarking automated clinical language simplification: Dataset, algorithm, and evaluation. In *Proceedings of the 29th International Conference on Computational Linguistics*, COLING 2022, pages 3550–3562, Gyeongju, Republic of Korea.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, ACL 2023, pages 9802–9822, Toronto, Canada.

Gary Marcus. 2022. The galactica ai model was trained on scientific knowledge – but it spat out alarmingly plausible nonsense. Accessed: 2025-09-17.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. In *Proceedings of the First Conference on Language Modeling*, COLM 2024.

Brian Ondov, Kush Attal, and Dina Demner-Fushman. 2022. A survey of automated methods for biomedical text simplification. *Journal of the American Medical Informatics Association*, 29(11):1976–1988.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL 2002, page 311–318, USA.

Tarek Saier, Johan Krause, and Michael Färber. 2023. unarXive 2022: All arXiv Publications Pre-Processed for NLP, Including Structured Full-Text and Citation Network. In *Proceedings of the 2023 ACM/IEEE Joint Conference on Digital Libraries*, JCDL 2023, pages 66–70.

Junhong Shen, Neil A. Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolò Fusi. 2024. Tag-llm: Repurposing general-purpose llms for specialized domains. In *Proceedings of the 41st International Conference on Machine Learning*, ICML 2024, pages 44759–44773.

Sotaro Takeshita, Simone Paolo Ponzetto, and Kai Eckert. 2024. ROUGE-K: do your summaries have keywords? In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics*, *SEM 2024, pages 69–79.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning.

David Wadden, Kejian Shi, Jacob Morrison, Alan Li, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, Shannon Zejiang Shen, Doug Downey, Hannaneh Hajishirzi, and Arman Cohan. 2025. Sciriff: A resource to enhance language model instruction-following over scientific literature. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2025, pages 6072–6109.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *Proceedings of the 8th International Conference on Learning Representations*, ICLR 2020.

9. Language Resource References

Asai, Akari and He, Jacqueline and Shao, Rulin and Shi, Weijia and Singh, Amanpreet and Chang, Joseph Chee and Lo, Kyle and Soldaini, Luca and Feldman, Sergey and D'arcy, Mike and Wadden, David and Latzke, Matt and Tian, Minyang and Ji, Pan and Liu, Shengyan and Tong, Hao and Wu, Bohao and Xiong, Yanyu and Zettlemoyer, Luke and Neubig, Graham and Weld, Dan and Downey, Doug and Yih, Wen-tau and Koh, Pang Wei and Hajishirzi, Hannaneh. 2024. *OpenScholar: Synthesizing Scientific Literature with Retrieval-augmented LMs*.

Auer, Sören and Barone, Dante A. C. and Bartz, Cassiano and Cortes, Eduardo G. and Jaradeh, Mohamad Yaser and Karras, Oliver and Koubarakis, Manolis and Mouromtsev, Dmitry and Pliukhin, Dmitrii and Radyush, Daniil and Shilin, Ivan and Stocker, Markus and Tsalapati, Eleni. 2023. *The SciQA Scientific Question Answering Benchmark for Scholarly Knowledge*.

Sören Auer and Allard Oelen and Muhammad Haris and Markus Stocker and Jennifer D'Souza and Kheir Eddine Farfar and Lars Vogt and Manuel Prinz and Vitalis Wiens and Mohamad Yaser Jaradeh. 2020. *Improving Access to Scientific Literature with Knowledge Graphs*.

Cachola, Isabel and Lo, Kyle and Cohan, Arman and Weld, Daniel. 2020. *TLDR: Extreme Summarization of Scientific Documents*.

Färber, Michael and Ao, Lin. 2022. *The Microsoft Academic Knowledge Graph enhanced: Author name disambiguation, publication classification, and embeddings*.

Färber, Michael and Lamprecht, David and Krause, Johan and Aung, Linn and Haase, Peter. 2023. *SemOpenAlex: The Scientific Landscape in 26 Billion RDF Triples*.

Jin, Qiao and Dhingra, Bhuwan and Liu, Zhengping and Cohen, William and Lu, Xinghua. 2019. *PubMedQA: A Dataset for Biomedical Research Question Answering*.

Rodney Kinney and Chloe Anastasiades and Russell Authur and Iz Beltagy and Jonathan Bragg and Alexandra Buraczynski and Isabel Cachola and Stefan Candra and Yoganand Chandrasekhar and Arman Cohan and Miles Crawford and Doug Downey and Jason Dunkelberger and Oren Etzioni and Rob Evans and Sergey Feldman and Joseph Gorney and David Graham and Fangzhou Hu and Regan Huff and Daniel King and Sebastian Kohlmeier and Bailey Kuehl and Michael Langan and Daniel Lin and Haokun Liu and Kyle Lo and Jaron Lochner and Kelsey MacMillan and Tyler Murray and Chris Newell and Smita Rao and Shaurya Rohatgi and Paul Sayre and Zejiang Shen and Amanpreet Singh and Luca Soldaini and Shivashankar Subramanian and Amber Tanaka and Alex D. Wade and Linda Wagner and Lucy Lu Wang and Chris Wilhelm and Caroline Wu and Jiangjiang Yang and Angele Zamarron and Madeleine Van Zuylen and Daniel S. Weld. 2023. *The Semantic Scholar Open Data Platform*.

Saier, Tarek and Krause, Johan and Färber, Michael. 2023. *unarXive 2022: All arXiv Publications Pre-Processed for NLP, Including Structured Full-Text and Citation Network*.

EarlySciRev: A Dataset of Early-Stage Scientific Revisions Extracted from LaTeX Writing Traces

Léane Jourdan[1], Julien Aubert-Béduchaud[1], Yannis Chupin[1], Marah Baccari[1] and Florian Boudin[2]
[1]Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France
[2]Inria, LS2N, Nantes Université, France
{firstname.lastname}@univ-nantes.fr

Abstract

Scientific writing is an iterative process that generates rich revision traces, yet publicly available resources typically expose only final or near-final versions of papers. This limits empirical study of revision behaviour and evaluation of large language models (LLMs) for scientific writing. We introduce EarlySciRev, a dataset of early-stage scientific text revisions automatically extracted from arXiv LaTeX source files. Our key observation is that commented-out text in LaTeX often preserves discarded or alternative formulations written by the authors themselves. By aligning commented segments with nearby final text, we extract paragraph-level candidate revision pairs and apply LLM-based filtering to retain genuine revisions. Starting from 1.28M candidate pairs, our pipeline yields 578k validated revision pairs, grounded in authentic early drafting traces. We additionally provide a human-annotated benchmark for revision detection. EarlySciRev complements existing resources focused on late-stage revisions or synthetic rewrites and supports research on scientific writing dynamics, revision modelling, and LLM-assisted editing.

Keywords: text revision, dataset, LLM filtering

1. Introduction

Academic writing is an inherently demanding task, requiring precision, clarity, and conciseness, often under time pressure and frequently in a second language. To support this process, researchers increasingly rely on LLMs, which can rewrite entire passages in response to high-level instructions. These models offer unprecedented assistance for revising drafts, improving readability, and articulating complex ideas with clarity and fluency.

However, assessing the impact of LLMs on scientific writing remains challenging. The key data required for such analysis, *drafts, revisions, and writing traces*, are largely inaccessible. While most scientific papers are publicly available in their final or near-final form, the iterative writing process that produced them typically remains hidden.

Because scientific writing is inherently incremental, authors progressively refine arguments, restructure paragraphs, clarify explanations, and adjust claims. This process naturally generates various intermediate artifacts, including discarded sentences, rewritten paragraphs, and commented alternatives. Yet these traces are rarely preserved in publicly accessible resources. As a result, existing research on revision modelling predominantly relies on late-stage revisions (e.g., between submission versions) or synthetic rewrites. Early drafting stages, where substantial conceptual and structural changes occur, remain underexplored.

The lack of access to early-stage revisions limits our ability to study authentic writing dynamics, measure quality improvements, train models that support in-depth revision, and evaluate the role of LLMs in scientific writing. It also hinders systematic investigation of issues such as stylistic homogenisation, bias propagation, or factual distortion introduced during automated rewriting.

This inaccessibility largely stems from ethical, legal, and ownership concerns: drafts may contain personal information, unpublished ideas, or confidential comments, and they often relate to papers that later fall under publisher copyright.

In this paper, we introduce EarlySciRev, a large-scale dataset of early-stage scientific revisions automatically extracted from arXiv LaTeX source files. Our key insight is that LaTeX comments often preserve discarded or intermediate versions of sentences and paragraphs. By mining these commented segments and aligning them with nearby final text, we recover fine-grained revision pairs that reflect authentic author rewriting. We present a complete pipeline[1] for (i) collecting computer science papers from arXiv, (ii) cleaning and processing LaTeX sources, (iii) extracting candidate revision pairs from commented text, and (iv) filtering genuine revisions using a LLM-based classification. Finally, we report an annotation study that benchmarks several LLMs on the revision detection task, and use the best model to curate the final dataset.

Our contributions are threefold:

- We propose a new method to retrieve early-

[1] https://github.com/JourdanL/EarlySciRev

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 119–126
12 May 2026.

stage scientific writing traces by exploiting commented content in LaTeX source files.

- We introduce EarlySciRev, a large-scale dataset of paragraph-level scientific revisions automatically extracted from arXiv papers, capturing authentic early drafting revisions.[2]
- We release a human-annotated benchmark for revision detection in scientific text, enabling systematic evaluation of both LLMs and future revision models.[2]

2. Related Work

A variety of datasets for text revision have been released over the years, reflecting growing interest in modelling writing and rewriting processes. Some of these datasets rely on synthetic revisions, generated either automatically (Ito et al., 2019) or manually by annotators who are not the original authors (Mita et al., 2024). While such resources are useful for controlled experimentation, they do not capture authentic authorial revision behaviour.

Among datasets that feature real author revisions, two primary sources have emerged: arXiv (Tan and Lee, 2014; Du et al., 2022; Jiang et al., 2022) and OpenReview (D'Arcy et al., 2023; Jourdan et al., 2024, 2025). These resources enable the study of revision behaviour in real-world scientific writing contexts.

Despite their usefulness, existing datasets have clear limitations. First, many are restricted in scope. Some focus exclusively on abstracts (Tan and Lee, 2014; Du et al., 2022), while others remain relatively small in scale, ranging from a few hundred (Jiang et al., 2022; Mita et al., 2024; Ruan et al., 2024) to a few thousand papers (Kuznetsov et al., 2022; D'Arcy et al., 2023; Dycke et al., 2023; Lin et al., 2023; Jourdan et al., 2025).

Second, most existing datasets primarily capture late-stage revisions, typically between near-final drafts posted on arXiv or submission platforms. These revisions often reflect already polished manuscripts prepared for public dissemination, rather than the exploratory and formative stages of writing. Early drafting phases, where substantial conceptual, structural, and stylistic changes are made, are therefore largely absent from current resources. To our knowledge, the only dataset that explicitly targets writing traces from the earliest stages of the drafting process is ScholaWrite (Wang et al., 2025). However, it is limited to only five papers, which restricts its applicability beyond exploratory analysis.

As a result, access to early-stage writing traces remains a major bottleneck for studying authentic scientific revision behaviour and for developing models that support in-depth revision. In this work, we hypothesize that such traces can be recovered directly from the LaTeX source files uploaded to arXiv. In practice, authors frequently leave commented-out sentences, paragraphs, or alternative phrasings in the source code (i.e. lines beginning with the "`%`" character). Mining these commented segments offers a promising and underexplored avenue for reconstructing fine-grained, early-stage scientific revisions.

3. Data Creation

This section describes the pipeline used to construct the dataset, from collecting raw data from arXiv to cleaning LaTeX sources and extracting candidate revision pairs.

3.1. Data Collection

We rely on the `arxiv-metadata-oai-snapshot.json`[3] dump downloaded on January 21, 2026. This metadata file contains 2,932,928 arXiv repositories, among which 596,118 are distributed under a permissive licence.[4]

In this work, we focus on computer science (CS) papers, which represent 286,747 with a valid licence. For each of these papers, we downloaded all available source archives (zip files), resulting in approximately 1.2 TB of source data.

3.2. LaTeX Source Processing

We first filter the source files to retain only LaTeX documents that contain potentially meaningful comments. Specifically, we select files that include commented lines that do *not* start with a backslash. This criterion excludes commented-out LaTeX commands, while preserving comments that may contain previous versions of sentences or paragraphs.

We then apply the following cleaning steps:

1. We retain only the content between `\begin{document}` and `\end{document}`.
2. We remove non-textual environments, including `table`, `figure`, `align`, `tikz`, and `algorithm`.
3. We replace each `equation` environment with a special token `[EQUATION]`, as equations may appear within running text.
4. We remove LaTeX commands that do not contain textual content (e.g., `\appendix`, `\vs-`

[2] https://huggingface.co/datasets/taln-ls2n/EarlySciRev

[3] https://www.kaggle.com/datasets/Cornell-University/arxiv

[4] CC BY-NC-SA 4.0, CC BY-SA 4.0, CC BY 4.0, Public Domain List, CC BY-NC-SA 3.0, CC BY 3.0, CC0 1.0

pace), while preserving structural commands such as \section.

These steps ensure that the resulting text primarily consists of natural language content suitable for revision analysis.

3.3. Candidate Revision Pair Extraction

Our objective is to extract candidate revision pairs composed of: i) a block of uncommented text that appears in the compiled document (the *final paragraph*), and (ii) a block of commented text that may correspond to an earlier version (the *commented paragraph*). We define a *block* as a sequence of consecutive lines of the same type (commented or uncommented), not interrupted by an empty line or by a line of the other type.

For each commented block c, we compare it to the five preceding and five following blocks. For each neighbouring block that corresponds to a final paragraph f, we compute a normalised difference ratio based on the Levenshtein distance between the two texts. To account for cases in which a revision affects only part of a paragraph, we compute this distance using a sliding window over the final paragraph. We define this normalised difference ratio as:

$$d_{norm}(f, c) = lev(f, c)/\max(|f|, |c|) \quad (1)$$

where $lev(\cdot, \cdot)$ denotes the Levenshtein distance and $|\cdot|$ the character length. We treat a pair as a candidate revision when $d_{norm} < 0.7$. This threshold was set empirically on a subset of the dataset.

For each final paragraph, we retain all associated candidate commented revisions. Before concatenating uncommented blocks, we remove inline trailing comments to ensure that the resulting text matches the content of the compiled document.

4. Data Annotation

The automatic extraction procedure described above result in 1,269,976 candidate revision pairs. However, not all of these candidates correspond to genuine rewriting instances. We therefore introduce an additional filtering step based on LLMs. To select an appropriate prompting strategy and model, we first construct a human-annotated gold subset.

4.1. Human Annotation

The objective of the annotation campaign is to determine whether a given pair of paragraphs constitutes a genuine revision instance. We randomly sampled 500 candidate pairs, each consisting of a *commented paragraph* (representing a potential original version) and a corresponding *final paragraph*. The annotation was carried out by five annotators: two master's students, two junior researchers, and one senior researcher. All annotators had prior experience with scientific writing and NLP research.

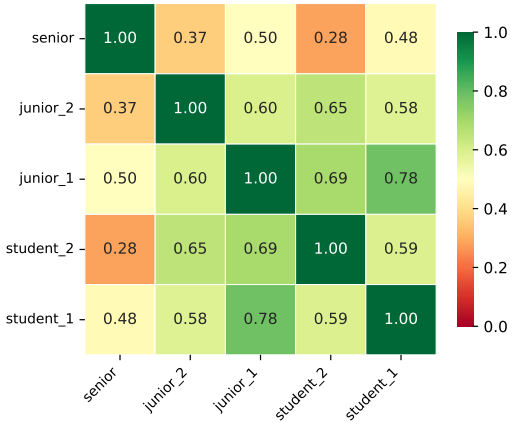


Figure 1: Pairwise Cohen's Kappa (κ_{Cohen}) scores between annotators.

For each paragraph pair, annotators answered the following binary question: *"Can the final paragraph be qualified as a revision of the original one(s)?"* Annotators were provided with detailed guidelines specifying decision criteria (see Appendix A). These guidelines were derived from the revision taxonomy proposed by Jourdan et al. (2025) and adapted to the present task.

Annotation was conducted using *Label Studio*. Paragraph pairs were displayed side by side. To facilitate comparison, identical text spans shared by the two paragraphs were highlighted (Appendix B).

Inter-annotator agreement. Figure 1 shows pairwise Cohen's κ_{Cohen} scores between annotators. Agreement varies across annotator pairs, with lower agreement observed for pairs involving the senior annotator.The senior annotator was more selective than other annotators on the pairs considered as revision.

To assess overall inter-annotator agreement, we apply Fleiss' κ_{Fleiss} under a partially overlapping design, in which each item is annotated by three annotators selected from the pool of five. The resulting $\kappa_{\text{Fleiss}} = 0.54$ indicates moderate agreement, according to the scale proposed by Landis and Koch (1977).

4.2. LLM-Based Filtering

Given the number of the extracted candidate pairs, manual annotation of the full dataset is infeasible. We therefore rely on automatic classification to identify genuine revision pairs. To select the most reliable approach, we evaluate several models and prompting strategies on the human-annotated subset. Since each item in this subset was labelled by

Instructions

```
You will receive two paragraphs P1
and P2. P2 is a final version of a
paragraph written for a scientific
article, and P1 is suspected to be an
 original version of P2 before
revision.
Can the P2 paragraph or a part of it
be qualified as a revision of P1?
If the changes only affect equations
or if here is too much equations it
is not considered a revision.
Answer with only one word "Yes" or "
No".
```

Figure 2: Prompt used in both *-PLUIE and LLM-choice settings.

three annotators, we use majority vote to derive the reference label for evaluation.

Approaches. We compare two LLM-based approaches: i) **Standard prompting**, where the model directly answers the binary question posed in the prompt, and the decision is extracted from the generated output. ii) ***-PLUIE** (Lemesle et al., 2026), a perplexity-based LLM-as-a-judge method that estimates the model's confidence without generating free-form text. In the *-PLUIE setting, the model estimates its confidence by computing the perplexity of candidate responses ("Yes" vs. "No") given the prompt. Positive values favour the revision hypothesis, negative values the opposite, and a binary decision is obtained by thresholding the score (default threshold 0, optimised a posteriori when annotations are available).

Models and setup. We evaluate several LLMs of varying sizes, including Qwen3 (4B,14B), phi-4 (14B), Olmo-3-7B-Instruct, and Llama-3.1-8B-Instruct. All models are run in a chat-completion setting using `bfloat16` precision. Sampling is disabled during inference (`do_sample=False`) to ensure deterministic outputs. The same prompt (Figure 2) is used across both standard prompting and *-PLUIE configurations, enabling a direct comparison between generation-based and perplexity-based classification for each model.

Results and model selection. Results are reported in Table 1. Across both approaches, Qwen3 (14B) achieves the best overall performance, with accuracy exceeding 80%. Balancing classification performance and computational efficiency, we select the *-PLUIE configuration with Qwen3 (14B) to filter the full dataset, using the optimal threshold determined on the annotated subset.

5. Dataset Statistics

Applying the selected LLM-based filtering strategy retain 578,440 revision pairs out of the initial 1,2M candidates (approximately 45.55%). These validated revisions correspond to 523,932 distinct final paragraphs. Among them, 46,192 paragraphs are associated with more than one revision candidate, reflecting cases where authors experimented with multiple alternative formulations before settling on a final version or cases where multiple previous paragraphs were merged into one. At the document level, the revisions are distributed across 104,023 articles.Those characteristics and more are summarised in Table 2.

Qualitative inspection suggests that revisions captured in EarlySciRev range from local fluency edits to more substantial restructuring and clarification of scientific arguments, to draft ideas left as to do with their fully written version.

The 500 paragraphs used for human annotation are also included and filtered at this step. Both the human annotated dataset and large LLM filtered one are openly available.[5]

	Model	Acc.	P.	R.	time
*-PLUIE	Qwen3 (4B) (thr=0)	0.77	0.77	0.76	22h
	Qwen3 (4B) (thr=-4.75)	0.78	0.75	0.81	
	Olmo-3 (7B) (thr=0)	0.74	0.73	0.76	35h
	Olmo-3 (7B) (thr=0.60)	0.74	0.78	0.67	
	Llama-3.1 (8B) (thr=0)	0.70	0.65	0.85	35h
	Llama-3.1 (8B) (thr=0.85)	0.73	0.75	0.67	
	phi-4 (14B) (thr=0)	0.77	0.71	0.92	55h
	phi-4 (14B) (thr=2.15)	0.79	0.75	0.85	
	Qwen3 (14B) (thr=0)	0.80	0.75	0.90	62h
	Qwen3 (14B) (thr=5.55)	**0.82**	0.80	0.86	
LLM-choice	Qwen3 (4B)	0.59	0.55	0.95	29h
	Llama-3.1 (8B)	0.59	0.55	**0.97**	46h
	Olmo-3 (7B)	0.68	**0.81**	0.45	45h
	phi-4 (14B)	0.78	0.80	0.75	81h
	Qwen3 (14B)	0.80	0.78	0.83	82h

Table 1: Alignment of LLM-based classifier to human majority vote. *Thr.* is the threshold used to binarise *-PLUIE values, *Acc.* the accuracy, *P.* the precision, *R.* the recall and *time* the estimated time to classify all the data. **Bold** values indicate the best results, and underlined values indicate the second-best results.

#rev	#paper	#rev/§	#words/§	% words diff
578,440	104,023	1.10	82.42	56.85

Table 2: Characteristics of EarlySciRev. In this order: number of revision pairs, number of articles, average number of commented paragraphs per final paragraph, average number of words per paragraph (final version), average percentage of difference in words per revision pair

[5] https://huggingface.co/datasets/taln-ls2n/EarlySciRev

6. Conclusion

We introduced EarlySciRev, a dataset of paragraph-level scientific revisions extracted from LaTeX writing traces in arXiv source files, together with a human-annotated benchmark for revision detection. By focusing on early drafting stages, this resource makes visible revision phenomena that are typically inaccessible in existing datasets.

EarlySciRev supports empirical study of scientific writing dynamics and provides a foundation for developing and evaluating revision models, including LLM-based systems. To facilitate reproducibility and further development, we release the full extraction and filtering framework, enabling updates as new papers become available.[6]

A future step could be to label all data with a revision intention and see how the distribution compare to dataset focusing on late stage revision.

7. Limitations

The current pipeline is restricted to computer science papers, as we only process licensed CS articles from arXiv. Writing practices and revision behaviours may differ across disciplines, limiting the generalizability of our findings. Extending the approach to other domains is a natural direction for future work.

Additionally, we do not explicitly control for the language of the paper. A part of papers submitted to arXiv are written in languages other than English. Our LLM-based filtering relies on prompts written in English, which may affect classification reliability for non-English texts. Adapting prompts to the language of each document or incorporating language identification into the pipeline could improve robustness.

Acknowledgements

This work was partly supported the AID-CNRS NaviTerm project (convention 2022 65 0079 CNRS Occitanie Ouest).

8. Bibliographical References

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2023. Aries: A corpus of scientific paper edits made in response to peer reviews.

Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pages 96–108, Dublin, Ireland. Association for Computational Linguistics.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023. NLPeer: A unified resource for the computational study of peer review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5049–5073, Toronto, Canada. Association for Computational Linguistics.

Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Léane Jourdan, Florian Boudin, Richard Dufour, Nicolas Hernandez, and Akiko Aizawa. 2025. ParaRev : Building a dataset for scientific paragraph revision annotated with revision instruction. In *Proceedings of the First Workshop on Writing Aids at the Crossroads of AI, Cognitive Science and NLP (WRAICOGS 2025)*, pages 35–44, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Léane Jourdan, Florian Boudin, Nicolas Hernandez, and Richard Dufour. 2024. CASIMIR: A corpus of scientific articles enhanced with multiple author-integrated revisions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2883–2892, Torino, Italia. ELRA and ICCL.

Ilia Kuznetsov, Jan Buchmann, Max Eichler, and Iryna Gurevych. 2022. Revise and resubmit: An intertextual model of text-based collaboration in peer review. *Computational Linguistics*, 48(4):949–986.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

[6] https://github.com/JourdanL/EarlySciRev

Quentin Lemesle, Léane Jourdan, Daisy Munson, Pierre Alain, Jonathan Chevelu, Arnaud Delhay, and Damien Lolive. 2026. *-pluie: Personalisable metric with llm used for improved evaluation.

Jialiang Lin, Jiaxin Song, Zhangping Zhou, Yidong Chen, and Xiaodong Shi. 2023. MOPRD: A multidisciplinary open peer review dataset. *Neural Computing and Applications*, 35(34):24191–24206.

Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. Towards automated document revision: Grammatical error correction, fluency edits, and beyond. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.

Qian Ruan, Ilia Kuznetsov, and Iryna Gurevych. 2024. Re3: A holistic framework and dataset for modeling collaborative document revision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4635–4655, Bangkok, Thailand. Association for Computational Linguistics.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.

Linghe Wang, Minhwa Lee, Ross Volkov, Luan Tuyen Chau, and Dongyeop Kang. 2025. Scholawrite: A dataset of end-to-end scholarly writing process.

A. Annotation Guidelines

Annotation Guidelines for Revision Detection

1. Introduction
The goal of this annotation campaign is to detect text revisions amid paragraphs originating from computer science scientific papers. A paragraph level revision is defined as a paragraph that is substantially modified for clarity, simplicity, style and other aspects. To that end, some final paragraphs have been selected and each one of them was provided with one or more original paragraphs that were under comment in the latex file. A final paragraph is a paragraph that is not commented and is suspected to be a revision of an original paragraph(s). In this task, we aim to characterize the final paragraphs' relationship with the suspected original paragraph(s), so that they can be classified as revisions or not down the line.

2. Annotation Task
Annotators are presented with a pair of paragraphs: an original version composed of one or several paragraphs and a final version. Their task is to answer the following question: *Can the final paragraph be qualified as a revision of the original one(s)?*
Annotators must select one of the following labels:
- **YES**: The final paragraph constitutes a revision of the original paragraph.
- **NO**: The final paragraph does not constitute a revision (e.g., different scientific content, the idea developed is not the same, introduces too much new information, or does not change the text).

As several original candidates are proposed, the annotator can answer Yes for multiple paragraphs (e.g. in cases of paragraph merging or iterative revision).

2.1 Positive example

Original Paragraph	Final paragraph
Therefore, the generalization rapidly decreases after augmentationinterrupted when training with a single background because the learning direction toward generalization about various backgrounds is not helpful to train. On the other hand, the training can have helpwhen their difculty is solved by augmentation, such as Figure 2(b) and Figure 2(c).	Therefore, the generalization rapidly decreases after augmentation is interrupted during training with a single background because the learning direction toward generalization about various backgrounds is not helpful to train. In contrast, the training can help when their difficulty is solved by augmentation (Figure 2(b), 2(c)).

2.2 Negative example

Original Paragraph	Final paragraph
In future research, the multi-mode characteristics will be studied to improve the representativeness of degradation features and the trendability of HI, and transfer learning approaches will be investigated to improve the generalization ability of the proposed framework and extend it to different systems.	Based on the ablation study, it can be concluded that the proposed SkipAE, inner HI-prediction block, and the HI-generating module jointly improve the ability of HI for reliable and accurate prognostics.

2.3 Annotation Procedure
For each pair of paragraphs (original and final), annotators must proceed as follows:
1. Read the final paragraph carefully to understand its scientific content and intent.
2. Read the original paragraph to identify any differences with respect to the final version.
3. Assess whether each original is rephrased in the final paragraph considering aspects such as: grammatical correctness, clarity and readability, fluency and coherence, appropriateness of scientific style.
4. Determine whether the scientific meaning of the paragraph is preserved in the final version.
5. Assign a label (YES or NO) according to the decision rules defined below.

Annotators should base their decision solely on the information contained in the paragraph pair and should not rely on external context. Also, annotators are prohibited to invent things.

2.4 Decision Rules
Annotators must apply the following rules when assigning labels:
Assign YES if at least one of the following conditions are met for a part or the whole paragraph:
1. The final paragraph is a revised version of the original paragraph, incorporating changes ranging from minor edits to substantial rephrasing.
2. The final version has been modified through the addition, the substitution or the deletion of ideas or facts.
3. The revision expands on the same idea with additional or withdrawn details.
4. The differences between the original and final paragraphs indicate the correction of document processing errors (e.g., parsing issues, segmentation errors, or misaligned paragraphs).

Assign NO if:
1. None of the above conditions are met.
2. If the annotator is unsure whether the revised paragraph constitutes a valid paragraph-level revision.
3. If there are only equations or code.
4. If the two paragraphs are the exact same.

If presented with multiple commented paragraphs for the same final paragraph, one or more commented paragraph can independently be considered as a revision. Classifying a commented paragraph as a revision does not disqualify the other proposed candidates. The same goes with the negative label : all the commented paragraphs may not qualify as a revision.

Table 3: Detailed annotation guidelines provided to the annotators for the revision detection task.

B. Annotation Examples

Reference	Candidate	Revised
We observed that AR2VP demonstrates superior entity perception outcomes, achieving the highest overall perception performance. This analysis underscores that current V2X technologies rarely rely on RSUs to expand perception horizons. In contrast, AR2VP harnesses the latent strengths of RSUs to address intra-scene changes, which enhances the vehicle's ability to adapt to dynamic scenes, consequently elevating the overall perception capabilities. However, AR2VP does exhibit a performance drawback in pedestrian detection, implying a particular challenge in detecting small targets.	We find that AR2VP shows superior entity perception results, with overall perception performance being the best. This analysis suggests that existing V2X technologies merely utilize RSU to extend perception horizons. In contrast, AR2VP leverages the latent advantages of RSU to model intra-scene changes, further enhancing vehicle adaptability to scene dynamics, thereby augmenting overall perception capabilities. However, in comparison to V2V, AR2VP exhibits a performance disadvantage in pedestrian detection, indicating a certain discrepancy in detecting small targets.	yes
Under \cref{ass:runtimecomplexity} and following \cref{tab:setops}, the outer approximative Minkowski sum from \cref{prop:minkSum_polyzono}, the Minkowski difference, and the linear map in the computation of the outer approximation $\outerBRSAE{-t}$ are all $\bigO{n^3}$, while the computation of the inner approximation $\innerBRSAE{-t}$ is dominated by the conversion to a constrained zonotope, which is $\bigO{n^4}$.	Under \cref{ass:runtimecomplexity}, the computation of the outer approximation $\outerBRSAE{-t}$ is marginally dominated by the over-approximative Minkowski sum, which is $\bigO{(\cons{}+2n)n, since the Minkowski difference and linear map are at most $\bigO{(\cons{}+2n)n \steps{}n}$ and $\bigO{(\cons{}+2n)n^2}$, respectively; all these operations are essentially $\bigO{n^3}$ for $n \gg \steps{}$ and under \cref{ass:runtimecomplexity}.	yes
$$\sum_{t=1}^T \sum_{y \in \{0, 1\}} p_t^y \hat{\ell}_t(y) - \inf_{j \in [N]}\sum_{t=1}^T \hat{\ell}_t(\mathcal{E}_t^{j}) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \hat{\ell}_t(1) + \eta\sum_{t=1}^T p_t^1(1 - p_t^1) \hat{\ell}_t(0)^2 + \eta\sum_{t=1}^T p_t^1 \hat{\ell}_t(1)^2.$$ \end{lemma}	$$\sum_{t=1}^T \sum_{y \in \{0, 1\}} p_t^y \hat{\ell}_t(y) - \sum_{t=1}^T \sum_{y \in \{0, 1\}} \hat{\ell}_t(\mathcal{E}_t^{j}) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \hat{\ell}_t(1) + \eta\sum_{t=1}^T \sum_{y \in \{0, 1\}} p_t^y \hat{\ell}_t(y)^2.$$	no
\iuhead{Date of fault-triggering test creation and modification:} We identified all the commits that are associated with the fault-triggering tests and analyzed when the commits happened (e.g., before or after the bug was reported/fixed). We used the git command "\inlineCode{git log -L:[funcname]:[file]}" to identify the list of commits that modified the fault-triggering test and the modification date.	We collected the date and time information provided in the output of this command to track the development activities associated with each fault-triggering test. This allowed us to better understand how the fault was identified and resolved over time with respect to the changes in fault-triggering tests. \peter{maybe remove this, I don't quite get this detail}Note that if a test is inherited from a parent class, we perform our analysis directly on the parent class since any changes would only occur in that class.	no

Table 4: Sample of annotated paragraph pairs, identical chain of texts are in green.

Enhancing Factuality and Transparency in Generative Models for Biomedical Question Answering

Ankita Behura, Siting Liang, Daniel Sonntag
German Research Center for Artificial Intelligence, Oldenburg University
Germany
{siting.liang, daniel.sonntag}@dfki.de
{siting.liang, daniel.sonntag}@uni-oldenburg.de

Abstract

Biomedical Question Answering (BQA) systems are vital for providing clinicians and researchers with efficient access to large amount of biomedical scientific studies. Existing automated BQA models, however, often rely on complex hybrid architectures to handle diverse question and answer formats, leading to inefficiency and high complexity. While domain-specific generative language models like BioBART offer a unified and simplified alternative capable of producing fluent human-like responses, they are prone to hallucination and lack interpretability, undermining their trustworthiness in critical healthcare domains. To address these limitations, this work introduces an enhanced model that augments BioBART with a pointer network for accurate token copying and a novel Keyphrase Filter (KPF) to guide attention toward critical information during generation. Experimental results on the BioASQ challenge demonstrate that the proposed Pointer-KPF model significantly outperforms the baseline BioBART, particularly on metrics for ideal answers. Furthermore, our evaluation shows that the model enhances transparency: pointer-guided attention heatmaps reveal improved input-output alignment, while keyphrase scores act as saliency maps to identify the most influential input segments. This approach not only reduces hallucination by strengthening textual grounding but also provides crucial insights into the model's reasoning, thereby increasing confidence and trust in its outputs.

Keywords: Biomedical Question Answering, Biomedical Language Model, Unified Question Answering

1. Introduction

Biomedical Natural Language Processing (BioNLP) plays a crucial role in improving information retrieval and processing within the biomedical domain by integrating expertise from bioinformatics, medicine, and artificial intelligence. Over the years, NLP techniques have been applied to classification, information extraction, question answering (QA), and drug discovery (Cohen and Hersh, 2005; Cohen and Demner-Fushman, 2014). Early approaches relied on semantic parsing technologies for identifying relations between disease (Mkrtchyan and Sonntag, 2014), rule-based techniques for negation handling, such as NegEx (Profitlich and Sonntag, 2021). Although ontology-based approaches (Sonntag et al., 2009, 2016) enhance biomedical entity recognition by incorporating structured knowledge, they suffer from high annotation costs and limited adaptability to evolving biomedical literature.

Traditional Biomedical Question Answering (BQA) systems have relied on information retrieval (IR) techniques (Voorhees, 2001; Niu et al., 2003), which, despite retrieving relevant text spans, struggle with complex medical queries, domain-specific terminology, and the need for precise, interpretable answers (Schmidt et al., 2016). To advance BQA research, the BioASQ challenge (Nentidis et al., 2023) provides expert-curated biomedical question answering datasets for model evaluation. As shown in Table 1, the questions cover a range of types, including **"yes/no"**, **"factoid"**, **"list"**, and **"summary"** questions. Answers are categorized as either **"exact"** (derived directly from the provided text) or **"ideal"** (human-like summaries of the relevant information).

Question Type	Example Question, Exact Answer, and Ideal Answer
Yes/No	**Question:** Proteomic analyses need prior knowledge of the organism's complete genome. Is the complete genome of the bacteria of the genus *Arthrobacter* available? **Exact Answer:** yes **Ideal Answer:** Yes, the complete genome sequence of *Arthrobacter* (two strains) is deposited in GenBank.
List	**Question:** List Hemolytic Uremic Syndrome Triad. **Exact Answer:** [anaemia, thrombocytopenia, renal failure] **Ideal Answer:** Hemolytic uremic syndrome (HUS) is a clinical syndrome characterized by the triad of anaemia, thrombocytopenia, renal failure.
Factoid	**Question:** What enzyme is inhibited by Opicapone? **Exact Answer:** [catechol-O-methyltransferase] **Ideal Answer:** Opicapone is a novel catechol-O-methyltransferase (COMT) inhibitor to be used as adjunctive therapy in levodopa-treated patients with Parkinson's disease.
Summary	**Question:** What kind of affinity purification would you use to isolate soluble lysosomal proteins? **Ideal Answer:** The rationale for purification of the soluble lysosomal proteins resides in their characteristic sugar, the mannose-6-phosphate (M6P), which allows an easy purification by affinity chromatography on immobilized M6P receptors.

Table 1: Examples of BioASQ Question Types with Exact and Ideal Answers for four question types. Exact answers (highlighted in red) are concise, fact-based responses directly extracted from input texts, typically named entities. Ideal answer generation in BQA aims to provide comprehensive, well-structured responses synthesizing relevant implicit knowledge.

Recent approaches leverage transformer-based models like BERT (Devlin et al., 2019) and its biomedical adaptations, such as BioBERT (Lee et al., 2020), and BioGPT (Luo et al., 2022), achieving strong performance on BQA tasks (Jin et al., 2022; Hu et al., 2023; Kim et al., 2023). However,

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 127–139
12 May 2026.

these models require separate fine-tuning for different question types, increasing training costs and system complexity. Additionally, scalability and interpretability remain significant challenges, limiting their clinical applicability (Jin et al., 2022; Lyu et al., 2024).

In this work, we propose a unified BQA system based on a domain-specific BioBART (Lewis et al., 2019; Yuan et al., 2022) model with multi-task learning to handle diverse question types within a Sequence to Sequence (Seq2Seq) architecture. Specifically, we integrate a pointer network (Vinyals et al., 2015; See et al., 2017) and a key phrase filtering mechanism into the Seq2Seq architecture. This improves the factual accuracy of generated answers by dynamically emphasizing relevant information during training and inference. Unlike existing key phrase labeling techniques (Liang et al., 2022), our approach enables end-to-end learning without requiring explicit key phrase annotations, making our system more efficient. Our experimental results demonstrate that incorporating a pointer network and key-phrase filtering improves both the precision of answers and the transparency of the BioBART model, thereby addressing critical challenges in BQA. Our goal is to foster trust and usability of pre-trained generative language models in clinical applications. We will publish our code and fine-tuned models.

2. Related Work

PLMs for Biomedical NLP. The scarcity of annotated biomedical data limits the performance of general-purpose language models in this specialized domain. Models like BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019), ClinicalBERT (Alsentzer et al., 2019), and PubMedBERT (Gu et al., 2021) have demonstrated superior performance in biomedical NLP tasks by effectively capturing domain-specific semantics. Models like BioELMo (Jin et al., 2019), and Bio-ELECTRA (Ozyurt, 2020) integrate domain-specific ontologies and knowledge graphs to enhance their understanding of biomedical concepts. Within the BioASQ challenge (Nentidis et al., 2022), PLMs-based extractive approaches have been widely adopted, with models such as BioM-ELECTRA (Alrowili and Vijay-Shanker, 2021), BioM-ALBERT (Alrowili, 2021), and Bio-ELECTRA (Ozyurt, 2020) achieving strong performance by treating BQA as a span prediction task (Jin et al., 2022). Fine-tuning on QA datasets has further enhanced these models, allowing them to learn task-specific answer extraction patterns (Nentidis et al., 2023).

Generative models for summary generation. Unlike extractive approaches that label text spans, generative models synthesize information from input sources to produce coherent, human-like responses. Encoder-decoder frameworks have become a standard approach for generating concise yet informative summaries across domains (Nallapati et al., 2017; Chopra et al., 2016). Transformer-based architectures such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) have been widely used in abstractive summarization tasks. BioBART (Yuan et al., 2022) adapts the BART architecture to the biomedical domain. BioGPT (Luo et al., 2022) is a decoder-only generative model designed for biomedical text generation. To ensure factual consistency with the source text, hybrid approaches combining extractive and abstractive techniques have been proposed (Gu et al., 2016; Kryściński et al., 2018). See et al. (2017) and Nallapati et al. (2016) integrated a pointer network (Vinyals et al., 2015) into an RNN-based encoder-decoder model, allowing the system to either generate words from a predefined vocabulary or copy words directly from the source text. Building on a BERT-based encoder-decoder model, Liang et al. (2022) integrated a pointer mechanism to enable direct copying of text spans from the source input, combined with span extraction supervision to enhance source text reconstruction during generation. Prompting large language models (Hsueh et al., 2023) has been explored to improve model performance. However, this approach requires key phrase annotations as generation targets during training. Despite these advancements, PLM-based methods face challenges in providing explainable answers, motivating further research into hybrid approaches for accountable biomedical QA.

3. Method

3.1. BioBART Baseline

We utilize BioBART (Yuan et al., 2022) as our baseline, a variant of BART (Lewis et al., 2020) pre-trained on biomedical corpora, including PubMed abstracts. This domain-specific pre-training enables BioBART to handle the complexities of biomedical vocabulary and concepts. BioBART employs a standard transformer encoder-decoder architecture with bidirectional encoder self-attention and causal decoder self-attention.

3.2. Pointer Network

To enhance BioBART's ability to leverage both generative and extractive capabilities for BQA, we integrated a pointer network as shown in Figure 1. The pointer network introduces an additional mechanism that adjusts the cross-attention scores so that the model can either copy words from the input

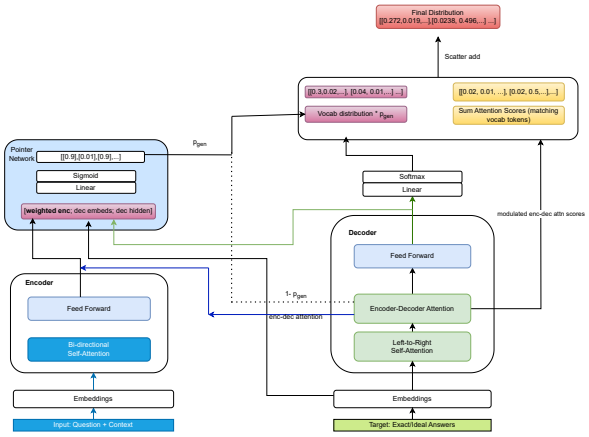


Figure 1: The pointer network runs in parallel to the standard BART generation process. It takes the decoder hidden state, the current input token embedding, and the weighted encoder hidden states as input. It produces a generation probability, p_{gen}, which is used directly to modulate the cross-attention scores. These modulated cross-attention scores are then summed to match the vocabulary tokens. The final output distribution is produced by combining the vocabulary distribution with these summed attention scores.

(close attention weight on source tokens) or generate new words (standard decoding). Specifically, the model dynamically decides between generating a token from the vocabulary or copying one from the input, using a learned probability distribution at each decoding step. At each decoding step t, the pointer network computes a generation probability p_{gen}, calculated as:

$$p_{\text{gen}} = \sigma(\mathbf{w}_{\text{ptr}}^T[h_t^x; y_t; s_t] + b_{\text{ptr}}) \tag{1}$$

where σ is the sigmoid function, $\mathbf{w}_{\text{ptr}}$ is a learnable weight vector, b_{ptr} is a bias term, $\mathbf{h}_{x_t}$ is the encoder hidden state, y_t is the current decoder input token embedding, and s_t is the decoder's hidden state. These hidden states from the final encoder are used as context vectors passed to each decoding step. Each decoder state s_t predicting the next token is also from the last decoder.

The attention-weighted encoder representation h_t^x is computed using the encoder-decoder cross-attention. Since the model uses multi-head attention, we compute the mean over all attention heads as suggested in Liang et al. (2022). Let $\alpha_{t,j}^{(i)}$ denote the attention weight from decoder step t to encoder token j for head i. The averaged attention is:

$$\bar{\alpha}_{t,j} = \frac{1}{N_{\text{heads}}} \sum_{i=1}^{N_{\text{heads}}} \alpha_{t,j}^{(i)} \tag{2}$$

The attended encoder representation is then:

$$h_t^x = \sum_{j=1}^{T_x} \bar{\alpha}_{t,j} \cdot h_j^x \tag{3}$$

where j indexes the source tokens and T_x is the input sequence length.

The pointer network modulates the cross-attention scores by scaling them with $(1 - p_{\text{gen}})$. The final output token distribution $P_{\text{final}}(w)$ is then computed as:

$$P_{\text{final}}(w) = p_{\text{gen}} \cdot P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \cdot \sum_{j:x_j=w} \bar{\alpha}_{t,j} \tag{4}$$

where $P_{\text{vocab}}(w)$ is the vocabulary distribution, and the second term aggregates attention scores over all source positions j where the input token x_j matches the vocabulary token w. The benefit of incorporating a pointer network is to improve the alignment between the generated output and the input context, reducing factual inconsistencies.

3.3. Keyphrase Filter

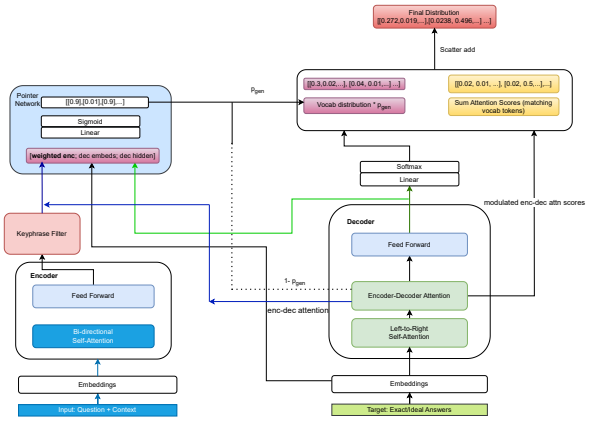


Figure 2: The KPF module is implemented as a linear layer that predicts keyphrase scores for each input token using the encoder hidden states. These scores are used to weight the encoder hidden states, which are then employed in the cross-attention mechanism and as input to the pointer network. The cross-attention weights are also weighted further by the keyphrase predictions.

To ensure that the model focuses on crucial information in the input text, we introduce a Keyphrase Filter (KPF) between the encoder and decoder. See Figure 2 for an architectural overview of the Pointer-KPF network. The KPF is implemented as a linear layer (binary classifier) trained jointly with the model. It predicts keyphrase scores based on encoder hidden states and is trained implicitly to identify important keyphrases without explicit annotations.

For each encoder token position j, the KPF predicts a score k_j using:

$$e_j = \text{KPF}(h_j^x) \tag{5}$$

$$k_j = \sigma(e_j) \tag{6}$$

where σ is the sigmoid function.

The keyphrase scores are used to modulate the cross-attention weights before aggregation. Specifically, the attention weights are re-weighted as:

$$\tilde{\alpha}_{t,j} = k_j \cdot \bar{\alpha}_{t,j} \tag{7}$$

The modified attended encoder representation is then:

$$h_t^{x'} = \sum_{j=1}^{T_x} \tilde{\alpha}_{t,j} \cdot h_j^x \tag{8}$$

These modified attention scores are also incorporated into the pointer mechanism. The final output distribution becomes:

$$P_{\text{final}}(w) = p_{\text{gen}} \cdot P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \cdot \sum_{j:x_j=w} k_j \cdot \bar{\alpha}_{t,j} \tag{9}$$

The KPF assigns the saliency of each token in the input sequence, which is then used to modulate both the encoder hidden states and the cross-attention scores in the decoder. This mechanism guides the model to prioritize relevant keyphrases while reducing the influence of less informative tokens, ultimately strengthening its extractive capabilities and improving the accuracy of generated answers. While the pointer network ensures that factual accuracy is preserved by copying key information from the input, the KPF reinforces this process by explicitly guiding attention toward keyphrases in the input text.

4. Experiments

We evaluated our models using the 11th release of BioASQ dataset (Nentidis et al., 2023). The BioASQ dataset provides expert-curated training questions, relevant abstracts, snippets, and correct answers. Table 2 presents the final distribution of question types across training, validation, and test sets.

4.1. Data Pre-processing

We adopted the "Snippet as is" pre-processing strategy, which utilized individual-provided snippets as contexts for each question, without further concatenation or processing. This approach was selected due to its balanced performance across all question types (Yes/No, List, Factoid, and Summary), as demonstrated by Yoon et al. (2019). This strategy offers data augmentation, focused attention on crucial terms, enhanced semantic nuance understanding, and promotes answer diversity. The preprocessing extracts a single snippet per question, preserving its original structure across the dataset.

Split		Yes/No	List	Factoid	Summary
Train		25,010	13,817	26,685	12,997
Val		20,007	10,878	17,790	8,665
	Set 1	319	432	591	326
	Set 2	269	210	660	655
Test	Set 3	378	310	400	341
	Set 4	327	150	414	318
	Set 5	596	499	527	293

Table 2: Data Statistics for Train, Validation, and Test Sets based on the four question types.

4.2. Task-Specific Token

Adapting pretrained language models (PLMs) to downstream tasks often involves addressing variations in data structure and desired output formats. While post-processing techniques are common, they can be tedious and may not generalize well. To address the diverse question types and answer formats, we implemented a task-specific token strategy. Inspired by prior work (Ateia and Kruschwitz, 2023; Achiam et al., 2023), we prepend task-specific tokens to the input sequence, explicitly informing the model about the question type and expected answer format, see Table 3. This enables the model to learn distinct answer patterns associated with specific question types during training.

Answer Format	Question Type	Task-Specific Token
Exact	Factoid	`<factoid_exact_answer>`
	List	`<list_exact_answer>`
	Yes/No	`<yesno_exact_answer>`
Ideal	Factoid	`<factoid_ideal_answer>`
	List	`<list_ideal_answer>`
	Yes/No	`<yesno_ideal_answer>`
	Summary	`<summary_ideal_answer>`

Table 3: Task-specific tokens used for different question types and answer formats.

4.3. Training and Inference

We evaluated BioBART, BioBART with a pointer network (Pointer-Only), BioBART with a pointer network and keyphrase filter (Pointer-KPF), and removing the pointer network from the Pointer-KPF model

(KPF-Only). All models were initialized with the pre-trained BioBART[1] model (Yuan et al., 2022). We train the models on a single NVIDIA A100-SXM4-80GB GPU, with a batch size of 16, a maximum input length of 512, a maximum of 3 training epochs, and with a learning rate of 2×10^{-5}.

4.4. Evaluation Metrics

To evaluate the robustness and consistency of our models, we examined their performance across the five test sets. For each model, question type, and test set, we generated multiple answers while systematically varying the beam size from 1 to 5. The BioASQ challenge evaluates exact answers differently based on the question type. Set N as the total number of questions.

Strict Accuracy (SAcc) SAcc is the metric used for factoid questions. It measures the proportion of questions for which the correct answer is predicted and placed at the top of the ranked list of answers:

$$SAcc = \frac{c_1}{N} \tag{10}$$

where c_1 is the number of questions answered correctly when only the top answer is considered.

Lenient Accuracy (LAcc) LAcc is a metric that increases when the correct answer is predicted and placed anywhere within the top k answers:

$$LAcc = \frac{c_k}{N} \tag{11}$$

where c_k is the number of questions answered correctly when all top k answers are considered.

Mean Reciprocal Rank (MRR) MRR is a metric that evaluates the effectiveness of a system in ranking correct answers. It increases inversely with the position of the correct answer in the ranked list:

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i} \tag{12}$$

where $rank_i$ is the position of the correct answer for the i-th question. In the BioASQ challenge, the top 5 (k = 5) answers are predicted for each question, and MRR is calculated based on these predictions.

Precision, Recall, and F1-Measure For list questions, the evaluation is based on precision (P), recall (R), and F1-measure ($F1$). These metrics are averaged across all questions to compute the Mean Average Precision (MAP), Mean Average Recall (MAR), and Mean Average F1-measure (MAF). The formulas for precision, recall, and F1-measure are:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{15}$$

where TP is the number of correct answers predicted, FP is the number of incorrect answers predicted, FN is the number of correct answers not predicted.

ROUGE Metrics ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) measures the overlap between system-generated and reference (golden) ideal answers. The metric applied in our evaluation is:

- ROUGE-1: Evaluates unigram overlap between the generated and reference answers.

5. Results

5.1. Results of Exact Answer Generation

Table 4 presents the exact answer generation performance of each model across the test sets. The mean reflects the average performance across the generations with varying beam sizes, while the standard deviation indicates the variability in performance across these generations. A smaller standard deviation signifies more stable performance across different beam sizes.

Unlike Factoid and List questions, all models perform well in Yes/No questions, with the Baseline model already achieving accuracy (>69%). Pointer-KPF and Pointer-Only demonstrate the benefits of pointer networks, but KPF-Only surprisingly performs best in some test batches, in particular for Factoid and Yes/No questions, suggesting a complex interaction between pointer networks and keyphrase filtering. Pointer-Only consistently outperforms other models in MAP and MAF for List questions, demonstrating the effectiveness of Pointer Networks for list-type answers. Pointer-KPF exhibits a substantial performance boost, achieving up to a 50x increase in MAP (test set 2, from 0.005 to 0.202). Keyphrase filtering also contributes, but its impact is less consistent.

5.2. Results of Ideal Answer Generation

For each question type (yes/no, factoid, list, summary), systems are expected to provide an "ideal" answer resembling a concise response that a

[1] GanjinZero/biobart-v2-base

Test Set	Model	Factoid			List			Yes/No	
		LAcc	MRR	SAcc	MAP	MAR	MAF	Acc	MAF
1	Baseline	0.200 ± 0.013	0.160 ± 0.027	0.135 ± 0.033	0.077 ± 0.011	0.089 ± 0.013	0.082 ± 0.012	0.696 ± 0.000	0.582 ± 0.000
	Pointer-Only	0.294 ± 0.021	0.264 ± 0.023	0.235 ± 0.021	0.410 ± 0.006	0.480 ± 0.009	0.411 ± 0.007	$\mathbf{0.774 \pm 0.020}$	$\mathbf{0.633 \pm 0.049}$
	Pointer-KPF	$\mathbf{0.318 \pm 0.025}$	$\mathbf{0.296 \pm 0.016}$	$\mathbf{0.282 \pm 0.016}$	$\mathbf{0.445 \pm 0.019}$	$\mathbf{0.479 \pm 0.023}$	$\mathbf{0.446 \pm 0.018}$	0.739 ± 0.000	0.425 ± 0.000
	KPF-Only	0.276 ± 0.039	0.257 ± 0.040	0.247 ± 0.040	0.357 ± 0.033	0.442 ± 0.052	0.366 ± 0.035	0.739 ± 0.000	0.546 ± 0.000
2	Baseline	0.288 ± 0.025	0.263 ± 0.026	0.241 ± 0.032	0.005 ± 0.007	0.007 ± 0.007	0.006 ± 0.007	0.778 ± 0.000	0.723 ± 0.000
	Pointer-Only	0.459 ± 0.034	0.382 ± 0.026	0.335 ± 0.033	$\mathbf{0.260 \pm 0.018}$	$\mathbf{0.191 \pm 0.007}$	$\mathbf{0.207 \pm 0.012}$	0.767 ± 0.025	0.669 ± 0.022
	Pointer-KPF	0.453 ± 0.034	0.379 ± 0.035	0.341 ± 0.033	0.202 ± 0.023	0.144 ± 0.010	0.157 ± 0.014	0.744 ± 0.031	0.606 ± 0.067
	KPF-Only	$\mathbf{0.465 \pm 0.025}$	$\mathbf{0.394 \pm 0.023}$	$\mathbf{0.353 \pm 0.029}$	0.205 ± 0.006	0.174 ± 0.004	0.179 ± 0.004	$\mathbf{0.833 \pm 0.000}$	$\mathbf{0.778 \pm 0.000}$
3	Baseline	0.238 ± 0.035	0.176 ± 0.035	0.150 ± 0.035	0.061 ± 0.003	0.058 ± 0.002	0.059 ± 0.002	0.792 ± 0.000	0.705 ± 0.000
	Pointer-Only	$\mathbf{0.325 \pm 0.036}$	$\mathbf{0.268 \pm 0.042}$	$\mathbf{0.244 \pm 0.041}$	$\mathbf{0.225 \pm 0.007}$	0.213 ± 0.006	$\mathbf{0.214 \pm 0.006}$	$\mathbf{0.816 \pm 0.037}$	$\mathbf{0.747 \pm 0.070}$
	Pointer-KPF	0.275 ± 0.074	0.214 ± 0.059	0.194 ± 0.051	0.138 ± 0.008	0.129 ± 0.009	0.130 ± 0.008	0.725 ± 0.023	0.561 ± 0.055
	KPF-Only	0.287 ± 0.034	0.237 ± 0.034	0.219 ± 0.038	0.218 ± 0.026	$\mathbf{0.221 \pm 0.010}$	0.214 ± 0.018	0.792 ± 0.000	0.705 ± 0.000
4	Baseline	0.310 ± 0.029	0.274 ± 0.027	0.252 ± 0.027	0.091 ± 0.020	0.083 ± 0.014	0.083 ± 0.016	0.792 ± 0.000	0.736 ± 0.000
	Pointer-Only	0.374 ± 0.018	0.335 ± 0.017	0.310 ± 0.018	0.247 ± 0.010	0.210 ± 0.016	0.219 ± 0.012	0.875 ± 0.000	0.845 ± 0.006
	Pointer-KPF	0.361 ± 0.027	0.303 ± 0.037	0.265 ± 0.058	$\mathbf{0.290 \pm 0.011}$	$\mathbf{0.252 \pm 0.007}$	$\mathbf{0.264 \pm 0.008}$	0.900 ± 0.023	0.870 ± 0.026
	KPF-Only	$\mathbf{0.413 \pm 0.014}$	$\mathbf{0.381 \pm 0.015}$	$\mathbf{0.361 \pm 0.027}$	0.264 ± 0.013	0.226 ± 0.004	0.237 ± 0.007	$\mathbf{0.917 \pm 0.000}$	$\mathbf{0.889 \pm 0.000}$
5	Baseline	0.172 ± 0.024	0.133 ± 0.028	0.103 ± 0.035	0.138 ± 0.007	0.137 ± 0.006	0.137 ± 0.006	0.714 ± 0.000	0.689 ± 0.000
	Pointer-Only	$\mathbf{0.214 \pm 0.015}$	0.165 ± 0.012	0.131 ± 0.016	$\mathbf{0.380 \pm 0.004}$	$\mathbf{0.386 \pm 0.003}$	$\mathbf{0.378 \pm 0.003}$	0.786 ± 0.000	0.775 ± 0.000
	Pointer-KPF	0.200 ± 0.016	0.165 ± 0.005	$\mathbf{0.138 \pm 0.000}$	0.342 ± 0.020	0.337 ± 0.018	0.337 ± 0.020	0.743 ± 0.039	0.714 ± 0.056
	KPF-Only	0.207 ± 0.000	$\mathbf{0.167 \pm 0.000}$	0.138 ± 0.000	0.341 ± 0.017	0.347 ± 0.019	0.340 ± 0.017	$\mathbf{0.786 \pm 0.000}$	$\mathbf{0.775 \pm 0.000}$

Table 4: Performance metrics for different models for exact answer generation. Values are reported as Mean ± Std across varying beam sizes.

Test Set	Model	Factoid	List	Yes/No	Summary
1	Baseline	0.361 ± 0.008	0.163 ± 0.023	0.457 ± 0.039	0.414 ± 0.020
	Pointer-only	$\mathbf{0.383 \pm 0.031}$	0.360 ± 0.017	0.476 ± 0.027	0.473 ± 0.036
	Pointer-KPF	0.378 ± 0.029	$\mathbf{0.441 \pm 0.056}$	$\mathbf{0.477 \pm 0.011}$	$\mathbf{0.509 \pm 0.032}$
	KPF-Only	0.331 ± 0.023	0.401 ± 0.067	0.459 ± 0.070	0.429 ± 0.043
2	Baseline	0.362 ± 0.018	0.098 ± 0.020	0.445 ± 0.029	0.416 ± 0.011
	Pointer-only	0.390 ± 0.011	0.289 ± 0.031	0.445 ± 0.040	0.447 ± 0.043
	Pointer-KPF	$\mathbf{0.409 \pm 0.004}$	$\mathbf{0.313 \pm 0.022}$	$\mathbf{0.455 \pm 0.017}$	$\mathbf{0.473 \pm 0.061}$
	KPF-Only	0.378 ± 0.028	0.3 ± 0.027	0.428 ± 0.057	0.392 ± 0.050
3	Baseline	0.449 ± 0.028	0.113 ± 0.009	0.393 ± 0.008	0.400 ± 0.017
	Pointer-only	0.457 ± 0.009	0.273 ± 0.007	0.405 ± 0.014	0.435 ± 0.003
	Pointer-KPF	$\mathbf{0.467 \pm 0.016}$	$\mathbf{0.299 \pm 0.043}$	0.406 ± 0.013	$\mathbf{0.457 \pm 0.020}$
	KPF-Only	0.444 ± 0.053	0.283 ± 0.037	$\mathbf{0.455 \pm 0.060}$	0.443 ± 0.014
4	Baseline	0.361 ± 0.028	0.077 ± 0.006	0.356 ± 0.017	0.359 ± 0.011
	Pointer-only	0.377 ± 0.010	0.271 ± 0.008	0.395 ± 0.043	$\mathbf{0.462 \pm 0.034}$
	Pointer-KPF	$\mathbf{0.412 \pm 0.013}$	$\mathbf{0.337 \pm 0.064}$	0.391 ± 0.028	0.452 ± 0.032
	KPF-Only	0.388 ± 0.041	0.264 ± 0.014	$\mathbf{0.416 \pm 0.058}$	0.384 ± 0.009
5	Baseline	0.403 ± 0.008	0.145 ± 0.010	0.354 ± 0.043	0.367 ± 0.020
	Pointer-only	0.393 ± 0.011	0.353 ± 0.003	0.381 ± 0.010	0.377 ± 0.010
	Pointer-KPF	$\mathbf{0.432 \pm 0.052}$	0.350 ± 0.009	$\mathbf{0.430 \pm 0.008}$	$\mathbf{0.429 \pm 0.020}$
	KPF-Only	0.376 ± 0.034	$\mathbf{0.357 \pm 0.024}$	0.418 ± 0.059	0.354 ± 0.021

Table 5: ROUGE-1 F1-scores (Mean ± Std) of varying beam sizes. Bold values indicate the best mean score within each batch.

biomedical expert might provide. Table 5 presents ROUGE-1 F1-scores for ideal answer generation across different question types (Factoid, List, Yes/No, and Summary) and all test sets.

For factoid questions, performance is generally similar across models, though the Pointer-KPF model tends to perform best. This suggests that the combination of the pointer network and keyphrase filtering yields the best results for this question type. Pointer-Only consistently shows improvement over both Baseline and KPF-Only models, indicating that the pointer network plays a crucial role in enhancing factoid answer generation.

For list questions, Pointer-KPF frequently achieved the best ROUGE-1 scores, leveraging both keyphrase filtering and the pointer network. Notably, it showed a substantial improvement over the Baseline.

For yes/no questions, performance differences were minimal, suggesting BioBART's inherent effectiveness for these simpler questions. For factoid questions, Pointer-KPF generally achieved the highest scores, demonstrating the complementary benefits of pointer networks and keyphrase filtering. Pointer-Only consistently outperformed the Baseline, highlighting the importance of the pointer network. However, in Test Set 1, the Baseline slightly surpassed Pointer-Only, underscoring dataset-specific performance variations.

For summary questions, Pointer-KPF consistently demonstrated the strongest performance, reinforcing trends observed in other question types. Pointer-Only also outperformed the Baseline, further highlighting the pointer network's role in improving long-form answer generation.

Our analysis of ideal answer generation revealed several key findings:

- Synergistic Effect of Pointer Network and Keyphrase Filtering: Pointer-KPF, which combines both mechanisms, generally achieved the strongest performance across factoid, list, and summary questions, demonstrating the benefits of this combined approach.

- Effectiveness in List Questions: Significant performance improvements were observed for list questions, particularly with Pointer-KPF, highlighting the effectiveness of both pointer networks and keyphrase filtering for this question type.

- While Pointer-KPF often achieves the highest scores, the performance across the three model settings is relatively close. In some test sets, KPF-Only even surpasses Pointer-Only, albeit by a small margin.

5.3. Impact of beam size on Answer Generation

To further investigate the impact of decoding parameters on Pointer-KPF model performance, we conducted beam size experiments, varying the beam size from 1 to 5. This allowed us to explore the sensitivity of our models to the breadth of the search space during answer generation for both exact and ideal answers. The results show that the impact of beam size on exact answer generation is minimal (Figure 3). Performance remains relatively stable across different beam sizes, with only slight fluctuations in metrics like factoid lenient accuracy and list MAP/F1. This consistent trend suggests that the model's ability to identify exact answer spans is not significantly affected by increasing the beam size. The model appears to identify relevant answer spans with relatively small beam sizes. One possible explanation is that the search space for exact matches is inherently smaller and more focused than that for ideal matches, making the beam size less influential.

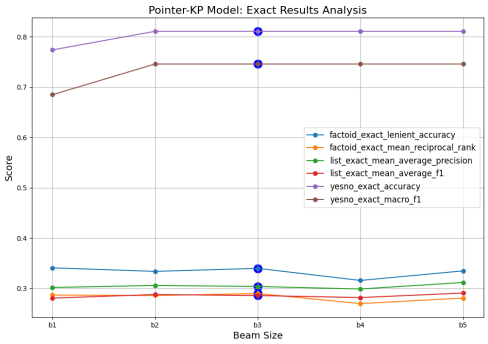


Figure 3: Pointer-KPF - Exact Answers

For ideal answer generation, the impact of beam size is more pronounced (Figure 4). For instance, the Pointer-KPF model's list ROUGE-1 improves from 0.297 at beam size 1 to 0.340 at beam size 3. This suggests that exploring a wider range of paraphrased answers and keyphrase combinations with larger beam sizes is particularly beneficial for generating better ideal answers, especially for list questions. This is likely because the search space for ideal matches is significantly larger and more diverse, requiring a broader search to identify optimal combinations of paraphrased text and keyphrases.

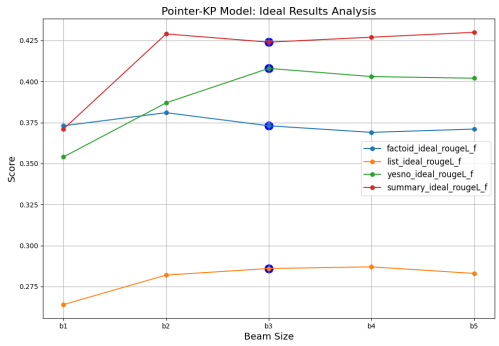


Figure 4: Pointer-KPF - Ideal Answers

6. Interpretability

Question	Which drugs are included in the CABENUVA pill?
Context	A regimen comprising extended release injectable suspensions of cabotegravir and rilpivirine for concurrent administration (CABENUVA™) is being developed by ViiV Healthcare and Janssen Pharmaceutica (Janssen) as a complete regimen for HIV infection.
Ideal Answer Reference	CABENUVA contains cabotegravir and rilpivirine. It is used for treatment of HIV.
Baseline	CABENUVA pill includes travaginal progesterone. It is approved for use in HIV-infected patients in the USA and Canada.
Pointer	CABNERUVA includes cabotegravir, rilpivirine, and tmburt, for the concurrent treatment of HIV infection.
Pointer-KPF	CABENUVA includes cabotegravir and rilpivirine for concurrent administration to HIV-infected patients.

Table 6: Comparison of generated ideal answers of different model settings for a list-type question. Text highlighted in red indicates a factual error or hallucination.

To understand performance improvements, we analyzed the model generations and attention patterns for a list-type question. Firstly, Table 6 presents the generations of different model settings and the reference, illustrating the progressive improvement from the Baseline BioBART model to the Pointer-KPF model. The Baseline model fails, hallucinating an unrelated drug ("travaginal progesterone"). The Pointer model shows improved factual grounding, correctly identifying the two primary drugs, but introduces new errors, including a misspelling of the product name ("CABNERUVA") and an invented component ("tmburt"). The Pointer-

KPF model generates the most accurate and faithful response, correctly stating the drug components.

Baseline model generated an incorrect answer, relying on generic vocabulary, as evidenced by its diffused cross-attention shown in Figure 6 (in the Appendix). The Pointer model improved, identifying correct components, but hallucinated an irrelevant term, displaying broader attention (see Figure 7). Conversely, the Pointer-KPF model generated accurate answers, demonstrating the KPF's ability to focus on relevant keyphrases, confirmed by their concentrated attention (Figure 8).

However, while cross-attention heatmaps provide useful insights, they have limitations: they may not fully capture model reasoning due to averaging across layers and attention heads, they carry a risk of misinterpreting high attention as meaningful contribution, and they are not intuitive for non-ML users. To address these issues and improve interpretability, particularly regarding keyphrases, we utilize saliency maps, specifically keyphrase importance scores derived from the KPF module (Figure 5). These scores highlight input tokens receiving the most attention, providing a more human-readable interpretation than cross-attention alone. The KPF scores, as seen in Figure 5, demonstrate the model's focus on relevant input parts, with higher saliency indicating more important keyphrases. Compared to heatmaps, these scores offer a clearer and more intuitive representation of model behavior, enabling more direct tracing of how specific keyphrases contribute to the final prediction. This visualization offers a more accessible way to understand the model's decision-making process, especially for non-expert users.

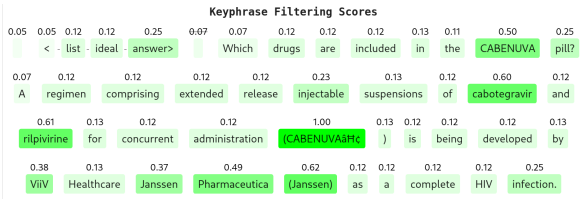


Figure 5: Keyphrase saliency scores overlaid on the input sequence for the list-type question. Color gradient indicates keyphrase saliency.

6.1. Comparison with BioASQ Leaderboard Systems

The BioASQ challenge remains the definitive benchmark for evaluating semantic question-answering capabilities in the biomedical domain. By examining the performance of the Pointer-KPF model relative to the official results from BioASQ 11b (2023), 12b (2024), and the most recent 13b (2025–2026) batches, the improvements of the proposed architectural enhancements can be contextualized within the broader state-of-the-art. Table 7 compares our self-interpretable Pointer-KPF model with recent large language model (LLM)-based approaches across multiple BioASQ editions. LLM systems such as IISR-1 based on (Hsueh et al., 2023) and UR-IW-1 using Mistral-7B, as well as fine-tuned and prompt-based variants (Kim et al., 2025), consistently achieve near-perfect Yes/No accuracy and strong performance on factoid and list questions, benefiting from their large parameter scales (7B–1.8T). Similarly, domain-specific models like BioLinkBERT (Kim et al., 2023) demonstrate competitive results with fewer parameters (340M), highlighting the advantage of biomedical pretraining. Frontier LLMs and heavily ensembled systems demonstrate superior absolute scores on the BioASQ 11b and 12b leaderboards. Unlike these LLM-based systems, which function largely as black boxes, Pointer-KPF, extending the previous research work, introduces an explicit, self-interpretable keyphrase filtering mechanism. This enables transparent reasoning by highlighting task-relevant keyphrases, offering a clear trade-off between performance and interpretability. For instance, our model's averaged Factoid MRR of 0.303 is comparable to results seen in few-shot prompting of much larger models, such as the Mistral-7B 10-shot system (MRR 0.346) in the 13b challenge. Crucially, the Pointer-KPF achieves this performance with a BioBART-base backbone (140M parameters), which is significantly more efficient than the 7B–70B parameter models currently dominating the leaderboard. The results highlight an important direction for future research: bridging the gap between interpretability and performance.

System / Model Category	Edition	Yes/No Acc.	Factoid MRR	List F1	Params
IISR-1 (GPT-4)	11b (2023)	1.0000	0.4211	0.7043	∼1.8T
DMIS-KU-5 (BioLinkBERT)	11b (2023)	0.9583	0.6053	0.7219	340M
UR-IW-1 (Mistral-7B)	12b (2024)	1.0000	0.3254	0.4266	7.3B
Mistral-7B (Fine-tuned)	12b (2024)	0.9600	0.2381	0.5265	7.3B
2025-DMIS-KU-2	13b (2025)	1.0000	0.5321	0.5322	∼7B
Mistral-7B-Ins (10-shot)	13b (2025)	1.0000	0.3462	0.2374	7.3B
Pointer-KPF (this study)	11b (2023)	0.779*	0.303*	0.254*	140M

Table 7: Performance comparison across BioASQ editions. *Pointer-KPF results are averaged over five BioASQ 11b test sets.

7. Conclusion

In this work, we aim to demonstrate that a single, generative model could effectively handle diverse question types and answer formats, overcoming the limitations of complex, multi-model architectures. To achieve this, we introduce two key enhance-

ments: a pointer network that selectively copies tokens from the input context, modulating cross-attention scores to bridge abstractive summarization and precise information extraction, and a KPF module that predicts keyphrase scores to guide attention towards salient information, refining the model's ability to identify and utilize crucial context.

For exact answers, we observe substantial performance gains, particularly in list-type questions, with a consistent 20% increase in Mean Average F1 (MAF) scores using the pointer network. Factoid questions also highlight the importance of pointer mechanisms, with Pointer-KPF and Pointer-Only models demonstrating superior extractive capabilities. In ideal answer generation, the Pointer-KPF model consistently outperforms the baseline model, showcasing enhanced information extraction and paraphrasing abilities. Furthermore, to enhance interpretability and user confidence, we apply keyphrase saliency maps, providing user-friendly visualizations of the model's reasoning. In summary, our work demonstrates the effectiveness of integrating pointer networks and KPF into BioBART for enhanced BQA, highlighting the potential of refining generative models in biomedical information extraction.

While Pointer-KPF operates with a significantly smaller parameter size (140M) and does not yet match the raw performance of LLM-based systems, it establishes a strong foundation for interpretable QA. Future work will explore further refinements of the KPF module and evaluate performance on a broader range of biomedical tasks and datasets. Further improvements are needed to enhance answer accuracy while preserving the model's ability to provide clear, faithful explanations, potentially through better integration with pretrained knowledge, more robust keyphrase extraction, or hybrid approaches that combine interpretability with the strengths of large-scale models.

Limitations

While our research demonstrated promising results, several limitations warrant further investigation. The potential inadequacy of ROUGE scores in capturing semantic correctness and the sometimes misleading nature of attention visualizations suggest a need for alternative evaluation and analysis methods. To address these limitations, future work should focus on developing interactive keyphrase interfaces to enhance user trust, conducting user studies to assess keyphrase interactivity, implementing interactive mechanisms for keyphrase manipulation, exploring advanced attention analysis and decoding strategies, and leveraging extended context utilization with advanced data augmentation techniques. Ultimately, these efforts aim to enhance model performance, improve interpretability, and foster a more robust and reliable biomedical question answering system.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant numbers 01IW24006 (NoIDLEChatGPT), as well as by the Endowed Chair of Applied AI at the University of Oldenburg.

Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sultan Alrowili. 2021. Large biomedical question answering models with albert and electra.

Sultan Alrowili and K Vijay-Shanker. 2021. Biomtransformers: building large biomedical language models with bert, albert and electra. In *Proceedings of the 20th workshop on biomedical language processing*, pages 221–227.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Samy Ateia and Udo Kruschwitz. 2023. Is chatgpt a biomedical expert. *Exploring the Zero-Shot Performance of Current GPT Models in Biomedical Tasks*.

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

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

Aaron M Cohen and William R Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71.

Kevin Bretonnel Cohen and Dina Demner-Fushman. 2014. *Biomedical natural language processing*, volume 11. John Benjamins Publishing Company.

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.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Chun-Yu Hsueh, Yu Zhang, Yu-Wei Lu, Jen-Chieh Han, Wilailack Meesawad, and Richard Tzong-Han Tsai. 2023. Ncu-iisr: Prompt engineering on gpt-4 to stove biological problems in bioasq 11b phase b. In *11th BioASQ Workshop at the 14th Conference and Labs of the Evaluation Forum (CLEF)*.

Zhongjian Hu, Peng Yang, Bing Li, Yuankang Sun, and Biao Yang. 2023. Biomedical extractive question answering based on dynamic routing and answer voting. *Information Processing & Management*, 60(4):103367.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 82–89.

Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Huaiyuan Ying, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. 2022. Biomedical question answering: a survey of approaches and challenges. *ACM Computing Surveys (CSUR)*, 55(2):1–36.

Hajung Kim, Hoonick Lee, Yewon Cho, Jungwoo Park, Jueon Park, Soyon Park, Yan Ting Chok, Seungheun Baek, Donghyeon Lee, and Jaewoo Kang. 2025. Prompting matters: snippet-aware strategies for biomedical qa with llms in bioasq 13b. In *CLEF*.

Hyunjae Kim, Hyeon Hwang, Chaeeun Lee, Minju Seo, Wonjin Yoon, and Jaewoo Kang. 2023. Exploring approaches to answer biomedical questions: From pre-processing to gpt-4.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Siting Liang, Klaus Kades, Matthias Fink, Peter Full, Tim Weber, Jens Kleesiek, Michael Strube, and Klaus Maier-Hein. 2022. Fine-tuning bert models for summarizing german radiology findings. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 30–40.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

Daoming Lyu, Xingbo Wang, Yong Chen, and Fei Wang. 2024. Language model and its interpretability in biomedicine: A scoping review. *Iscience*.

Tigran Mkrtchyan and Daniel Sonntag. 2014. Deep parsing at the clef2014 ie task (dfki-medical). In *CEUR Workshop Proceedings*, volume 1180, pages 138–146.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016. Classify or select: Neural architectures for extractive document summarization. *CoRR*, abs/1611.04244.

Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, Salvador Lima López, Eulália Farré-Maduell, Luis Gasco, Martin Krallinger, and Georgios Paliouras. 2023. Overview of bioasq 2023: The eleventh bioasq challenge on large-scale biomedical semantic indexing and question answering. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 227–250. Springer.

Anastasios Nentidis, Georgios Katsimpras, Eirini Vandorou, Anastasia Krithara, Antonio Miranda-Escalada, Luis Gasco, Martin Krallinger, and Georgios Paliouras. 2022. Overview of bioasq 2022: The tenth bioasq challenge on large-scale biomedical semantic indexing and question answering. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 337–361. Springer.

Yun Niu, Graeme Hirst, Gregory McArthur, and Patricia Rodriguez-Gianolli. 2003. Answering clinical questions with role identification. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, pages 73–80.

Ibrahim Burak Ozyurt. 2020. On the effectiveness of small, discriminatively pre-trained language representation models for biomedical text mining. *bioRxiv*, pages 2020–05.

Hans-Jürgen Profitlich and Daniel Sonntag. 2021. A case study on pros and cons of regular expression detection and dependency parsing for negation extraction from german medical documents. technical report. *arXiv preprint arXiv:2105.09702*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Danilo Schmidt, Hans-Jürgen Profitlich, and Daniel Sonntag. 2016. Towards integrated information extraction and facetted search applications in nephrology. In *EMSA-RMed@ ESWC*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Daniel Sonntag, Volker Tresp, Sonja Zillner, Alexander Cavallaro, Matthias Hammon, André Reis, Peter A Fasching, Martin Sedlmayr, Thomas Ganslandt, Hans-Ulrich Prokosch, et al. 2016. The clinical data intelligence project: a smart data initiative. *Informatik-Spektrum*, 39:290–300.

Daniel Sonntag, Pinar Wennerberg, Paul Buitelaar, and Sonja Zillner. 2009. Pillars of ontology treatment in the medical domain. *Journal of Cases on Information Technology (JCIT)*, 11(4):47–73.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Ellen M Voorhees. 2001. The trec question answering track. *Natural Language Engineering*, 7(4):361–378.

Wonjin Yoon, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong, and Jaewoo Kang. 2019. Pre-trained language model for biomedical question answering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 727–740. Springer.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022. Biobart: Pretraining and evaluation of a biomedical generative language model. *arXiv preprint arXiv:2204.03905*.

Appendix

In the Appendix, Figures 6, 7, and 8 provide a comparative visualization of cross-attention patterns across the three models. These attention weights represent the strength of interaction between a specific input and output token. It is indicated by the intensity of the cell in which they meet. Higher intensity values correspond to stronger attention, revealing which input elements influence the model's predictions the most at each decoding step. By examining these patterns, we can gain insight into the model's reasoning process, identify whether it focuses on relevant information.

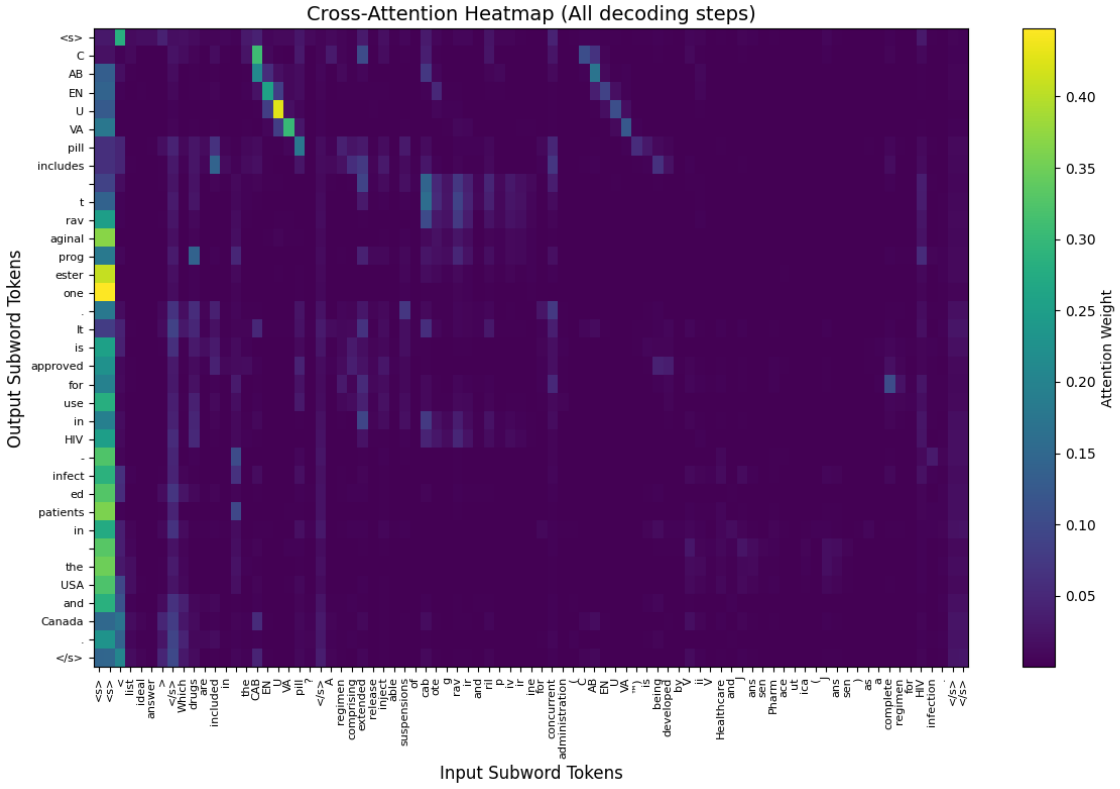


Figure 6: Baseline Cross-Attention Heatmap.

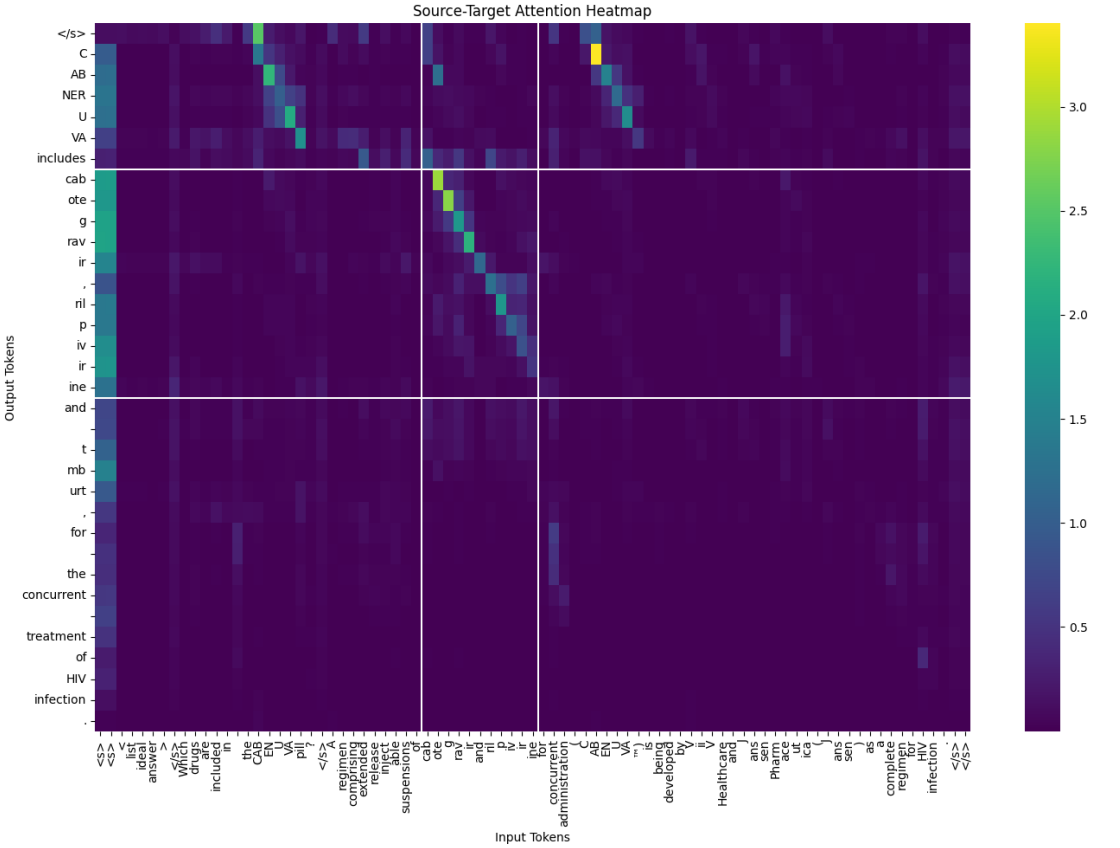


Figure 7: Cross-attention heatmaps for Pointer-only generation.

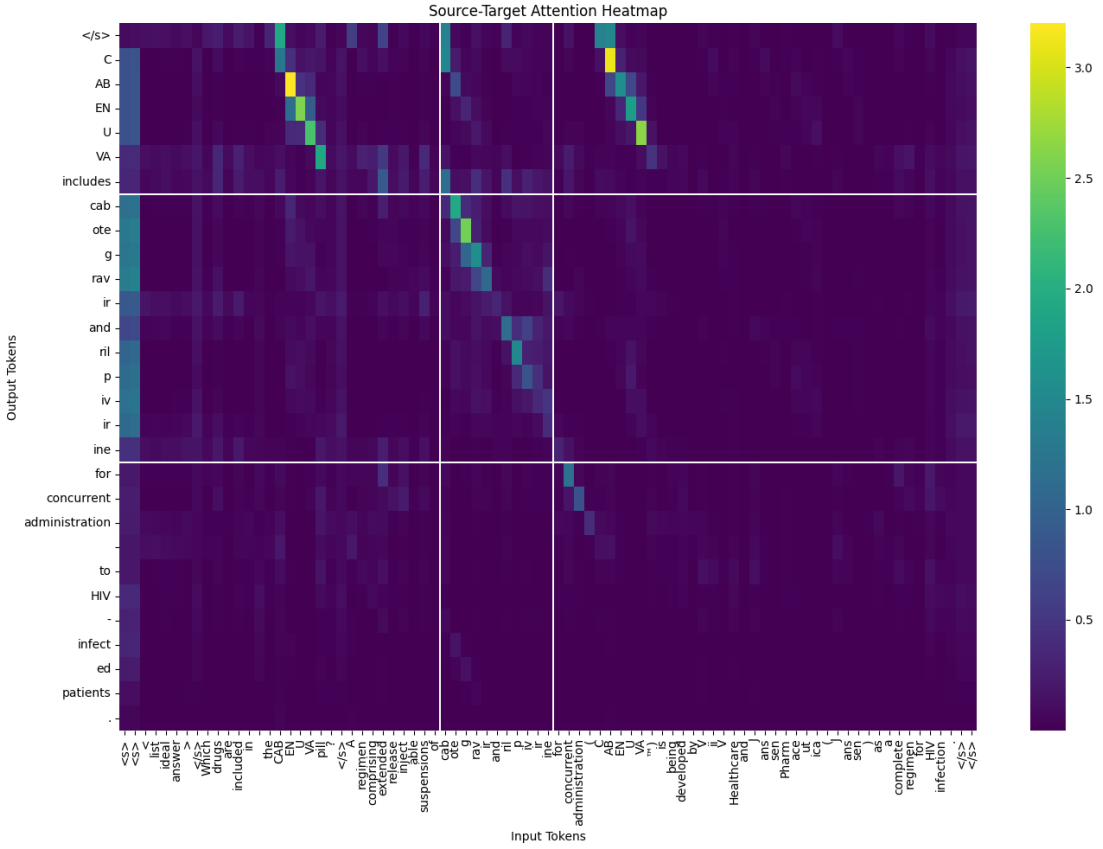


Figure 8: Cross-attention heatmaps for Pointer + KPF generation.

Enhancing Scholarly Knowledge Graphs via Domain-Specific Entity Detection and Linking

Nicolau Duran-Silva[1,2], **César Parra-Rojas**[1], **Pablo Accuosto**[1], **Julian Moreno-Schneider**[3], **Georg Rehm**[3]
[1]SIRIS Lab, Research Division of SIRIS Academic, Barcelona, Spain
[2]LaSTUS Lab, TALN Group, Universitat Pompeu Fabra, Barcelona, Spain
[3] Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Berlin, Germany
{nicolau.duransilva, cesar.parra, pablo.accuosto}@sirisacademic.com,
{julian.moreno_schneider, georg.rehm}@dfki.de

Abstract

Navigating scholarly content presents important challenges due to the fragmented and heterogeneous nature of research production and outputs. Scholarly Knowledge Graphs offer an efficient means to integrate diverse data sources and consolidate knowledge across outputs in a structured manner. This representation, combined with the grounding of unstructured textual data to well-defined research-related concepts, has great potential for enhancing knowledge discovery and supporting researchers navigating through vast amounts of scientific information. Knowledge extraction capabilities are commonly limited by the availability of large collections of annotated data supporting named-entity recognition (NER) and linking (EL), and the enormous effort that their elaboration entails for domain experts. Recent advances in natural language processing and generative artificial intelligence provide valuable opportunities to reduce the data annotation toll and produce high-quality NER with minimal expert involvement. Here, we present a pipeline for domain-specific NER and EL, leveraging LLMs and knowledge from experts in a human-in-the-loop approach to streamline the annotation process, along with transformer-based models and few-shot techniques. While the application focuses on showcasing four specific domains, the pipeline is designed to be flexible and domain agnostic for scientific fields.

Keywords: Named Entity Recognition, Entity Linking, Scholarly Document Processing

1. Introduction

Scholarly knowledge is inherently heterogeneous in nature, and the effective integration of its components has become increasingly critical in the current data-intensive research landscape (Manghi, 2024). Research outputs can take different forms—e.g., publications, datasets, software—and be scattered across diverse databases and repositories (Manghi et al., 2019; Peroni and Shotton, 2020; Priem et al., 2022; Hendricks et al., 2020; Wilkinson, 2010; Neylon et al., 2026). In addition to this, each research community has its own terminology and standards, and cataloguing and curation practices usually do not translate to other fields (Friedman et al., 2002). Even within a single field, the conventions adopted by different researchers may be inconsistent and scientific knowledge be often fragmented, unstructured and/or duplicated across outputs under different names, hindering data-informed decision-making by research-funding and research-performing organisations, as well as individual researchers themselves.

In recent years, scholarly knowledge graphs (SKGs) (Jaradeh et al., 2019; Manghi et al., 2019; Priem et al., 2022; Peroni and Shotton, 2020) have emerged as a core infrastructure for metadata-driven discovery. By integrating general and domain-specific knowledge from diverse data sources into interconnected entities and relationships, and linking concepts across different types of research outputs, SKGs transform fragmented and heterogeneous information into a structured and flexible representation that supports knowledge discovery, facilitating the uncovering of hidden patterns and connections between research-related entities (Vergoulis et al., 2019).

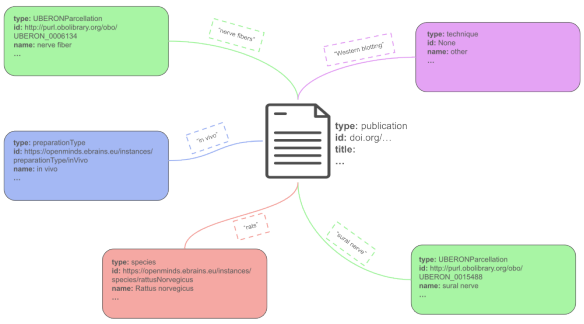


Figure 1: Example of enriching SKG in neuroscience domain with entity detection and linking.

Enriching domain-specific SKGs with relevant entities has the potential to streamline the research process (Leipzig et al., 2021; Wu et al., 2026), as researchers can uncover novel associations arising from the identification and linking of men-

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 140–154
12 May 2026.

tions of these entities across different works and/or databases, which can in turn inform potential avenues for further investigation (Taslimi et al., 2025; Liu et al., 2024). This information can originate from curated sources, with some fields boasting a plethora of highly developed platforms and resources for commonly-encountered entities—e.g., diseases, genes, or chemicals in the case of biomedical research—either in the sense of being added as metadata at the moment of creation of the research output or to be employed as training data for automated extraction pipelines via named entity recognition (NER) (Hong et al., 2020; Seow et al., 2025).

Due to the diversity and heterogeneity of research domains, however, these "canonical" or standard collections of domain-relevant terms may not necessarily align with the specific needs of a given subfield. For other fields that lack an extensive range of curated sources, there may be limited or non-existent training data for the identification of relevant entities (Yakimovich et al., 2021; Xu et al., 2023). In this regard, the advent of large language models presents valuable opportunities to enhance the scope of SKGs and increase entity coverage—thus grounding the unstructured text within the graphs in precise concepts—by lowering the barrier for enrichment through high-quality NER and entity linking, achievable with minimal task-specific training, in combination with the ability to generate domain- and context-aware synthetic examples (Shlyk et al., 2024; Zhang et al., 2025b; Xin et al., 2025). This, in turn, lowers curation friction, enabling researchers to navigate domain entities that were previously infeasible due to lack of labelled data.

We posit that high-recall, high-precision named-entity recognition tuned to specific scientific verticals is the keystone for unlocking richer SKGs. To this end, we introduce a flexible pipeline that (i) automatically LLM-generated preliminary annotations (pseudo-labels) for a dataset of full-text article sections sourced from open repositories, (ii) involves domain experts for pseudo-label validation, and (iii) fine-tunes transformer-based NER models for the five pilot domains of the EU-funded SciLake project —Neuroscience, Energy, Maritime Transport, and Connected Cooperative Autonomous Mobility (CCAM). The extracted entities can then be seamlessly fed back into interoperable SKGs. Due to the specific needs and idiosyncrasies of the target domains, the effort is focused towards creating a solution that can accommodate them all individually, in order to provide a tailored experience while being as domain-agnostic as possible. This ensures the enrichment pipeline can be adapted to new use cases in the future with minimal effort (Vergoulis et al., 2019).

We release code, models, and dataset under an open license to support research and further work on domain-adaptive NER and EL for scientific content[1].

2. Related Work

Concept recognition, commonly referred to as Named Entity Recognition (NER) and Entity Linking (EL), is a fundamental task in scholarly document processing and is essential for structuring scientific knowledge and enabling downstream applications such as knowledge graph construction, semantic search, and information extraction (Li and Zhang, 2023; Shlyk et al., 2024). Scientific text presents particular challenges due to specialised terminology, long-tail entities, and frequent ambiguity (Zhu et al., 2023). In domains such as neuroscience, transport, and energy research, substantial domain knowledge is embedded in complex and highly specialised concepts, making accurate identification and grounding of entities critical for connecting unstructured text with structured knowledge bases (Mihalcea and Csomai, 2007; Wu et al., 2020a; Xin et al., 2025). A persistent challenge for scientific NER and EL is the scarcity of labelled training data: supervised approaches typically require large annotated corpora, yet annotation is costly and time-consuming, as it demands domain expertise, and model performance is strongly correlated with the amount of available labelled data (Zhu and Jiang, 2021; Yakimovich et al., 2021).

To address domain mismatch, a range of domain-adaptive pre-trained language models have been proposed (Beltagy et al., 2019; Gururangan et al., 2020; Hong et al., 2022), which show that pre-training on scientific corpora substantially improves downstream performance while further emphasising the benefits of targeted pre-training, the latter explicitly leveraging citation links. Other approaches integrate structured or external knowledge sources, including document relations and ontologies, as in SPECTER (Cohan et al., 2020), KeBioLM (Yuan et al., 2021), and DiseaseBERT (He et al., 2020). Despite these advances, adapting such models to fine-grained, domain-specific entity recognition remains challenging, particularly in low-resource settings (Zhu and Jiang, 2021).

Transformer-based architectures dominate NER in scientific text (Seow et al., 2025), supported by scientific datasets, mainly from the biomedical domain, such as GENIA (Kim et al., 2011), BC4CDR (Li et al., 2016), AIONER (Luo et al., 2023) or SciERC (Luan et al., 2018). To reduce reliance on large labelled corpora, label-efficient approaches such as GLiNER (Zaratiana et al., 2024)

[1] Codes and datasets available at https://github.com/sirisacademic/meloner

formulate NER as span classification conditioned on natural-language label descriptions, enabling zero-shot and few-shot recognition, while GLiNER-BioMed (Yazdani et al., 2025) adapts this paradigm to biomedical text. Although these methods improve flexibility and reduce annotation costs, performance remains sensitive to label definitions and contextual coverage, particularly for highly specialised or domain-specific concepts.

Entity Linking (EL) aligns entity mentions in unstructured text with entries in a knowledge base. Most EL systems follow a two-stage pipeline consisting of candidate retrieval and candidate re-ranking (Mihalcea and Csomai, 2007; Wu et al., 2020a; Xu et al., 2023). Candidate retrieval can be based on sparse methods (e.g., TF–IDF or BM25) or dense methods that embed mentions and entity descriptions into a shared vector space (Mustafa et al., 2024). Dense retrieval approaches, such as BLINK (Wu et al., 2020b), employ bi-encoders for scalable candidate retrieval and cross-encoders for re-ranking, achieving strong performance by jointly modelling mention context and entity descriptions. However, most EL methods assume that entities observed at test time were seen during training, which is unrealistic for large and continuously evolving KBs such as Wikidata or UMLS (Ayoola et al., 2022; Zhu et al., 2023).

Some studies explore generative (Xiao et al., 2023), context augmentation (Xin et al., 2025) and retrieval-augmented approaches for entity linking (Shlyk et al., 2024) in order to bypass the training data bottleneck. LLMAEL (Xin et al., 2025) uses generative LLMs as knowledgeable context augmenters, generating mention-centric descriptions that supplement the original text before applying traditional EL models. Rather than performing EL directly with LLMs, this approach enhances the input context while preserving task-specific EL architectures. However, a major challenge in entity linking is determining none of the candidates is correct(Zhou et al., 2024), which can be challenging.

In summary, prior works show the effectiveness of domain-adaptive language models, dense retrieval-based EL architectures, and emerging context augmentation techniques. However, existing approaches often rely on large annotated datasets, focus on canonical entity types, or make assumptions that do not hold in heterogeneous and low-resource scientific domains. Our work builds on these insights by combining weak supervision, expert validation, and modular NER and EL components to support domain-specific entity recognition and linking across four scientific domains.

3. Domains of Interest and their Entities and Knowledge Bases

Each scientific domain considered in this work defines a set of domain-specific entity types of interest and a corresponding reference knowledge base (KB) for entity linking. These entities are not covered by existing NER datasets and were defined in close collaboration with domain experts from each pilot community. Below, we summarise the target domains and their selected entity types; concrete examples and the associated KBs are reported in Appendix 7.

- **Neuroscience**: `technique`, `preparation type`, `parcellation`, `species`, `biological sex`.
- **Energy**: `energy type`, `energy storage`.
- **Maritime Transport**: `vessel type`.
- **Connected Cooperative Autonomous Mobility (CCAM)**: `vehicle type`, `VRU type`, `communication type`, `entity connection type`, `sensor type`, `scenario type`, `level of automation`.

For each domain, entities extracted from text are linked to a domain-appropriate taxonomy or ontology, including openMINDS[2] and UBERON[3] for Neuroscience, IRENA[4] for Energy, AIS vessel classifications[5] for Maritime Transport, and a subset of SINFONICA[6] and FAME[7] for CCAM.

4. Methods and Materials

4.1. Data

We introduce a collection of scientific papers[8] parsed and segmented by section with GROBID (Lopez, 2009), primarily designed for research in the development and evaluation of NLP models. This dataset contains 1,000 full-text papers from various scientific domains, including Neuroscience, Transport, and Energy, in addition to Cancer, along with 1,000 other random papers from general scientific domains (Pàmies et al., 2023). The list has been curated so that all papers in it are published using licenses that allow for legal reuse, specifically CC-BY and Public Domain. The papers

[2] openminds.docs.om-i.org
[3] ebi.ac.uk
[4] irena.org
[5] api.vtexplorer.com
[6] sinfonica.eu
[7] taxonomy.connectedautomateddriving.eu
[8] Available at https://huggingface.co/datasets/SIRIS-Lab/scilake-fulltext-corpus

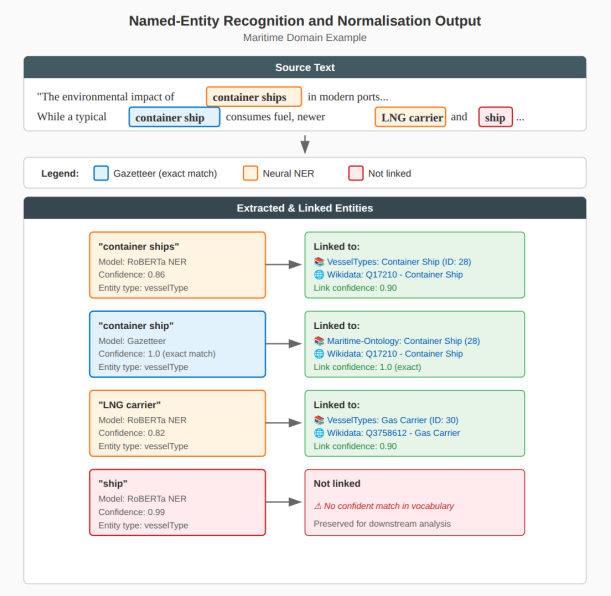


Figure 2: Example output showing entities identified and linked to domain KBs.

included in this dataset were sourced through OpenAIRE (Manghi et al., 2019), with random selection to ensure diverse content. The license information was verified by cross-referencing the publisher's landing pages, metadata from OpenAIRE, and the Unpaywall (Chawla, 2017).

4.2. Annotation

4.2.1. NER

To minimise manual annotation effort while retaining domain accuracy, we adopted a two-step annotation strategy combining LLM-based pseudo-labelling with expert refinement. For each pilot domain, we sampled 150–200 document excerpts from the corpus, consisting of titles, abstracts, and randomly selected body sections per paper (Table 1). In the first step, pseudo-labels were generated using a large language model (Claude 3.7 Sonnet)[9], prompted with general extraction instructions and domain- and entity-specific guidelines (Appendix B). The model produced candidate entity spans, and their corresponding entity types. In the second step, the pseudo-labelled data was reviewed and refined by domain experts using the Argilla (Daniel and Francisco, 2023).[10] Annotators validated entity spans and types, corrected errors, and added missing entities. This human-in-the-loop process ensured both domain relevance and annotation consistency, while substantially reducing the overall labelling effort compared to fully manual annotation.

[9] Anthropic Claude Sonnet 3.7 documentation
[10] https://docs.argilla.io/

Dataset	# Entities	# Samples
Neuroscience	5	182
Energy	2	147
CCAM	7	191
Maritime transport	1	189

Table 1: Overview of the initial annotated datasets, number of entity types defined for the domain and number of samples.

After validation, the annotated documents were segmented at the sentence level and split into training and test sets, reserving 20% of the data for evaluation in each domain.

4.2.2. Entity Linking

For each domain, we have a domain-specific target knowledge base (KB) to support entity normalisation. As summarised in Table 2, KB sizes vary substantially across domains, ranging from 22 entities in Maritime Transport to 2,565 in Neuroscience, reflecting differences in domain scope and conceptual granularity. For each domain, between 700-1,000 entity mentions were manually annotated by domain experts with links to KB entities when a suitable match existed.

Domain	# KB Entities	# Samples	NoC (%)
Neuroscience	2,565	1,000	47.9%
Energy	267	964	20.9%
CCAM	176	734	35.0%
Maritime	22	1,000	20.6%

Table 2: Statistics of the entity linking datasets. None-of-the-Candidates (NoC) denotes the number of mentions for which no suitable candidate existed in the target knowledge base.

To improve coverage, KB entries were enriched with GENRE model (Cao et al., 2021) to Wikipedia identifiers when available, enabling the inclusion of alternative labels, aliases, and acronyms. Each KB is stored in a structured format including a `unique identifier, preferred name, (optional) hierarchy, (optional) type, description, and (optional) Wikidata metadata (id, variants)`. Entity linking candidates were generated from NER-extracted mentions and a gazetteer built from KB labels and aliases. During annotation, mentions were either linked to the most appropriate KB entry or marked as `UNLINKED` when no corresponding concept existed in the KB.

Example 1 (CCAM). *"Since Distributed Acoustic Sensing constitutes a (curvi-)linear array of single-component* ***seismic sensors****, it is*

amenable to beamforming analysis."
→ linked to `SINFONICA::Sensor`

Example 2 (Maritime Transport). *"Both approaches are tested for two case studies, a bulk carrier and a **small cruise ship**."*
→ linked to `Passenger Ship`

4.3. Models & Training

4.3.1. NER

We fine-tune transformer encoder models for token classification on each domain-specific dataset, as suggested by (Devlin et al., 2019), splitting 80/20 across domains. Our model selection considers general-purpose language models (RoBERTa-base and RoBERTa-large (Liu et al., 2019)), and scientific language models pre-trained on academic documents (SPECTER2 (Cohan et al., 2020; Singh et al., 2022) and ScholarBERT (Hong et al., 2022)). We train all four general-purpose and scientific backbones from scratch on each domain's labelled data. In addition, across all four domains we fine-tune a GliNER model (Zaratiana et al., 2024), a generalist entity-recognition framework that accepts arbitrary natural-language label, providing a complementary zero-shot baseline. Full hyperparameters and training details are provided in Appendix C.

4.3.2. Entity Linking

We frame entity linking as a two-stage retrieval and reranking pipeline, systematically evaluating combinations of retrievers, rerankers, and an optional LLM-based context augmentation inspired from (Xin et al., 2025) step across our four domains. We conduct an ablation study of different retriever and reranking models, with focus on NoC. However, most models used are not explicitly trained for entity-level retrieval and ranking, they are optimized for longer text passages, which limits their effectiveness when only minimal contextual information is available. We split each domain dataset as 50% for train and 50% for test. We use TF-IDF retrieval combined with a threshold-based decision rule as a baseline for entity linking, representing a competitive baseline approach in low-resource settings.

Retrieval. Given a mention in context, the retriever returns the top-k candidates (k=10) from a domain-specific knowledge base constructed from each taxonomy's concepts, descriptions, and aliases. Each entity is encoded by concatenating the concept label, its description, Wikidata aliases, and synonyms (separated by `[SEP]` tokens). Mentions are encoded as the entity surface form followed by its surrounding sentence context (formatted as `"{mention}. Context: {sentence}"`). Following (Mustafa et al., 2024), we compare a sparse baseline (character n-gram TF-IDF) with six dense bi-encoder retrievers spanning general-purpose models (BGE-M3 (Chen et al., 2024), E5-base and E5-large-instruct (Wang et al., 2024), Jina Embeddings v3 (Sturua et al., 2024)), a biomedical model (SapBERT (Liu et al., 2021)), and a scientific model (SPECTER (Cohan et al., 2020)). We additionally include our fine-tuned variant of E5-base, trained on the entity linking pairs from our training split.

Reranking. Candidates are then rescored by one of five rerankers representing different architectural families: a cross-encoder baseline (MS MARCO MiniLM (Reimers and Gurevych, 2019)), Jina Reranker v3 (Wang et al., 2025), Qwen3-Reranker-0.6B (Zhang et al., 2025a) using yes/no logit scoring, and our fine-tuned variant of the Qwen3 reranker trained on our entity linking data, and two generative models (Qwen3-1.7B and Qwen3-8B (Team, 2025)), suggested by (Shlyk et al., 2024), that are asked to output either a candidate index or `REJECT` for NoC prediction.

Context augmentation. Following Xin et al. (2025), we optionally augment mention context with LLM-generated entity descriptions before reranking. A Qwen3-1.7B model (Team, 2025) generates a short technical description of the mention using domain-specific few-shot prompts, described in Appendix D, which is appended to the original context before being passed to the reranker.

NoC prediction and threshold tuning. As described by (Zhou et al., 2024), a key challenge in zero-shot entity linking re-ranking settings is that mentions may not correspond to any concept in the taxonomy (NoC entities). Generative rerankers handle this natively via the `REJECT` option. For score-based retrievers and rerankers, we use the training split (50%) to tune a decision threshold: mentions whose top-candidate score falls below the threshold are predicted as NoC. All reported metrics are computed on the held-out test split, focusing on evaluating the None-of-the-Candidates (NoC).

Fine-tuning. To assess importance of task adaptation given that most retriever and rerankers are trained on textual passages, not on entity linking. Two components are fine-tuned on the training split: an E5-base retriever (Wang et al., 2024), adapted via contrastive learning on (mention, gold concept) pairs, and a Qwen3-Reranker-0.6B (Zhang et al., 2025a), fine-tuned on candidate lists, using one positive and nine negative candidates per mention, where negatives are drawn from the top candidates

retrieved by a TF-IDF retriever. Hyperparameters and training details are provided in Appendix C.

5. Results & Discussion

5.1. Evaluation

We evaluate NER and Entity Linking (EL) separately using test splits for each domain. For NER, annotated sentence-level data is split into training and test sets, reserving 20% for test, and performance is measured using standard micro-average span-based precision, recall, and F1-score. For EL, annotated mentions are split evenly into training and test sets (50/50), where the training split is used for threshold tuning or model fine-tuning, when applicable. EL performance is evaluated using Accuracy@1 on the test set, complemented by Recall@K to assess retrieval quality and by precision and recall for None-of-the-Candidates (NoC) prediction.

5.2. NER

Table 3 reports micro-averaged F1-score across the four scientific domains. Zero-shot GLiNER models perform poorly in all settings, confirming that domain-specific supervision is essential for extracting specialised scientific and technical entities. Fine-tuning leads to substantial improvements for all models.

Model	CCAM	Mar.	Ener.	Neuro.
ZERO SHOT				
Gliner-large	.21	.37	.21	.29
Gliner-medium	.16	.24	.19	.24
Gliner-small	.16	.15	.28	.23
FINE-TUNED				
Gliner-large	.765	.356	.359	.660
Gliner-medium	.798	.398	.418	.680
Gliner-small	.738	.468	.476	.636
Roberta-base	.739	.745	.652	.697
Roberta-large	.804	.861	.652	.785
ScholarBERT	.775	.798	.670	.721
Specter2	.762	.822	.647	.763

Table 3: Named Entity Recognition performance across domains, reported using micro-averaged F1-score on held-out test split.

Among fine-tuned approaches, token classification models consistently outperform prompt-based methods, with RoBERTa-large achieving the strongest overall performance. Scientific pre-trained models (ScholarBERT and SPECTER2) provide stable gains in domains such as Neuroscience, CCAM and Maritime transport. Fine-tuned GLiNER models remain competitive in some domains but show higher variance, suggesting sensitivity to label semantics and prompt formulation. To improve GIINER, we note that despite the fine-tuning with the specific entity label names, more descriptive label—e.g., "neuroanatomical region" instead of `UBERONParcellation`, or "ship" or "vessel" replacing `vesselType`, to list some straightforward examples—may further improve prompt-based extraction. We leave this for future work.

Finally, the small size of the annotated training and test data remains a key limitation across domains. While fine-tuning yields strong gains, performance on sparse or semantically broad entity types remains lower. Future work could explore data augmentation strategies, such as synthetic example generation, as well as alternative few-shot and instruction-based information extraction methods, to further reduce annotation costs while maintaining domain accuracy.

5.3. Entity Linking

We evaluate entity linking as a two-stage retrieval–reranking steps and report results of each component to isolate its contribution and capacities. We first assess candidate retrieval, reranking, and the effect of an intermediate LLM-based context augmentation step.

Table 4 reports retrieval-only performance using Recall@K, which represents an upper bound on downstream linking reranking. Across domains, some dense retrievers generally outperform the TF-IDF lexical baseline, although it remains very competitive, specially for Neuroscience and Maritime. Retrieval-oriented encoders (E5, BGE-M3) outperform document-encoder models (SPECTER, JINA), indicating that task alignment for short-text retrieval is more challenging due to the lack of enough context, being SPECTER trained on scholarly documents. Instruction-tuned retrieval (E5-INSTRUCT) shows strong gains in CCAM and Maritime but degrades in Neuroscience and Energy, suggesting domain-sensitive instruction effects. Fine-tuning E5 on (mention, gold concept) pairs leads to consistent and substantial gains, with Recall@10 approaching saturation in all domains and Recall@1 improving by more than 30 points on average. This suggests that most errors in retrievers could stem from domain mismatch and most models are not trained for entity-level retrieval, rather than inherent ambiguity in the candidate set. Despite high Recall@10, Recall@1 remains comparatively low for non-fine-tuned retrievers, motivating the use of reranking.

Table 5 reports macro-averaged results across domains for each retriever–reranker combination, including Accuracy@1 and precision/recall for None-of-the-Candidates (NoC) prediction. Values

Retriever	R@1	R@5	R@10
TFIDF			
Neuroscience	.421	.633	.734
CCAM	.377	.701	.734
Energy	.373	.686	.743
Maritime	.546	.744	.876
SAPBERT			
Neuroscience	.270	.544	.633
CCAM	.270	.467	.537
Energy	.267	.478	.589
Maritime	.355	.672	.806
SPECTER			
Neuroscience	.286	.471	.556
CCAM	.295	.520	.627
Energy	.298	.488	.607
Maritime	.318	.648	.854
BGE-M3			
Neuroscience	.351	.653	.730
CCAM	.410	.664	.770
Energy	.424	.661	.740
Maritime	.447	.769	.901
E5			
Neuroscience	.351	.645	.741
CCAM	.402	.709	.811
Energy	.398	.702	.787
Maritime	.496	.836	.931
E5-INSTRUCT			
Neuroscience	.154	.282	.344
CCAM	.512	.803	.902
Energy	.201	.504	.697
Maritime	.543	.896	.965
JINA			
Neuroscience	.189	.398	.471
CCAM	.299	.541	.639
Energy	.265	.530	.630
Maritime	.397	.767	.908
E5 (FINE-TUNED)			
Neuroscience	.784	.938	.946
CCAM	.758	.939	.963
Energy	.640	.846	.923
Maritime	.739	.928	.978

Table 4: Retriever-only entity linking performance across domains. Recall@K indicates the upper bound for reranking accuracy.

Reranker	Acc@1	NoC P.	NoC R.
TFIDF (avg R@10 = .764)			
Threshold baseline	.500	0.555	.803
Cross-encoder	.440 (-0.10)	0.435	.773
Jina Reranker	.620 (+0.03)	0.543	.849
Qwen Reranker	.589 (+0.03)	0.546	.898
Generative (Qwen 1.7B)	.446 (+0.05)	0.715	.353
Generative (Qwen 8B)	.595 (-0.01)	0.495	.866
Qwen Reranker (fine-tuned)	.671 (-0.02)	0.555	.852
BGE-M3 (avg R@10 = .785)			
Threshold baseline	.437	0.420	.690
Cross-encoder	.414 (-0.11)	0.400	.779
Jina Reranker	.564 (+0.01)	0.490	.795
Qwen Reranker	.535 (+0.05)	0.484	.846
Generative (Qwen 1.7B)	.417 (+0.05)	0.566	.251
Generative (Qwen 8B)	.589 (-0.01)	0.495	.832
Qwen Reranker (fine-tuned)	.644 (-0.02)	0.527	.875
E5 (avg R@10 = .817)			
Threshold baseline	.445	0.442	.693
Cross-encoder	.413 (-0.10)	0.391	.799
Jina Reranker	.565 (+0.02)	0.466	.851
Qwen Reranker	.539 (+0.04)	0.483	.861
Generative (Qwen 1.7B)	.417 (+0.04)	0.562	.257
Generative (Qwen 8B)	.589 (-0.01)	0.502	.835
Qwen Reranker (fine-tuned)	.657 (-0.02)	0.558	.815
E5-INSTRUCT (avg R@10 = .727)			
Threshold baseline	.419	0.404	.806
Cross-encoder	.398 (-0.10)	0.386	.793
Jina Reranker	.522 (+0.02)	0.438	.854
Qwen Reranker	.521 (+0.02)	0.464	.869
Generative (Qwen 1.7B)	.338 (+0.03)	0.483	.106
Generative (Qwen 8B)	.545 (-0.00)	0.525	.775
Qwen Reranker (fine-tuned)	.617 (-0.03)	0.515	.854
JINA (avg R@10 = .662)			
Threshold baseline	.389	0.389	.818
Cross-encoder	.407 (-0.09)	0.382	.785
Jina Reranker	.526 (+0.01)	0.446	.815
Qwen Reranker	.513 (+0.04)	0.463	.862
Generative (Qwen 1.7B)	.408 (+0.05)	0.503	.398
Generative (Qwen 8B)	.527 (+0.00)	0.436	.859
Qwen Reranker (fine-tuned)	.589 (-0.01)	0.458	.886
E5-FINETUNED (avg R@10 = .952)			
Threshold baseline	.689	0.617	.778
Cross-encoder	.422 (-0.12)	0.410	.737
Jina Reranker	.598 (+0.02)	0.519	.778
Qwen Reranker	.573 (+0.04)	0.532	.863
Generative (Qwen 1.7B)	.419 (+0.08)	0.603	.218
Generative (Qwen 8B)	.598 (-0.01)	0.504	.839
Qwen Reranker (fine-tuned)	.681 (-0.00)	0.577	.821

Table 5: Entity linking performance of reranking models using a fixed retriever (with R@10 ceiling), macro-averaged across domains. Values in parentheses indicate the change in Acc@1 obtained by augmenting mention context with LLM-generated descriptions. Thresholds are fixed on the training split.

in parentheses indicate the average change in Acc@1 when augmenting mention context with LLM-generated descriptions.

Across all retrievers, the threshold-based baseline provides a strong and competitive reference point, particularly for NoC detection. Its high NoC recall indicates that simple score calibration is already effective at rejecting spurious candidates, and most rerankers do not substantially improve this aspect. This highlights that NoC prediction is largely constrained by retrieval confidence rather than by ranking capacity.

Generative rerankers show mixed behaviour, while they occasionally improve accuracy, they tend to trade off NoC recall for precision, making them less reliable in settings where rejection is critical. Fine-tuning the Qwen reranker leads to consistent gains over its base version, but these improvements remain moderate compared to the substantial gains achieved through retriever fine-tuning. This suggests that reranking performance is still bottlenecked by candidate quality and by the lim-

ited amount of task-specific supervision available for reranker adaptation.

The impact of LLM-based context augmentation remains inconsistent across domains. While small improvements are observed in some configurations, augmentation does not systematically improve reranking performance and can introduce noise or redundant information in some cases. This suggests that current prompt-based augmentation strategies may lack the level of control required for fine-grained entity-level disambiguation. Retrieval quality remains the dominant factor in the overall performance, with context augmentation providing only secondary gains.

Domain	Acc@1	NoC P.	NoC R.
CCAM	.576 (+0.14)	.539	.671
Energy	.661 (+0.08)	.503	.866
Maritime	.646 (+0.11)	.422	.851
Neuroscience	.830 (+0.15)	.771	.963

Table 6: Entity linking performance across domains using a fine-tuned E5 retriever and fine-tuned Qwen reranker. Values in parentheses indicate the absolute change in Acc@1 compared to the non-finetuned configuration.

Table 6 reports per-domain results using the fully fine-tuned configuration (E5 retriever + Qwen reranker). Performance varies considerably across domains, with the strongest gains observed in Neuroscience and the weakest in CCAM and Maritime. Interestingly, this variation correlates more strongly with size of their KBs. Domains with large, fine-grained taxonomies and lexically specific entities (e.g., neuroanatomical regions) benefit from closer surface-form alignment between mentions and concepts. In contrast, the maritime KB involve an smaller taxonomy but require higher levels of abstraction (e.g., mapping “small cruise ship” to “passenger ship”’), increasing semantic distance between mentions and KB entries and making linking more challenging. However, under specific configurations, context augmentation can provide benefits across domains with both large and small KBs, as observed in Neuroscience and CCAM/Maritime in Table 6. This suggests that its effectiveness is not only dependent on KB size, but also on the interaction between retrieval quality and the generated contextual information.

These results highlight that cross-domain variability is not only a consequence of model performance, but is strongly influenced by properties of the target knowledge bases and domains. In particular, domains with larger and more fine-grained taxonomies benefit from improved lexical alignment, while smaller or more abstract taxonomies require higher levels of semantic generalisation. This suggests that entity linking performance in scientific domains is inherently tied to knowledge base properties, and that domain-agnostic pipelines must account for differences in conceptual granularity and coverage when transferring across areas.

Despite the substantial gains achieved through fine-tuning and modular evaluation, named entity recognition and entity linking remain challenging in scientific domains with limited and specialised training data. Domain mismatch, abstract terminology, and sparse supervision continue to affect both retrieval and reranking, particularly for low-frequency or weakly lexicalised entities, specially dealing with low-quality KBs.

Moreover, the enrichment of large-scale SKGs introduces constraints on efficiency and scalability. While large language models enable richer context modelling and semantic abstraction, their computational cost limits large-scale deployment. Future work must therefore balance enrichment quality with throughput, exploring entity-oriented reranking, favouring lightweight or hybrid pipelines that apply expensive components selectively. In this setting, human-in-the-loop approaches remain essential in low-resource and highly-specialised domains. Future work should also explore more robust rejection mechanisms and tighter integration between retrieval and reranking for improved NoC prediction, as handling these cases remains a persistent challenge.

6. Conclusions

In this work, we presented a modular evaluation of named entity recognition and entity linking across four scientific domains, explicitly separating retrieval, reranking, and context augmentation, motivated by the fragmented and heterogeneous nature of scholarly content and the need to ground unstructured text to well-defined concepts in Scholarly Knowledge Graphs and domain-specific knowledge bases. Our results suggest that domain-specific fine-tuning is useful for both tasks, with retrieval quality emerging as the primary driver of end-to-end entity linking performance. Reranking and context augmentation provide consistent but secondary improvements, particularly in ambiguous cases. While the observed performance improvements are incremental, our contribution lies in providing a modular and domain-adaptive pipeline, together with a systematic evaluation across different scientific domains, which remains underexplored in prior work. Overall, our findings highlight the importance of efficient, domain-adaptive, and human-guided approaches for scalable knowledge graph enrichment in low-resource domains.

7. Acknowledgements

Supported by the Industrial Doctorates Plan of the Department of Research and Universities of the Generalitat de Catalunya, by Departament de Recerca i Universitats de la Generalitat de Catalunya (grant reference 2022/DI /00017). This work was co-funded by the EU HORIZON project SciLake (Grant Agreement 101058573) and the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[11] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.). The consortium is funded by the Federal Republic of Germany and its states through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the funders. Neither the European Union nor the granting authority can be held responsible for them.

We thank the anonymous reviewers for their constructive feedback and suggestions, which improved the clarity and quality of this work.

8. Bibliographical References

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. Refined: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval.

Dalmeet Singh Chawla. 2017. Unpaywall finds free versions of paywalled papers. *Nature*.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. Specter: Document-level representation learning using citation-informed transformers.

Vila-Suero Daniel and Aranda Francisco. 2023. Argilla - Open-source framework for data-centric NLP.

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.

Carol Friedman, Pauline Kra, and Andrey Rzhetsky. 2002. Two biomedical sublanguages: a description based on the theories of zellig harris. *Journal of biomedical informatics*, 35(4):222–235.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee. 2020. Infusing Disease Knowledge into BERT for Health Question Answering, Medical Inference and Disease Name Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4604–4614, Online. Association for Computational Linguistics.

Ginny Hendricks, Dominika Tkaczyk, Jennifer Lin, and Patricia Feeney. 2020. Crossref: The sustainable source of community-owned scholarly metadata. *Quantitative Science Studies*, 1(1):414–427.

Zhi Hong, Aswathy Ajith, Gregory Pauloski, Eamon Duede, Carl Malamud, Roger Magoulas, Kyle Chard, and Ian T. Foster. 2022. Scholarbert: Bigger is not always better. *ArXiv*, abs/2205.11342.

Zhi Hong, Roselyne Tchoua, Kyle Chard, and Ian Foster. 2020. Sciner: extracting named entities from scientific literature. In *International Conference on Computational Science*, pages 308–321. Springer.

Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer.

[11] https://www.nfdi4datascience.de

2019. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. In *Proceedings of the 10th international conference on knowledge capture*, pages 243–246.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of genia event task in bionlp shared task 2011. In *Proceedings of BioNLP shared task 2011 workshop*, pages 7–15.

Jeremy Leipzig, Daniel Nüst, Charles Tapley Hoyt, Karthik Ram, and Jane Greenberg. 2021. The role of metadata in reproducible computational research. *Patterns*, 2(9):100322.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Mingchen Li and Rui Zhang. 2023. How far is language model from 100% few-shot named entity recognition in medical domain. *arXiv preprint arXiv:2307.00186*.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Yiren Liu, Si Chen, Haocong Cheng, Mengxia Yu, Xiao Ran, Andrew Mo, Yiliu Tang, and Yun Huang. 2024. How ai processing delays foster creativity: Exploring research question co-creation with an llm-based agent. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Patrice Lopez. 2009. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer.

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

Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. 2023. Aioner: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics*, 39(5):btad310.

Paolo Manghi. 2024. Challenges in building scholarly knowledge graphs for research assessment in open science. *Quantitative Science Studies*, 5(4):991–1021.

Paolo Manghi, Alessia Bardi, Claudio Atzori, Miriam Baglioni, Natalia Manola, Jochen Schirrwagen, Pedro Principe, Michele Artini, Amelie Becker, Michele De Bonis, et al. 2019. The openaire research graph data model. *Zenodo*.

Rada Mihalcea and Andras Csomai. 2007. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 233–242, New York, NY, USA. Association for Computing Machinery.

Faizan E Mustafa, Corina Dima, Juan Ochoa, and Steffen Staab. 2024. Leveraging Wikidata for biomedical entity linking in a low-resource setting: A case study for German. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 202–207, Mexico City, Mexico. Association for Computational Linguistics.

Cameron Neylon, Bianca Kramer, Alysson Fernandes Mazoni, Rodrigo Costas, Najko Jahn, and Nees Jan van Eck. 2026. Sharing the load: Building a collective to support open research information online. *Upstream*.

Marc Pàmies, Joan Llop, Francesco Multari, Nicolau Duran-Silva, César Parra-Rojas, Aitor Gonzalez-Agirre, Francesco Alessandro Massucci, and Marta Villegas. 2023. A weakly supervised textual entailment approach to zero-shot text classification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–296.

Silvio Peroni and David Shotton. 2020. Opencitations, an infrastructure organization for open scholarship. *Quantitative Science Studies*, 1(1):428–444.

Jason Priem, Heather A. Piwowar, and Richard Orr. 2022. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *ArXiv*, abs/2205.01833.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Wei Liang Seow, Iti Chaturvedi, Amber Hogarth, Rui Mao, and Erik Cambria. 2025. A review of named entity recognition: from learning methods to modelling paradigms and tasks. *Artificial Intelligence Review*, 58(10):315.

Darya Shlyk, Tudor Groza, Marco Mesiti, Stefano Montanelli, and Emanuele Cavalleri. 2024. REAL: A retrieval-augmented entity linking approach for biomedical concept recognition. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 380–389, Bangkok, Thailand. Association for Computational Linguistics.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2022. Scirepeval: A multi-format benchmark for scientific document representations. In *Conference on Empirical Methods in Natural Language Processing*.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora.

Sina Taslimi, Artemis Capari, Hosein Azarbonyad, Zi Long Zhu, Zubair Afzal, Evangelos Kanoulas, and George Tsatsaronis. 2025. Extracting, detecting, and generating research questions for scientific articles. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8573–8588.

Qwen Team. 2025. Qwen3 technical report.

Thanasis Vergoulis, Serafeim Chatzopoulos, Ilias Kanellos, Panagiotis Deligiannis, Christos Tryfonopoulos, and Theodore Dalamagas. 2019. Bip! finder: Facilitating scientific literature search by exploiting impact-based ranking. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2937–2940.

Feng Wang, Yuqing Li, and Han Xiao. 2025. jina-reranker-v3: Last but not late interaction for document reranking.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Max Wilkinson. 2010. Datacite: The international data citation initiative: Datasets programme.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020a. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020b. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Mingfang Wu, Felicitas Löffler, Brigitte Mathiak, Fotis Psomopoulos, Uwe Schindler, Amir Aryani, Jordi Bodera Sempere, Antica Culina, Andreas Czerniak, Chris Erdmann, et al. 2026. Bridging the data discovery gap: User-centric recommendations for research data repositories. *Data Science Journal*, 25(6).

Zilin Xiao, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, and Daxin Jiang. 2023. Instructed language models with retrievers are powerful entity linkers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2267–2282, Singapore. Association for Computational Linguistics.

Amy Xin, Yunjia Qi, Zijun Yao, Fangwei Zhu, Kaisheng Zeng, Bin Xu, Lei Hou, and Juanzi Li. 2025. Llmael: Large language models are good context augmenters for entity linking. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 3550–3559.

Zhenran Xu, Yulin Chen, Baotian Hu, and Min Zhang. 2023. A read-and-select framework for zero-shot entity linking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13657–13666, Singapore. Association for Computational Linguistics.

Artur Yakimovich, Anaël Beaugnon, Yi Huang, and Elif Ozkirimli. 2021. Labels in a haystack: Approaches beyond supervised learning in biomedical applications. *Patterns*, 2(12):100383.

Anthony Yazdani, Ihor Stepanov, and Douglas Teodoro. 2025. Gliner-biomed: A suite of efficient models for open biomedical named entity recognition. *arXiv preprint arXiv:2504.00676*.

Zheng Yuan, Yijia Liu, Chuanqi Tan, Songfang Huang, and Fei Huang. 2021. Improving biomedical pretrained language models with knowledge. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 180–190, Online. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025a. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Rui Li, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. 2025b. Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In *Proceedings of the ACM on Web Conference 2025*, pages 2032–2042.

Kang Zhou, Yuepei Li, Qing Wang, Qiao Qiao, and Qi Li. 2024. GenDecider: Integrating "none of the candidates" judgments in zero-shot entity linking re-ranking. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 239–245, Mexico City, Mexico. Association for Computational Linguistics.

Minghao Zhu and Keyuan Jiang. 2021. Semi-supervised language models for identification of personal health experiential from Twitter data: A case for medication effects. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 228–237, Online. Association for Computational Linguistics.

Tiantian Zhu, Yang Qin, Qingcai Chen, Xin Mu, Changlong Yu, and Yang Xiang. 2023. Controllable contrastive generation for multilingual biomedical entity linking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5742–5753, Singapore. Association for Computational Linguistics.

A. Table of Entities and Target KGs

Table 7 summarises the entities of interest for each of the pilots, along with their corresponding taxonomy of choice for linking and integration within the corresponding SKGs.

B. Pseudo-labelling instructions

General Instructions

```
Extract and categorize these [DOMAIN_NAME]
     entities, focusing on precise entity
     identification.

Entity types:

1. ...
2. ...
3. ...

Output Format

For each document, produce a json output of
     the form

{
    "entities": [
        {
            "entity": the EXACT entity text,
            "label": the entity type
        }
    ]
}

The 'entity' field must contain the exact
     text as it appears in the document (
     maintaining capitalization). When an
     acronym is introduced with its full
     form (e.g., 'electroencephalogram (EEG)
     '), extract the complete text including
      both the full form and the acronym.
     When only the acronym is used
     subsequently (e.g., just 'EEG'),
     extract only the acronym. Never expand
     acronyms that aren't explicitly
     expanded in the text, and never modify
     the entity text in any way from how it
     originally appears.

Only map entities to taxonomy categories
     when there is explicit textual evidence
      supporting that specific
     categorization. Do not make assumptions
      about entities based on common
     associations if not stated in the text.
      Maintain a high standard of evidence---
     if in doubt, do not extract the entity.

Do not ignore documents without extracted
     entities. Provide outputs for all
     documents, even if empty
```

C. Experimental setup

We provide experimental details of our fine-tuned models. For *fine-tuning* NER, GLiNER models, re-

Domain	Entity	Examples	Relevant Taxonomy
Neuroscience	`technique`	DNA sequencing; sodium MRI; voltage clamp	openMINDs
	`preparationType`	in vitro; in vivo	openMINDs
	`UBERONParcellation`	spinal cord; pineal tract; medial orbital frontal cortex	UBERON, restricted to Central Nervous System
	`species`	mice; human; bovine	openMINDS
	`biologicalSex`	male; female	openMINDS
Energy	`energyType`	biogas; photovoltaic; peat	IRENA
	`energyStorage`	pumped hydro storage; thermal storage; ultracapacitors	IRENA
Maritime	`vesselType`	bulk carrier; ferry; yacht	Ad-hoc based on AIS ship types
CCAM	`vehicleType`	car; truck; shuttle	Ad-hoc based on SINFONICA and FAME
	`VRUType`	pedestrian; scooter; bicycle	Ad-hoc based on SINFONICA and FAME
	`communicationType`	5G; cellular communication; DSRC	Ad-hoc based on SINFONICA and FAME
	`entityConnectionType`	vehicle-to-pedestrian; I2X; V2V	Ad-hoc based on SINFONICA and FAME
	`sensorType`	camera; LIDAR; odometer	Ad-hoc based on SINFONICA and FAME
	`scenarioType`	platooning; traffic scenario; test scenario	Ad-hoc based on SINFONICA and FAME
	`levelOfAutomation`	electronic stability control; lane centering; park assist	Ad-hoc based on SINFONICA and FAME

Table 7: List of research communities and their entities of interest, with the correspondent taxonomy of choice.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	$2e^{-5}$
Learning rate schedule	Linear
Warm-up steps	100
Weight decay	0.01
Training batch size	16
Evaluation batch size	16
Maximum epochs	6

Table 8: Hyperparameters used for fine-tuning NER models across all domains.

triever and reranker models, we make use of the `huggingface` library. Training was run (using 1x NVIDIA A100 GPU) for NER models with hyperparameter defined Table 8, GLiNER models in Table 9, E5 retriever model and for Qwen Reranker in Table 10.

Hyperparameter	Value
Training objective	Focal loss
Learning rate	$5e^{-6}$
Learning rate (non-backbone)	$1e^{-5}$
Weight decay	0.01
Batch size	8
Warm-up ratio	0.1
Learning rate scheduler	Linear
Prediction threshold	0.5
Evaluation metric	seqeval F_1

Table 9: Hyperparameters used for fine-tuning GLiNER models across domains.

Hyperparameter	Value
Training objective	MNRLoss
Optimizer	AdamW
Learning rate	$2e^{-5}$
Batch size	32
Epochs	3
Warm-up ratio	0.1
Negative sampling	In-batch negatives
Hard negatives	10 (TF–IDF)
Mixed precision (AMP)	Enabled
Random seed	42

Table 10: Hyperparameters for fine-tuning the E5 model for entity linking retrieval and reranking.

D. Prompt Templates for Context Augmentation

For context augmentation, we employ a few-shot prompting strategy inspired by LLMaEL (Xin et al., 2025), using 2–3 domain-specific examples per domain (3 for maritime transport, 2 for neuroscience, energy, and CCAM), followed by the target mention in context, as described in the fllowing template.

Context Augmentation Template for Entity Linking

```
Example i. Consider the following text from
     a scientific or technical document.
Text: {EXAMPLE_TEXT_WITH_MARKED_ENTITY}
Please provide more descriptive information
     about { {EXAMPLE_MENTION} } from the
     text above.
Make sure to include {EXAMPLE_MENTION} in
     your description.
Answer:
{EXAMPLE_DESCRIPTION}

Now consider the following text from a
     scientific or technical document.
Text: {LEFT_CONTEXT} { {MENTION} } {
     RIGHT_CONTEXT}
Please provide more descriptive information
     about { {MENTION} } from the text above.

Make sure to include {MENTION} in your
     description.
Answer:
```

Evaluating Generative Large Language Models for Portuguese Scientific Information Extraction

Tomás Pinto, Catarina Silva, Hugo Gonçalo Oliveira
University of Coimbra, CISUC/LASI, DEI, Coimbra, Portugal
{tomaspinto, catarina, hroliv}@dei.uc.pt

Abstract

Scientific Information Extraction (IE), which identifies entities and their relations from scientific texts, is essential for building Scientific Knowledge Graphs (SciKGs) that encode structured knowledge and enable applications such as semantic search, question answering, and literature reasoning. Large Language Models (LLMs) have shown strong capabilities in processing unstructured text, yet most advances focus on English, with limited exploration for less-resourced languages like Portuguese. The reliability of generative LLMs, including Portuguese-targeted models like the sovereign AMALIA, for structured extraction of scientific knowledge from literature text remains underexplored. We evaluate low- to mid-scale generative LLMs (8–12B parameters) on scientific Named Entity Recognition (NER) and Relation Extraction (RE), using a Portuguese-translated dataset of computer science article abstracts. Overall, our results show moderate performance and indicate that the adaptation strategy has a greater impact than model choice: prompting yields unstable performance with poor RE scores, while fine-tuning consistently improves both NER and RE and reduces cross-model variability. These findings suggest that, at this scale, prompting alone is insufficient for SciKG construction and underscore the need for supervised adaptation. We provide a detailed error analysis and outline directions for advancing Portuguese scientific IE.

Keywords: Scientific Information Extraction, Large Language Models, Portuguese Language, Knowledge Graphs

1. Introduction

The exponential growth of scientific literature has made it increasingly difficult for researchers to comprehensively access, integrate, and reason over existing knowledge (Hanson et al., 2024). This challenge is exacerbated by the predominantly unstructured nature of scientific texts, where core contributions, experimental results, and relationships between concepts are expressed in free-form language rather than in machine-interpretable formats. As a result, transforming unstructured scientific documents into structured representations is a fundamental prerequisite for scalable knowledge access, integration, and discovery (Asai et al., 2026).

Automated scientific information extraction (IE) plays a central role in addressing this problem (Hong et al., 2021), as it enables the identification of entities, relations, and higher-level semantic structures that can support downstream applications such as knowledge graph construction, literature exploration, and question answering (Zhong et al., 2023).

Large language models (LLMs) have recently demonstrated strong performance on a variety of Natural Language Processing (NLP) tasks (Yang et al., 2024), including core IE subtasks such as named entity recognition (NER), relation extraction (RE), and joint entity–relation extraction, showing the ability to compete with traditional supervised approaches (Xu et al., 2024).

However, most existing studies focus on high-resource languages, particularly English, and relatively little is known about the behaviour of LLM-based scientific IE systems in multilingual or low-resource settings (Zhao et al., 2024). This gap is particularly evident for scientific texts in Portuguese, where domain-specific terminology, limited annotated resources, and linguistic variation pose additional challenges, despite the existence of relevant publications in repositories such as RCAAP[1] and SciELO[2].

In parallel, there is growing interest in using LLMs as scalable engines for constructing scientific knowledge graphs (SciKGs) (Yang et al., 2025b), which provide structured, interpretable, and interlinked representations of scientific knowledge (Ding et al., 2025). In such settings, the quality and characteristics of extracted entities and relations directly shape the resulting graph structure and constrain which information can be retrieved and reasoned over in downstream tasks, including graph-grounded retrieval-augmented generation (GraphRAG) (Zhang et al., 2025a; Peng et al., 2025) and question answering (Ma et al., 2025). Understanding the trade-offs of different extraction strategies is therefore not only an IE problem, but also a foundational concern for reliable scientific knowledge infrastructures.

In this work, we present an empirical study of LLM-based scientific IE for European Portuguese,

[1] https://www.rcaap.pt/
[2] https://www.scielo.org/

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 155–167
12 May 2026.

focusing on generative models around the 10B parameter scale. This includes AMALIA[3] (Simplício et al., 2026), a sovereign PT-PT LLM, alongside other comparably-sized open models to enable controlled benchmarking while avoiding the extreme computational and monetary costs associated with the largest models. We evaluate multiple extraction paradigms, including pipelined NER and RE, joint extraction via prompting, and fine-tuning-based adaptation strategies. Experiments are conducted on a European Portuguese version of the SCIERC dataset (Luan et al., 2018), which we translated and curated as part of this work to establish a benchmark for controlled evaluation of scientific IE in Portuguese.

Beyond reporting extraction performance, we analyse how different modelling and adaptation choices affect properties that are critical for downstream KG construction, such as modularity, error isolation, and schema compatibility. Based on these observations, we discuss the implications for building reliable SciKGs and outline future research directions towards end-to-end LLM–SciKG pipelines that support structured reasoning and scientific discovery, particularly in multilingual and low-resource contexts.

2. Background and Related Work

SciKGs structure entities and semantic relations from scholarly literature into explicit relational graph representations, capturing dependencies among concepts, methods, tasks, datasets, metrics, and findings (Auer et al., 2018; Ding et al., 2025). This structured modelling enables advanced applications such as semantic search, literature exploration, and question-answering systems (Peng et al., 2023; Auer et al., 2023). The quality of SciKGs is inherently dependent on the performance of upstream IE. Scientific IE aims to identify entities and semantic relations within research literature, enabling structured access to scientific knowledge (Verma et al., 2023). NER identifies domain concepts, while RE links them into structured triples. Early approaches relied on feature-based methods and later neural sequence labelling architectures for NER, combined with supervised classifiers for RE (Zhao et al., 2024). These components were typically organised in pipelined systems, where entity detection precedes relation prediction (Zhong and Chen, 2021).

While pipelined approaches offer modularity and interpretability, they are susceptible to error propagation where incorrect entity spans or types can directly degrade relation prediction performance (Yan et al., 2022). To mitigate this issue, joint extraction models were proposed to simultaneously predict entities and relations within a unified architecture (Santosh et al., 2021).

Benchmark datasets such as SciERC (Luan et al., 2018), SemEval-2018 Task 7 (Gábor et al., 2018), and later SCIER (Zhang et al., 2024) have supported progress in scientific IE by providing annotated entities and relations grounded in scientific discourse. These datasets are designed for scientific text in relatively general research domains, with entity types such as *Task*, *Method*, and *Material*, and relation types such as *used-for*, *part-of*, and *feature-of*. Rather than focusing on a single specialised domain like biomedicine, they aim to capture cross-domain conceptual structures common across many areas of scientific writing.

Recent work increasingly integrates SciKGs with LLMs (Zhu et al., 2025; Yang et al., 2025c). Graph-based retrieval and GraphRAG use structured context to ground generation and reduce hallucinations, while LLMs are also used to construct and update graphs (Yang et al., 2025b; Peng et al., 2025). This bidirectional synergy depends critically on robust extraction, as graph quality determines the effectiveness of grounding and reasoning.

LLMs have reshaped IE by replacing task-specific architectures with generative, prompt-based formulations (Xu et al., 2024). Through instruction prompting and in-context learning, entities and relations can be produced directly as structured outputs, often in a single pass. However, generative prediction introduces new challenges. Models must internalise schema constraints, produce well-formed outputs, and remain consistent despite decoding variability, while exact-match evaluation becomes sensitive to boundary and formatting differences (Zhang et al., 2025b). Performance is highly dependent on the adaptation strategy. Zero-shot prompting requires no annotated data but can struggle in specialised domains due to unfamiliar terminology and implicit schemas (Brown et al., 2020). Few-shot prompting can improve schema alignment and output consistency by providing task-specific demonstrations, but remains sensitive to example selection and context length limits. Fine-tuning, whether full-parameter or parameter-efficient, typically yields greater stability and reduced formatting errors (Dagdelen et al., 2024). Nevertheless, it requires annotated data, hyperparameter tuning, and additional computational resources, creating a trade-off between performance, cost, and scalability.

NLP in most languages, particularly in specialised domains such as scientific text processing, remains comparatively underexplored when compared to English. For instance, Portuguese NLP has advanced in areas such as general Open IE (Silva et al., 2024; Cabral et al., 2024), narrative

[3] https://amaliallm.pt/

extraction (Nunes et al., 2024), clinical/biomedical IE (Sousa et al., 2023), but these lack the domain-agnostic, paper-centric scientific focus of SCIERC-style benchmarks.

Multilingual LLMs are often trained on data heavily skewed toward English, which can lead to uneven performance in Portuguese tasks (Xu et al., 2025). In response, there has been increasing interest in sovereign and regionally developed LLMs that prioritize linguistic fidelity, cultural representation, and local data governance (Bondarenko et al., 2025). A prominent example in Portugal is AMALIA, a government-backed sovereign LLM designed specifically for the European Portuguese context.

Building on these developments, systematically comparing models specifically targeted to Portuguese with similarly sized multilingual counterparts, particularly in scientific IE settings, represents an open field of possibilities. Such comparisons could clarify the benefits of language-specific modeling choices and inform the design of more reliable extraction methods. This work contributes to addressing this gap by evaluating mid-scale instruction-tuned LLMs, including the sovereign AMALIA model, under different extraction paradigms and adaptation strategies. We position scientific IE as a foundational component of the SciKG construction pipeline, and discuss how modelling choices at the extraction stage influence the reliability of downstream knowledge graphs.

3. Experimental Setup

This section describes the dataset, models, extraction paradigms, adaptation strategies, and evaluation protocol used in our experiments.

3.1. Dataset

We translate the SCIERC dataset (Luan et al., 2018) into Portuguese and conduct our experiments on this resulting resource. SCIERC is a widely used benchmark for scientific IE built from abstracts of computer science research papers. Each instance in SCIERC consists of a single abstract annotated with entity mentions and semantic relations between them, reflecting core elements of scientific discourse. A simple example of an annotated entry from our translated corpus is provided in Table 1. The dataset comprises a total of 500 abstracts, split into training (70%), development (10%), and test (20%) sets.

SCIERC defines a fixed schema of scientific entity and relation types, detailed in Table 2. The entity types cover both high-level research concepts and more general scientific terminology while relations capture common semantic connections in scientific writing. This combination of entity and relation types makes SCIERC particularly suitable for studying IE as a foundation for SciKG construction.

Text
As fontes de dados de treino adequadas para modelação da linguagem da fala conversacional são limitadas. Neste artigo, mostramos como os dados de treino podem ser complementados com texto da web filtrado de modo a corresponder ao estilo e/ou ao tópico da tarefa de reconhecimento alvo, e também que é possível obter maiores ganhos de desempenho a partir desses dados através da utilização de interpolação de N-gramas dependente da classe.
Entities
{"text": "modelação da linguagem", "type": "Method"}, {"text": "fala conversacional", "type": "Material"}, {"text": "tarefa de reconhecimento", "type": "Task"}, {"text": "interpolação de N-gramas dependente da classe", "type": "Method"}
Relations
{"head": "fala conversacional", "tail": "modelação da linguagem", "relation": "Usado-para"}, {"head": "interpolação de N-gramas dependente da classe", "tail": "tarefa de reconhecimento", "relation": "Usado-para"}

Table 1: Example instance from our Portuguese SCIERC translation.

Type	**Description**
Entity Types	
Task	Applications, problems to solve, systems to construct.
Method	Techniques, models, tools, components of a system, frameworks.
Evaluation Metric	Criteria or measures used to assess quality and performance.
Material	Data, datasets, resources, corpus, knowledge base.
Other Sci Terms	Scientific concepts that do not fall into any of the above classes.
Generic	Non-specific references to other entities.
Relation Types	
Used-for	A method/material applied to a task or method.
Feature-of	Relates a property, attribute, or quality to the subject that possesses it.
Hyponym-of	An entity is a more specific type of another, more general entity.
Part-of	An entity constitutes an integral part of a larger whole.
Compare	Comparison or contrast between two entities.
Conjunction	Symmetric relationship between two entities with a similar function or role.
Evaluate-for	One entity is used to evaluate another entity.

Table 2: SCIERC Entity and Relation Types

To enable controlled evaluation in European Portuguese (PT-PT), we translated the full SCIERC dataset into Portuguese using GPT-5.1. Due to structural differences between English and Portuguese, such as word order and morphology, automatic translation can introduce misalignments between annotated entity spans, relations, and the translated text.

To mitigate this issue, we first employed an automatic consistency-checking script to identify candidate misalignments. Specifically, all instances were processed to flag cases where annotated entities, both in the entity lists and as head and tail

Model	Parameters	Language Focus
AMALIA	9B	Portuguese (PT-PT)
Gervasio-8b-ptpt-decoder	8B	Portuguese (PT-PT)
Boto-9B-it	9B	Portuguese (PT-BR)
EuroLLM-9B-Instruct-2512	9B	Multilingual
Llama-3.1-8B-Instruct	8B	Multilingual
Qwen3-8B	8B	Multilingual
Gemma-3-12b-it	12B	Multilingual

Table 3: Generative decoder-only LLMs used in the IE experiments (FP16 variants).

elements in relation triples, did not explicitly appear in the translated text. This process allowed us to focus on likely error instances where entity spans no longer matched the translated content (e.g., due to reordering, inflection, or lexical variation). The flagged subset corresponded to approximately half of the dataset entries.

These flagged instances were then manually reviewed by a reviewer with a technical background in NLP and familiarity with scientific text. The review focused on restoring alignment between annotations and text. When misalignments were identified, either entity boundaries or the translated text were minimally adjusted to ensure consistency, preserving the intended meaning. Relation annotations were not modified and the original annotation schema, including entity and relation types, was strictly preserved.

Despite this review process, it is possible that minor residual inconsistencies persist in the translated version. Such potential noise is an inherent limitation of cross-lingual dataset adaptation and is taken into account when interpreting the experimental results.

This translated version of SCIERC constitutes a valuable resource for studying scientific IE in Portuguese and for assessing the capabilities of Portuguese-focused and multilingual LLMs on structured scientific processing. Although it is based on translated rather than native Portuguese text, it provides a controlled benchmark for this setting. We make the dataset and associated scripts publicly available on GitHub[4] to facilitate further research in this area.

3.2. Models

Our experiments focus on low to mid-scale instruction-tuned LLMs in the 8–12B parameter range. This size bracket reflects a practical balance between computational feasibility and model capacity, while allowing controlled comparison without large scale-induced performance disparities. Table 3 presents the models considered in this study, including their parameter sizes and language focus.

The evaluated models include:

- AMALIA[5] (9B) (Simplício et al., 2026), a sovereign European Portuguese LLM designed to prioritise PT-PT linguistic fidelity, preserve Portugal's cultural representation, and ensure data governance within the national research framework. It is derived from EuroLLM-9B by incorporating new sources of European Portuguese data during the annealing phase of pretraining.
- Gervásio (8B) (Santos et al., 2024), a European Portuguese model built on LLaMA 3.1 8B Instruct, with additional Portuguese-focused adaptation (mostly European).
- Boto[6] (9B), a Portuguese chat model obtained through fine-tuning Gemma2-9B-it, primarily optimised for Brazilian Portuguese. As of February 2026, it ranks highest among Portuguese-focused chat models in the 8B range on the Open Portuguese LLM Leaderboard[7] (Hugging Face).
- EuroLLM (9B) (Ramos et al., 2026), the multilingual foundation model underlying AMALIA, developed to support all European Union languages, including European Portuguese.
- LLaMA3.1 (8B) (Grattafiori et al., 2024), a multilingual instruction-tuned model supporting Portuguese, included as a strong general-purpose baseline without explicit PT specialisation.
- Qwen3 (8B) (Yang et al., 2025a), a multilingual instruction-tuned model that, as of February 2026, ranks highest overall among 8B chat models on the HuggingFace Portuguese leaderboard.
- Gemma3 (12B) (Team et al., 2025), a multilingual 12B instruction-tuned model, included to examine whether moderate scaling beyond the 8–9B range improves robustness in structured scientific extraction.

All models are evaluated under the same experimental conditions and deployed using their FP16 (half-precision) variants to ensure consistent memory usage and computational efficiency across experiments. We opt for instruction-tuned to better follow the task instructions presented. Unless otherwise stated, identical prompting templates, decoding parameters, and fine-tuning procedures are applied to all models.

[4] https://github.com/TomasCCPinto/NSLP26_SciIE_PT

[5] https://amaliallm.pt/

[6] https://huggingface.co/lucianosb/boto-9B-it

[7] https://huggingface.co/spaces/eduagarcia/open_pt_llm_leaderboard

3.3. Extraction Strategies

We explore two complementary strategies for scientific IE: a pipelined formulation that separates entity and relation extraction, and a joint formulation that performs end-to-end extraction in a single model invocation.

In the pipelined approach, extraction is decomposed into two stages: NER, which identifies and classifies entity mentions according to the SCIERC schema, followed by RE, which predicts relations between the extracted entity pairs. This formulation should provide great modularity, allowing independent analysis and refinement of each subtask and clearer error isolation. However, errors in NER propagate directly to RE, potentially limiting overall robustness.

In the joint extraction approach, entity recognition and relation prediction are performed simultaneously through a single prompt that generates structured subject–predicate–object triples with explicit entity spans and types. This end-to-end formulation enables the model to leverage global contextual dependencies and directly produce graph-ready outputs. While joint extraction may reduce cascading errors, it sacrifices modularity and makes error localisation and schema enforcement more difficult.

Overall, the comparison reflects core trade-offs for SciKG construction, including modularity versus integration, error isolation versus joint reasoning, and explicit schema control versus direct triple generation.

3.4. Adaptation Methods

We evaluate two adaptation regimes: prompt-based extraction and supervised fine-tuning, applied consistently across models and extraction strategies.

For prompt-based extraction, we consider both zero-shot and few-shot settings. In the zero-shot configuration, used for both pipelined and joint extraction, prompts explicitly specify the IE task, the expected JSON output format, and the complete set of entity and relation types defined in the SCIERC schema. For each entity and relation type, a short natural language definition is provided in the prompt to guide the model's interpretation of the schema and reduce ambiguity in scientific terminology. This setup enables schema-aware extraction without relying on task-specific examples.

In addition, for the joint extraction strategy, we evaluate a few-shot prompting setup using three examples (3-shot) sampled from the training split of the dataset. These examples demonstrate the target structured output for complete entity–relation triples (plus entity types), reinforcing correct schema usage. The prompt templates used across the experiments are shown in Section A.1 of the appendix.

For fine-tuning, we adapt the models using QLoRA (Dettmers et al., 2023), enabling parameter-efficient supervised training by freezing the 4-bit quantized base model weights and training lightweight LoRA adapters. Fine-tuning is performed exclusively for the joint extraction formulation, where the model is trained to generate structured outputs representing entities, their types, and relations in a single step. Training instances follow the same input–output structure used during joint extraction prompting, ensuring consistency between training and inference. We fine-tune the models for 5 epochs using a learning rate of 2e-4, a LoRA rank of 8, and a batch size of 1, using BF16 precision throughout.

3.5. Evaluation Metrics

We evaluate the performance of our methods using precision, recall, and F1-score, leveraging the gold-standard entity and relation annotations provided by the dataset. Metrics are reported using macro averaging, with NER scores averaged per entity type and RE scores averaged per relation type, allowing for fine-grained analysis across different categories of scientific information.

Our primary evaluation criterion is exact match, where predicted entity spans, entity types, and relations must exactly match the gold annotations. This strict evaluation protocol follows the standard practice adopted in prior work on SCIERC, enabling direct comparability with existing results (Eberts and Ulges, 2020; Zhong and Chen, 2021). Moreover, exact matching ensures a high level of precision and directly reflects the requirements of downstream applications, where deviations in extraction can propagate errors and negatively impact subsequent processing or analysis (Dagdelen et al., 2024).

4. Results and Analysis

This section presents the experimental results across models, extraction paradigms, and adaptation strategies. We analyse performance and examine how modelling and adaptation choices influence both NER and RE. In addition to quantitative results, we provide an error analysis to better identify recurring failure patterns and their implications for SciKG construction.

4.1. Overall Performance

Table 4 reports Macro-F1 scores for both NER and RE across models, extraction strategies, and adaptation settings.

Model	Pipeline		Joint 0-Shot		Joint 3-Shot		Fine-Tune	
	NER	RE	NER	RE	NER	RE	NER	RE
AMALIA	0.11	0.01	0.11	0.00	0.21	0.04	**0.44**	0.16
Llama-3.1	0.17	0.01	0.13	0.02	0.18	0.02	0.40	0.16
Qwen3-8b	0.25	0.06	0.21	0.03	0.21	0.04	0.42	**0.23**
Gemma-3-12b-it	**0.31**	**0.07**	**0.26**	**0.06**	**0.31**	**0.09**	0.39	0.14
Boto-9b-it	0.13	0.01	0.14	0.03	0.20	0.02	0.40	0.14
Gervásio 8B	0.16	0.01	0.11	0.02	0.18	0.02	0.37	0.13
EuroLLM 9b	0.11	0.00	0.10	0.01	0.13	0.02	0.42	0.18

Table 4: Comparison of model performance across NER and RE Tasks.

Overall, performance remains relatively low for both tasks. Unsurprisingly, RE is consistently more challenging than NER, regardless of model or strategy. This reflects the intrinsic difficulty of structured scientific IE under an exact-match evaluation regime, where our methods exhibited challenges in adapting effectively to the task.

It is noticeable that the adaptation strategy has a substantially larger impact than the model choice. Across models, fine-tuning leads to marked improvements in both tasks. NER performance increases from roughly 0.10–0.31 in zero-shot settings to approximately 0.37–0.44 after fine-tuning. RE exhibits an even stronger relative effect: while via prompting the performance is often close to zero, fine-tuning raises it to the 0.13–0.23 range. This consistent behaviour across models indicates that structured scientific extraction in Portuguese cannot be reliably achieved through prompting alone at this parameter scale.

Differences between models are relatively moderate. After fine-tuning, most models converge to similar levels of NER performance, and RE scores vary within a relatively narrow band. Notably the AMALIA model achieves competitive performance and attains the highest NER score among all evaluated models, suggesting that supervised task alignment may play a more decisive role than the Portuguese pretraining origin, in this specific setting. In contrast, under prompting-based evaluation, the Gemma3 model consistently delivers the strongest results. Interestingly, it also operates at a slightly larger parameter scale than the other models but given differences in architecture, pretraining data, and instruction tuning, performance gains cannot be attributed solely to model size. Qwen3 also demonstrates stable and competitive performance across all settings, which aligns with its strong positioning on the Portuguese leaderboard of Hugging Face.

4.2. Impact of Extraction Strategy and Adaptation

The pipeline formulation isolates entity detection, potentially improving stability by removing the need to jointly generate relational structure (Diaz-Garcia and Lopez, 2025; Zhong and Chen, 2021). In contrast, joint approaches may benefit from shared representations, where relational context helps refine entity predictions (Yan et al., 2022; Santosh et al., 2021). In practice, some models show slight advantages of the pipeline over joint zero-shot (e.g., Gemma reaches 0.31 NER in the pipeline setup versus 0.26 in joint zero-shot, and Qwen achieves 0.25 versus 0.21), but results are comparable to, or slightly lower than, joint 3-shot performance. The reduction in task complexity from isolating entity detection is not clearly reflected in the results.

For RE, joint extraction is often motivated by its ability to avoid error propagation from a separate NER module (Yan et al., 2022). However, our results do not allow for strong conclusions in this regard, as RE performance under prompting remains extremely low in both extraction settings.

Despite noise potentially introduced by the additional context (examples), few-shot prompting was expected to improve instruction adherence and task alignment. In our experiments, it leads to improvements over zero-shot mainly for NER. For instance, AMALIA increases from 0.11 to 0.21 macro-F1, and Boto from 0.13 to 0.20. Other models exhibit negligible change, and RE performance remains consistently low. This indicates that few-shot examples partially support entity recognition but are insufficient for robust RE.

Fine-tuning produces a qualitatively different behaviour. As previously mentioned, both NER and RE improve substantially across all models and, crucially, performance variance narrows. The relative gains for RE are particularly pronounced: Qwen3 increases from 0.06 (pipeline) to 0.23 after fine-tuning, and AMALIA from 0.01 to 0.16. These results indicate that supervised adaptation is not only beneficial but especially critical for learning coordinated entity–relation structures at this model scale.

Across English SciERC evaluations conducted under the same settings as ours, prior work shows that generative LLMs under prompting achieve relatively modest results, roughly 35–42% F1 for NER and about 21% for RE with DeepSeek-V3 (Zhao et al., 2025). Fine-tuning GPT-4o improves NER to around 60% and RE to approximately 23%, although RE performance remains comparatively weak (Tran et al., 2025). In contrast, earlier supervised span-based and dependency-aware transformer encoders report substantially stronger results (Eberts and Ulges, 2020; Joshi and Rekik, 2025), typically exceeding 70% F1 on NER and reaching 50–60% F1 on RE, underscoring the continued advantage of structurally explicit architectures for scientific IE.

Overall, the findings indicate that prompting-based strategies, under the considered configu-

rations, are insufficient for reliable scientific IE in Portuguese. Supervised fine-tuning, even if limited, is not merely advantageous but functionally necessary to achieve non-trivial extraction performance and to stabilise cross-model variability. This suggests that generative mid-scale LLMs are not yet fully adequate for high-quality structured scientific extraction and that substantial room for improvement remains.

4.3. Per-type Performance Analysis

Table 5 presents per-type F1 scores for the two best overall models (Qwen3 and Gemma3) across extraction strategies and adaptation settings, allowing a more fine-grained analysis of task behaviour.

Model	Pipeline		Joint 0-Shot		Joint 3-Shot		Fine-Tune	
	Q	G	Q	G	Q	G	Q	G
Entity Types								
Task	0.31	0.38	0.24	0.37	0.23	0.36	0.40	0.40
Method	0.40	0.37	0.36	0.42	0.41	0.46	0.62	0.53
Eval Metric	0.32	0.41	0.23	0.25	0.23	0.30	0.43	0.34
Material	0.31	0.35	0.31	0.29	0.26	0.32	0.42	0.46
Other Sci Terms	0.10	0.31	0.14	0.21	0.14	0.24	0.32	0.32
Generic	0.04	0.05	0.00	0.03	0.02	0.21	0.38	0.30
Relation Types								
Used-for	0.08	0.09	0.02	0.17	0.05	0.17	0.20	0.19
Feature-of	0.00	0.02	0.00	0.02	0.00	0.02	0.06	0.00
Hyponym-of	0.17	0.14	0.00	0.12	0.04	0.12	0.27	0.15
Part-of	0.03	0.05	0.05	0.05	0.00	0.05	0.17	0.03
Conjunction	0.01	0.05	0.03	0.09	0.00	0.09	0.28	0.23
Evaluate-for	0.00	0.00	0.01	0.08	0.07	0.08	0.24	0.19
Compare	0.14	0.11	0.09	0.11	0.12	0.09	0.36	0.20

Table 5: Per-type F1 scores across entity and relation types for Qwen3-8B (Q) and Gemma-3-12b-it (G).

Entity performance is clearly uneven across categories. *Method* consistently emerges as the strongest performing entity type, particularly after fine-tuning, where Qwen3 reaches 0.62 and Gemma3 0.53. This suggests that method mentions are comparatively well-delimited and semantically distinctive in scientific text. *Task*, *Eval Metric*, and *Material* achieve moderate performance and show consistent results across settings. In contrast, *Other Scientific Terms* and especially *Generic* entities are substantially more challenging. In the prompting settings, *Generic* entities are nearly absent (often close to zero), and although fine-tuning improves performance, they remain among the weakest categories. This pattern likely reflects both greater heterogeneity and annotation ambiguity: broader or less well-defined categories provide fewer lexical cues, making boundary detection and type assignment more difficult.

RE exhibits stronger sparsity and imbalance effects. In the prompting configurations, most relation types remain near zero for both models. Fine-tuning leads to substantial gains, but performance remains uneven. *Compare* and *Conjunction* show the largest improvements after fine-tuning (e.g., Qwen3 reaches 0.36 and 0.28, respectively), proving to be comparatively easier to capture once the model is aligned to the task. Conversely, *Feature-of* and *Part-of* remain difficult even after fine-tuning, often yielding near-zero or very low scores. These relations encode subtler semantic dependencies and may be more sensitive to contextual interpretation.

While both models benefit from fine-tuning, Qwen3 shows stronger gains and achieves higher peak scores across most entity and relation types. Gemma3, however, demonstrates more stable behaviour in prompting settings, especially in zero-shot joint extraction. This reinforces earlier observations: Gemma3 appears slightly more robust in instruction-following scenarios, whereas Qwen3 adapts particularly well under supervised alignment.

4.4. Error Analysis

To better identify the limitations behind the observed performance, we analyse the model predictions. Table 6 presents representative examples of some of the error categories described below.

Model Response	Expected Response
(“distribution”, “used-for”, “play organizer”)	(“this distribution”, “used-for”, “play organizer”)
(“dictionary lookup phase”, “conjunction”, “application of rules”)	(“dictionary lookup”, “conjunction”, “application of rules”)

(a) Boundary variation error examples in extracted triples.

Model Response	Expected Response
(“high-dimensional space”, “material”)	(“high-dimensional space”, “other scientific terms”)
(’density estimation’, ’method’)	(’density estimation’, ’task’)
(“spoken dialogue system”, “task”)	(“spoken dialogue system”, “method”)

(b) Entity type error examples.

Model Response	Expected Response
(“reservoir models”, “feature-of”, “inner hair cell synapse”)	(“reservoir models”, “used-for”, “inner hair cell synapse”)
(“error rate”, “feature-of”, “CRF model”)	(“error rate”, “evaluate-for”, “CRF model”)

(c) Relation type error examples.

Table 6: Comparative analysis of some of the most common errors encountered during the experiments.

The most common error in NER is the omission of gold entities. A large portion of annotated entities are simply not generated. Since RE depends

on the correct identification of both entities in a pair, these omissions propagate downstream, significantly increasing the difficulty of producing valid relational triples. As a consequence, most triple annotations are missed not only due to incorrect relation classification, but mostly because one or both entities are absent from the prediction.

Another relevant issue concerns deviations from the expected output format. In several runs, models did not strictly follow the prompt instructions, generating outputs that did not conform to the required JSON structure and therefore being penalised during evaluation. This problem occurs mainly in the pipeline and zero-shot joint formulations. For instance, in the pipeline setting, AMALIA presents 13% of entries not following the expected format, while Boto reaches 39%. The use of 3-shot prompting and the fine-tuning substantially improves structural compliance, bringing adherence to near 100%.

Boundary and lexical variation errors also contribute to performance degradation. Predicted spans are sometimes slightly longer or shorter than the gold annotation, or contain minimal lexical differences. Under exact match evaluation, even small variations, despite preserving meaning, are counted as incorrect, negatively affecting extraction scores.

Entity type confusions are another recurring NER error. In these cases, the span is correctly identified but assigned the wrong type. A frequent confusion occurs between *Method* and *Task*, which often appear in similar contexts in scientific text. Additionally, spans belonging to *Generic* and sometimes *Other scientific terms* are forced into one of the other most specific categories, suggesting imperfect schema internalisation.

Finally, relation prediction errors occur, although in smaller numbers compared to entity omissions. Some relations show systematic confusion, particularly *Feature-of* versus *Used-for*, and *Compare* versus *Conjunction*. This suggests that there is high semantic overlap between these categories and subtle syntactic cues that distinguish them in scientific discourse, making it challenging for the models to disambiguate.

5. Implications for Scientific Knowledge Graph Construction

Scientific IE plays a central role in the automatic construction of SciKGs. The reliability of downstream applications depends directly on the precision and structural consistency of the extracted entities and relations. In this context, the results obtained in this work fall short of the level of robustness required for robust SciKG construction.

The prompting-based configurations proved insufficient for reliable graph generation. Even after fine-tuning, while improvements were substantial relative to prompting settings, RE performance remains moderate, and entity-level errors persist. Given that KGs propagate extraction errors directly into structured representations, these limitations would likely translate into incomplete or noisy graphs, thereby compromising downstream usability. Rather than representing a dead end, these findings highlight several research directions for improving scientific IE in Portuguese.

One evident limitation of the present study is the moderate parameter scale of the evaluated models (8–12B). While this choice reduced computational barriers for inference and fine-tuning, it likely constrained representational capacity and reasoning depth. Exploring larger and more robust generative models may yield stronger and more stable outputs, particularly for RE.

In parallel, traditional supervised IE approaches, such as span-based models, and domain-adapted encoder-based transformers have historically demonstrated competitive performance in scientific domains and may offer higher precision under strict evaluation regimes (Eberts and Ulges, 2020; Joshi and Rekik, 2025). A hybrid comparison between generative and encoder-based paradigms would therefore be a valuable extension.

Data availability is another critical factor. Expanding annotated resources or constructing additional domain-aligned Portuguese corpora can enable more extensive supervised training, potentially improving both entity boundary detection and relation classification. Given the relatively specialised and technical nature of Portuguese scientific text, increased annotation scale may be particularly impactful. Whenever possible, such data should originate from native Portuguese scientific sources rather than translated material, as translation artefacts can introduce inconsistencies. Resources such as CorEGe-PT (Kuhn et al., 2026), a Portuguese corpus of academic texts compiled from the Estudo Geral repository, exemplify the type of native scientific material

Methodologically, further refinements in our methods are also possible. The error analysis revealed weaknesses that prompt engineering could try to address, for example through more specialized schema reinforcement or intermediate reasoning verification steps. On the fine-tuning side, systematic experimentation with hyperparameter configurations, instruction formatting, or multi-task objectives may yield additional gains.

Finally, evaluation methodology deserves consideration. The strict exact-match criterion, while standard in IE benchmarking, is highly penalising in cases of minimal lexical variation or semantically

equivalent span differences. Alternative or complementary evaluation schemes, such as string-similarity-based metrics, could provide a more nuanced assessment of extraction quality, particularly when the intended application tolerates minor lexical deviations.

Overall, the present results indicate that reliable SciKG construction in Portuguese remains an open challenge. However, the findings provide a diagnostic foundation that clarifies current bottlenecks and outlines concrete pathways for advancing IE in this domain.

6. Conclusion

Automatic extraction from rapidly expanding scientific literature is crucial for building SciKGs that support semantic search, question answering, and large-scale reasoning. Despite recent progress in LLMs with great demonstrations on text processing, their reliability for structured scientific IE in Portuguese remains underexplored. This work provides a systematic evaluation of low- to mid-scale generative LLMs for Portuguese scientific NER and RE, comparing pipeline and joint formulations under zero-shot, few-shot, and fine-tuned settings using a curated European Portuguese translation of the SCIERC dataset.

Three main findings emerge. First, the adaptation strategy has a substantially greater impact than model choice at this scale: prompting approaches yield unstable performance, especially for RE, whereas fine-tuning consistently improves both tasks. Second, RE remains markedly more challenging than entity recognition due to the compositional and boundary-sensitive nature of structured triple prediction. Third, model differences are pronounced under prompting but largely diminish after supervised adaptation, highlighting the importance of task alignment over architectural variation.

From a SciKG construction perspective, our results indicate that prompting alone is insufficient for reliable graph generation in scientific domains. Even supervised configurations, while promising, would require further optimisation before being suitable for downstream reasoning applications. Our error analysis, revealing entity omissions, boundary mismatches, type confusions, and relation misclassifications, identifies key bottlenecks and motivates future work on larger models, expanded annotated resources, alternative encoder-based approaches, and refined adaptation strategies. Overall, this study establishes a grounded empirical baseline for Portuguese scientific IE and outlines a concrete path for further exploration toward SciKG construction in Portuguese settings.

7. Acknowledgements

This work was partially supported by the AMALIA project, funded by FCT/IP in the context of measure RE-C05-i08 of the Portuguese Recovery and Resilience Program;

This work was also supported by the Portuguese Recovery and Resilience Plan (PRR), through project C645008882-00000055 – Center for Responsible AI; produced as part of the N-GenERP project with reference COMPETE2030-FEDER-02219400 (operation no. 21343) supported by the European Regional Development Fund (FEDER) through the Innovation and Digital Transition Programme (COMPETE 2030) of Portugal 2030 and the European Union; and Project LUMEN, funded by the European Union under Grant Agreement no. 101187940; and supported by FCT – Foundation for Science and Technology, I.P., under the projects UIDB/00326/2025 and UIDP/00326/2025;

8. Bibliographical References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, et al. 2026. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, pages 1–7.

Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, et al. 2023. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports*, 13(1):7240.

Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria Esther Vidal. 2018. Towards a knowledge graph for science. In *Proceedings of the 8th international conference on web intelligence, mining and semantics*, pages 1–6.

Mykhailo Bondarenko, Sviatoslav Lushnei, Yurii Paniv, Oleksii Molchanovsky, Mariana Romanyshyn, Yurii Filipchuk, and Artur Kiulian. 2025. Sovereign large language models: Advantages, strategy and regulations. *arXiv preprint arXiv:2503.04745*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bruno Cabral, Daniela Claro, and Marlo Souza. 2024. Exploring open information extraction for Portuguese using large language models. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 127–136, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. *Advances in neural information processing systems*, 36:10088–10115.

Jose A Diaz-Garcia and Julio Amador Diaz Lopez. 2025. A survey on cutting-edge relation extraction techniques based on language models. *Artificial Intelligence Review*, 58(9):287.

Keyan Ding, Zhihui Zhu, Yuqi Tang, Kehua Feng, Xiang Zhuang, Hongwei Wang, Yi Yang, Huifang Du, Zhangkai Ni, Shiqi Wang, Xiaohui Fan, Huabin Xing, Lei Bai, Qi Liu, Haofen Wang, Qiang Zhang, and Huajun Chen. 2025. Bridging data and discovery: A survey on knowledge graphs in ai for science.

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

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mark A Hanson, Pablo Gómez Barreiro, Paolo Crosetto, and Dan Brockington. 2024. The strain on scientific publishing. *Quantitative Science Studies*, 5(4):823–843.

Zhi Hong, Logan Ward, Kyle Chard, Ben Blaiszik, and Ian Foster. 2021. Challenges and advances in information extraction from scientific literature: a review. *Jom*, 73(11):3383–3400.

Devvrat Joshi and Islem Rekik. 2025. Dependency parsing-based syntactic enhancement of relation extraction in scientific texts. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24888–24897.

Tanara Zingano Kuhn, José Matos, Bruno Neves, Daniela Pereira, Elisabete Cação, Ivo Simões, Jacinto Estima, Delfim Leão, and Hugo Gonçalo Oliveira. 2026. CorEGe-PT: Compiling a Large Corpus of Academic Texts in Portuguese. In *Proceedings of 15th Language Resources and Evaluation Conference*, LREC 2026, page Accepted. ELRA.

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.

Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. 2025. Large language models meet knowledge graphs for question answering: Synthesis and opportunities. *arXiv preprint arXiv:2505.20099*.

Sérgio Nunes, Alípio Mario Jorge, Evelin Amorim, Hugo Sousa, António Leal, Purificação Moura Silvano, Inês Cantante, and Ricardo Campos. 2024. Text2story lusa: A dataset for narrative analysis in european portuguese news articles. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15773–15782.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems*, 44(2):1–52.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial intelligence review*, 56(11):13071–13102.

Miguel Moura Ramos, Duarte M Alves, Hippolyte Gisserot-Boukhlef, João Alves, Pedro Henrique Martins, Patrick Fernandes, José Pombal, Nuno M Guerreiro, Ricardo Rei, Nicolas Boizard, et al. 2026. Eurollm-22b: Technical report. *arXiv preprint arXiv:2602.05879*.

Rodrigo Santos, João Silva, Luís Gomes, João Rodrigues, and António Branco. 2024. Advancing generative ai for portuguese with open decoder gervásio pt. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Underresourced Languages@ LREC-COLING 2024*, pages 16–26.

Tokala Yaswanth Sri Sai Santosh, Prantika Chakraborty, Sudakshina Dutta, Debarshi Kumar Sanyal, and Partha Pratim Das. 2021. Joint entity and relation extraction from scientific documents: Role of linguistic information and entity types. *EEKE@ JCDL*, 21:15–19.

Gabriel Silva, Mário Rodrigues, António Teixeira, and Marlene Amorim. 2024. Advancing open information extraction for portuguese by leveraging graph structures and large language models. In *Proc. IberSPEECH 2024*, pages 61–65.

Afonso Simplício, Gonçalo Vinagre, Miguel Moura Ramos, Diogo Tavares, Rafael Ferreira, Giuseppe Attanasio, Duarte M. Alves, Inês Calvo, Inês Vieira, Rui Guerra, James Furtado, Beatriz Canaverde, Iago Paulo, Vasco Ramos, Diogo Glória-Silva, Miguel Faria, Marcos Treviso, Daniel Gomes, Pedro Gomes, David Semedo, André Martins, and João Magalhães. 2026. AMALIA technical report: A fully open source large language model for european portuguese.

Hugo Sousa, Alipio Mario Jorge, Arian Pasquali, Catarina Santos, and Mario Lopes. 2023. A biomedical entity extraction pipeline for oncology health records in portuguese. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 950–956.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Trang Tran, Trung Hoang Le, Huiping Cao, and Tran Cao Son. 2025. An LLM+ ASP workflow for joint entity-relation extraction. *arXiv preprint arXiv:2508.12611*.

Shilpa Verma, Rajesh Bhatia, Sandeep Harit, and Sanjay Batish. 2023. Scholarly knowledge graphs through structuring scholarly communication: a review. *Complex & intelligent systems*, 9(1):1059–1095.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Yuemei Xu, Ling Hu, Jiayi Zhao, Zihan Qiu, Kexin Xu, Yuqi Ye, and Hanwen Gu. 2025. A survey on multilingual large language models: Corpora, alignment, and bias. *Frontiers of Computer Science*, 19(11):1911362.

Zhaohui Yan, Zixia Jia, and Kewei Tu. 2022. An empirical study of pipeline vs. joint approaches to entity and relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 437–443.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Rui Yang, Boming Yang, Xinjie Zhao, Fan Gao, Aosong Feng, Sixun Ouyang, Moritz Blum, Tianwei She, Yuang Jiang, Freddy Lecue, et al. 2025b. Graphusion: A rag framework for scientific knowledge graph construction with a global perspective. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2579–2588.

Zhenyao Yang, Sha Yuan, Zhou Shao, Wenfa Li, and Runzhou Liu. 2025c. A review on synergizing knowledge graphs and large language models. *Computing*, 107(6):143.

Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. 2024. Scier: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13083–13100.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Junnan Dong, et al. 2025a. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. 2025b. A survey of generative information extraction. In *Proceedings of the 31st International conference on computational linguistics*, pages 4840–4870.

Dong Zhao, Yadong Wang, Xiang Chen, Chenxi Wang, Hongliang Dai, Chuanxing Geng, Shengzhong Zhang, Shaoyuan Li, and Sheng-Jun Huang. 2025. Reflect then learn: Active prompting for information extraction guided by introspective confusion. *arXiv preprint arXiv:2508.10036*.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 50–61.

Zhihui Zhu, Yuqi Tang, Qiang Zhang, and Keyan Ding. 2025. Synergizing large language models and knowledge graphs in science: A survey. In *NeurIPS 2025 AI for Science Workshop*.

A. Appendix

A.1. Prompt Templates Used

Figure 1 presents a segment containing the entity type definitions used across prompts, while Figure 2 presents the relation type definitions.

Figure 3 shows the pipeline NER prompt template, while Figure 4 corresponds to the pipeline RE prompt template. Lastly, Figure 5 shows the joint prompt template, which has the particularity of including three examples in the few-shot setting. Each individual example consists of an abstract followed by an array of JSON objects, one for each triple, each containing a subject span and type, an object span and type, and a relation type.

```
- Tarefa - Aplicações, problemas a re-
solver, sistemas a construir, fenómenos
ou tópicos em investigação.
- Método - Métodos, abordagens, técnicas,
modelos, sistemas, algoritmos, compo-
nentes ou ferramentas utilizados.
- Métrica - Métricas de avaliação, medi-
das ou entidades que exprimem a qualidade
de um sistema/método.
- Material - Dados, conjuntos de dados,
recursos, corpus, base de conhecimento.
- Outros Termos Científicos - Expressões
que são termos científicos mas que não se
enquadram nas categorias anteriores.
- Genérico - Termos gerais ou pronomes
que podem referir uma entidade mas que
não são informativos por si só, sendo
frequentemente usados como palavras de
ligação.
```

Figure 1: Entity type definitions.

```
- Usado-para: A relação entre um
Método/Ferramenta e a Tarefa/Propósito
para o qual é utilizado.
- Característica-de: Relaciona uma pro-
priedade, atributo ou qualidade com o
sujeito que a possui.
- Hipónimo-de: Uma entidade é um tipo
mais específico de outra entidade mais
geral.
- Parte-de: Uma entidade constitui uma
parte integrante de um todo maior.
- Avaliado-para: Indica que uma entidade
é usada para avaliar outra entidade.
- Comparado-com: Usado para comparar
explicitamente dois modelos, métodos ou
resultados.
- Conjunção: Relação simétrica entre
duas entidades com função ou papel semel-
hante.
```

Figure 2: Relation type definitions.

```
A tua tarefa é identificar e classificar
**termos científicos relevantes** no
texto, de acordo com o seguinte esquema
simplificado de tipos de entidade.

**TIPOS DE ENTIDADE:**
{entity_types_definitions}

**INSTRUÇÕES:**
- Identifica todas os termos científicos
explícitos no texto.
- Não uses frases completas como enti-
dade.
- Mantém o output em formato estruturado
JSON, sem explicações ou texto extra
- Se não identificares termos científicos,
devolve um array vazio.

**FORMATO DE OUTPUT:**
Retorna um array JSON de entidades, onde
cada entrada contém:
- "nome": texto da entidade
- "tipo": categoria da entidade (Tarefa,
Método, Métrica de Avaliação, Material,
Outros Termos Científicos, Genérico)

**TEXTO:**
{text_input}
```

Figure 3: Pipeline NER prompt template.

```
A tua tarefa é identificar relações semân-
ticas entre as entidades fornecidas,
baseando-te EXCLUSIVAMENTE no texto cien-
tífico apresentado.

**PREDICADOS (RELAÇÕES) PERMITIDOS:**
{relation_types_definitions}

**INSTRUÇÕES:**
- O "sujeito" e o "objeto" DEVEM ser ex-
atamente os nomes listados em "ENTIDADES
FORNECIDAS".  Não crie novas entidades.
- O "Predicado" DEVE ser um dos "PREDICA-
DOS PERMITIDOS".
- Se não houver relação clara entre duas
das entidades no texto, ignora.
- Mantém o output em formato estruturado
JSON SEM EXPLICAÇÕES OU TEXTO EXTRA.

**FORMATO DE OUTPUT:**
Retorna APENAS uma lista JSON de triplos.
Cada entrada deve conter:
- "sujeito":  termo científico ou enti-
dade principal
- "predicado":  relação entre o sujeito e
o objeto
- "objeto":  termo científico ou entidade
associada

**TEXTO**:
{text_input}
**ENTIDADES FORNECIDAS**:
{entities_input}
```

Figure 4: Pipeline RE prompt template.

```
És um assistente de IA que extrai conhec-
imento estruturado de textos científicos.
Extrai entidades e relações como triplos
e devolve-os em formato JSON.

**TIPOS DE ENTIDADE:**
{entity_types_definitions}

**TIPOS DE RELAÇÃO:**
{relation_types_definitions}

**EXEMPLOS:**
{3shot_examples}

**TEXTO:**
{text_input}

**FORMATO DE OUTPUT:**
Retorne um array JSON de triplos.  Cada
triplo deve conter:
- "sujeito":  texto da entidade
- "tipo_sujeito":  tipo do sujeito
- "relação":  tipo de relação
- "objeto":  texto da entidade
- "tipo_objeto":  tipo do objeto
```

Figure 5: Joint prompt template.

From Slides to Chatbots: Enhancing Large Language Models with University Course Materials

Tu Anh Dinh*, Philipp Nicolas Schumacher*, Jan Niehues
Karlsruhe Institute of Technology, Germany
tu.dinh@kit.edu, philipp.schumacher@gmx.de, jan.niehues@kit.edu

Abstract

Large Language Models (LLMs) have advanced rapidly in recent years. One application of LLMs is to support student learning in educational settings. However, prior work has shown that LLMs still struggle to answer questions accurately within university-level computer science courses. In this work, we investigate how incorporating university course materials can enhance LLM performance in this setting. A key challenge lies in leveraging diverse course materials such as lecture slides and transcripts, which differ substantially from typical textual corpora: slides also contain visual elements like images and formulas, while transcripts contain spoken, less structured language. We compare two strategies, Retrieval-Augmented Generation (RAG) and Continual Pre-Training (CPT), to extend LLMs with course-specific knowledge. For lecture slides, we further explore a multi-modal RAG approach, where we present the retrieved content to the generator in image form. Our experiments reveal that, given the relatively small size of university course materials, RAG is more effective and efficient than CPT. Moreover, incorporating slides as images in the multi-modal setting significantly improves performance over text-only retrieval. These findings highlight practical strategies for developing AI assistants that better support learning and teaching, and we hope they inspire similar efforts in other educational contexts.

Keywords: Information Extraction, Information Retrieval, Language Modelling, Training, Fine-tuning, Adaptation, Multimedia Document Processing

1. Introduction

In recent years, the development of Large Language Models (LLMs) has advanced rapidly (Brown et al., 2020; Grattafiori et al., 2024; Yang et al., 2024). There has been growing interest in using LLMs to support teaching and learning, for example, by developing chatbots that can assist students in understanding complex course content. However, despite their general knowledge and reasoning abilities, current LLMs often struggle to provide accurate and contextually grounded answers to questions within university-level computer science courses. Benchmarks such as SciEx (Dinh et al., 2024), which evaluate models on real exam questions, show that while LLMs perform well in topics like Deep Learning and Neural Networks, they remain weak in others, such as Human–Computer Interaction.

One potential reason for these gaps is that LLMs may not have been exposed to the course-specific materials used in teaching. The materials, such as lecture slides and transcripts, tend to be different from the types of text LLMs are typically trained on. Although relatively small in scale, they contain highly specialized knowledge. Moreover, they have unique characteristics: slides include visual content such as images, formulas, and structured layouts, while transcripts contain spontaneous, conversational explanations. These properties make course materials both valuable and challenging sources for building educational chatbots.

In this work, we explore how incorporating course materials can improve LLM-based assistants in educational contexts. We focus on two strategies: (1) Retrieval-Augmented Generation (RAG), which dynamically retrieves relevant content from course materials to assist answer generation, and (2) Continual Pre-Training (CPT), which further trains the base model on the course materials. For lecture slides, we additionally investigate a multi-modal RAG approach, where visual information is preserved by presenting retrieved slides as images to the generator.

Our experiments, conducted on computer science course materials and evaluated on the SciEx benchmark, show that RAG is generally more effective and efficient than CPT, given the small dataset size of the course materials. Furthermore, leveraging slides in image form leads to notable gains over text-only retrieval. These results highlight practical strategies for constructing chatbots that better understand and communicate course-specific knowledge.

2. Related Work

Scientific Evaluation Evaluating the capabilities of Large Language Models (LLMs) is crucial for understanding their potential as educational AI assistants. To evaluate the scientific capabilities of LLMs, several public benchmarks have been developed, such as SciQ (Welbl et al., 2017), Sci-

*Co-first authors.

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 168–179
12 May 2026.

enceQA (Lu et al., 2022), and M3Exam (Zhang et al., 2024). However, the questions in these benchmarks are multiple-choice, which do not reflect the open-ended interactions expected from an educational assistant. In real learning scenarios, students are likely to prefer free-form, natural-language explanations, rather than simple answer selection. The SciEx benchmark (Dinh et al., 2024) addresses this limitation by evaluating LLMs on university-level computer science exam questions, including a broader range of question types beyond multiple-choice. SciEx has shown that LLMs still underperform in certain topics. Since each SciEx exam corresponds to a real university course, the associated course materials, i.e., lecture slides and transcripts, offer a promising resource for improving the LLMs. In this work, we leverage such materials to enhance LLMs' ability to function as educational assistants, and use SciEx as our evaluation benchmark.

Retrieval-Augmented Generation Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) combines external knowledge retrieval with generative language models to provide informed responses. Recent work has focused on adapting RAG to scientifics domains. For example, Gokdemir et al. (2025) proposed HiPerRAG, which scales retrieval to millions of scientific articles. Shi et al. (2025) introduced Hypercube-RAG, a multidimensional retrieval method that better captures the structure of scientific knowledge. Domain-specific RAG benchmarks such as ChemRAG-Bench (Zhong et al., 2025) demonstrate substantial gains in performance on chemistry-related tasks. Beyond text retrieval, multimodal RAG systems such as Matsumoto et al. (2024) integrate knowledge graphs to enhance reasoning in biomedical contexts. However, most of these works rely on large-scale, text-only corpora, differing to our work, which investigates RAG using the small, specialized and partially multimodal source of university course materials.

Continual Pre-Training for Domain Adaptation Continual Pre-Training (CPT) is an effective method for domain adaptation by continuing the pre-training of a general LLM on domain-specific data, as shown by Gururangan et al. (2020). Subsequent studies have examined strategies to optimize CPT and mitigate catastrophic forgetting. Que et al. (2024) proposed the D-CPT Law to balance the mixture of domain and general corpora. Yıldız et al. (2024) analyzed learning dynamics during CPT, indicating the importance of monitoring domain and general performance for early stopping. Ibrahim et al. (2024) showed that a good selection of learning rate re-warming, learning rate re-decaying, and replay of previous data could be sufficient to match the performance of fully re-training from scratch. CPT has been successfully applied in domains such as biomedical (Jin et al., 2023) and financial (Xie et al., 2024). Differing to previous works, our work investigates CPT on very limited but specialized data from university course materials, which poses additional risks of overfitting and forgetting.

An important consideration is how and at which stage of model training CPT should be performed to preserve the model's instruction-following capability. Jindal et al. (2024) compared two strategies: (1) applying continual pre-training directly on an instruction-tuned model, and (2) continually pre-training the base model followed by instruction fine-tuning. Their results show that the first approach leads to catastrophic forgetting of instruction-following abilities, whereas the second approach better preserves them. However, instruction fine-tuning is computationally expensive, requiring large amounts of hand-annotated instruction data. Therefore, the authors proposed an alternative: adding *instruction residuals* to the base LLM after CPT to restore the instruction-following capability. We employ this approach in our work to avoid the expensive instruction fine-tuning.

3. Gathering Course Materials Data

We obtain course materials corresponding to the university exams used in the SciEx benchmark (Dinh et al., 2024) by contacting the dataset creators. These materials provide an authentic educational context from which the exam questions were drawn, allowing us to more realistically evaluate LLMs as potential assistants for university-level education.

We focus on two types of materials: lecture slides and lecture transcripts. These resources are widely available in most university courses and thus represent a practical solution for building educational chatbots. At the same time, they differ substantially in structure and modality. Slides are concise, visually organized, and often include non-textual information such as images and formulas. Transcripts, in contrast, contains spoken language, offering more explanations and contextual details. We aim to explore how LLMs can leverage diverse course material formats to better understand and generate course-specific explanations.

Slides We collected the slides as PDF files, where each PDF page corresponds to a separate slide in the presentation. For experiments with text-only LLMs, we extract the text from the PDFs. For experiments with visual-aided LLMs, we convert the PDF pages to images.

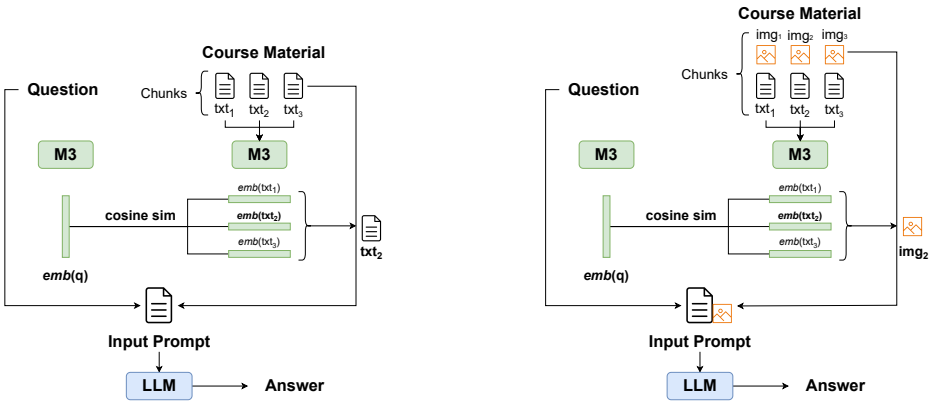


Figure 1: **Retrieval-Augmented Generation (RAG) workflow.** The retriever encodes slide text and lecture transcripts into a text-based vector database. At inference time, relevant text chunks or slides are retrieved. Left-hand side: **text-only RAG**. Right-hand side: **multimodal RAG**: when using a visually capable LLM, retrieved slides are passed as *images* to the generator, preserving visual information.

Transcripts We extract the textual transcriptions from the audios of the lecture recordings using the Lecture Translator framework (Huber et al., 2023), which employs *Whisper Large V2* (Radford et al., 2023) paired with SHAS segmentation (Tsiamas et al., 2022). We clean up the transcriptions afterward by manually checking and removing hallucinated sentences, i.e., sentences repeatedly presented in the transcriptions but not presented in the audio.

As the transcripts are from spoken language, they are non-fluent, which might be unsuitable for direct usage with LLMs. Therefore, we polish the transcriptions using the 4-bit-quantized Qwen2.5-Instruct 72B model. We ask the model to make the text more fluent and remove irrelevant content. We split the transcriptions into chunks of five sentences, and include the previously generated polished chunk in each input prompt to ensure smooth transitions. Finally, the resulting polished transcripts are manually processed to remove LLM-generated sentences that are irrelevant to the actual transcript. The detailed prompt is in Appendix A.

Data Statistics The final dataset includes course materials in the form of lecture slides and transcriptions in English and German. The dataset covers six topics: Algorithms (ALGO), Databases (DBS), Deep Learning and Neural Networks (DLNN), Human–Computer Interaction (HCI), Natural Language Processing (NLP), and Theoretical Foundations of Computer Science (TGI). Note that for DBS, there are no transcriptions available. In total, the slide materials comprise 3.9K slides containing 0.2M tokens, with an average of approximately 50 tokens per slide. The lecture transcriptions consist of 3.3M tokens.

4. Incoorporating Course Material

We extend LLMs' knowledge with the course materials in two different ways: Retrieval-Augmented Generation (RAG) and Continual Pre-Training (CPT).

4.1. Retrieval-Augmented Generation

In Retrieval-Augmented Generation (RAG), the goal is to retrieve relevant information from the course materials and include it in the input to the LLM at inference time. The overall workflow is shown in Figure 1.

RAG Components We follow the standard RAG architecture, which consists of a *retriever* and a *generator*. The retriever identifies the most relevant segments of course material, while the generator (the LLM) uses these segments to produce an answer. The generator is the model evaluated for its ability to serve as an educational assistant. For retrieval, we adopt the pre-trained **M3-Embedding** model (Chen et al., 2024), chosen for its multilingual coverage (as our materials are in both English and German) and its compact size (559M parameters), which makes it efficient for practical use.

RAG Knowledge Base To build the knowledge base, we segment course materials into semantically coherent chunks, each ideally representing an independent concept. For slides, each slide is treated as one chunk. For lecture transcripts, which are continuous in nature, we segment the text sequentially while ensuring that sentences remain intact and that each chunk does not exceed a pre-defined maximum size. We also apply a 10% overlap between consecutive chunks to reduce context loss. Each chunk is then embedded using the M3 model to create the vector database.

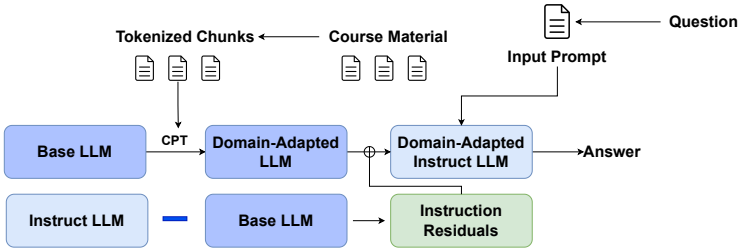


Figure 2: CPT workflow, assuming the availability of both base and instruction-tuned versions of the LLM.

RAG Workflow At inference time, a student question is embedded using the same M3 model as the course materials. Cosine similarity is then used to retrieve the k most relevant chunks from the vector database. These retrieved chunks are passed to the generator as contextual information for answer generation. The question and the retrieved content are clearly separated in the prompt to help the model distinguish between them, and the model is instructed to rely on the retrieved content only when relevant.

Multi-Modal RAG for Slide Materials A unique aspect of our approach is a **multi-modal RAG setup** designed for *visually capable LLMs*. While the retriever always operates on text embeddings, obtained from the textual content extracted from slides, the generator receives the *actual slide images* instead of their text representation. This design preserves visual information that would otherwise be lost during text extraction, such as images, slide layout, and formulas. In this configuration, the retriever efficiently locates relevant slides via text embeddings, and the generator benefits from the richer visual context during response generation.

4.2. Continual Pre-Training

In Continual Pre-Training (CPT), we continue pre-training the base LLMs on the next word prediction task using course material data (Figure 2).

Data Preprocessing The text is divided into semantically logical chunks, following the same procedure as in RAG to respect the LLM's maximum context length. However, unlike for RAG, we do not maintain overlaps between chunks.

Continual Pre-Training Special care is required when performing CPT with the very limited course material data, as the model is otherwise prone to catastrophic forgetting of previously acquired knowledge. To mitigate this risk, we reduce the maximum learning rate (LR) relative to the original pre-training process and apply both LR warm-up and LR decay. In addition, we replay a portion of the original pre-training data during CPT. Validation perplexity on both the original pre-training data and the course material data is monitored to determine an appropriate early stopping point, preventing overfitting to the course material.

Instruction Following After CPT, we aim to restore instruction-following capabilities of the LLM. To achieve this, we adopt the approach of Jindal et al. (2024), which introduces Instruction Residuals (IRs). This method leverages a pre-existing instruction-tuned version of the base LLM on which CPT was performed. The IRs are computed by subtracting the weights of the base model, denoted as θ_b, from those of the instruction-tuned model, denoted as θ_i:

$$IR(\theta_i, \theta_b) = \theta_i - \theta_b \tag{1}$$

Then, the residuals $IR(\theta_i, \theta_b)$ are added to restore the instruction-following of a continually pre-trained base LLM with weights $\theta_{b_{cpt}}$:

$$\theta_{i_{cpt}} = \theta_{b_{cpt}} + IR(\theta_i, \theta_b) \tag{2}$$

The resulting model with weights $\theta_{i_{cpt}}$ is expected to be adapted with the course materials while retaining its ability to follow instructions.

5. Experimental Setup

5.1. Data

Course Materials For extending LLMs' knowledge, we use the course materials data detailed in Section 3.

Replayed Data for CPT As replayed data to be used with CPT, we use English and German Wikipedia data. The amount of Wikipedia data used is half the amount (in terms of tokens) of the course materials data, as we defined the replay ratio to be 33% (detailed in Section 5.3).

	Full name	# Params	Quant.	Handle Image
LLaMA 3.1	meta-llama/Llama-3.1-8B-Instruct	8B	-	no
LLaMA 3.3	meta-llama/Llama-3.3-70B-Instruct	70B	4 bit	no
Qwen2-VL	Qwen/Qwen2-VL-72B-Instruct-GPTQ-Int4	72B	4 bit	yes
Qwen2.5	Qwen/Qwen2.5-72B-Instruct	72B	4 bit	no
Mistral Large 2	mistralai/Mistral-Large-Instruct-2411	123B	4 bit	no

Table 1: Details of the LLMs used.

Evaluation Data As discussed in Section 2, we use the SciEx benchmarking dataset (Dinh et al., 2024) to evaluate our approaches. SciEx has in total 10 exams and 154 unique questions. Each question is annotated with a difficulty level among {*easy, difficult, hard*}. SciEx also comes with answers to the questions from different LLMs, along with expert-annotated scores on the answers. The provided expert scores on pre-generated answers enable us to evaluate potential LLM graders for automatic evaluation used in our paper.

5.2. Models

The list of all LLMs used in our experiments can be found in Table 1. All of them are instruction-tuned versions of the base models and available on Hugging Face[0].

Models For Grading LLaMA 3.3 70B and Qwen2.5 72B are used as candidate grader LLMs for evaluation, as they are recent models with large parameter sizes. We additionaly include Mistral Large 2 123B as a candidate grader, as LLaMA 3.3 70B and Qwen2.5 72B might exhibit biases when grading their own answers.

Models For RAG We experiment with LLaMA 3.3 70B, Qwen2.5 72B, LLaMA 3.1 8B, and Qwen2-VL 72B in the RAG setting. LLaMA 3.3 and Qwen2.5 are included as they represent the flagship models of their respective families at the time of the experiments. LLaMA 3.1 8B is evaluated to examine whether incorporating course materials has a different effect on older, smaller models. Finally, Qwen2-VL is tested to assess the impact of including course material images, as it can directly process visual inputs.

Models For CPT We experiment with LLaMA 3.1 8B for Continual Pre-Training (CPT), as its smaller size makes CPT more computationally feasible. Using LLaMA 3.1 8B also enables a direct comparison between CPT and RAG, since the same model is employed in both settings.

[0] https://huggingface.co/docs/hub/en/models-the-hub

5.3. Hyperparameters

RAG For RAG, we experiment with different numbers of retrieved chunks k. For slide chunks, we vary k between 1 and 20 and find that $k = 4$ yields the best average performance. For transcripts, we fix $k = 4$ and vary the chunk size across 100, 300, and 750 tokens, observing that 300 tokens provides the best results.

CPT We use an effective batch size of 32, a maximum sequence length of 2048, a cosine learning rate schedule, and a weight decay of 0.1. Since CPT is performed on a very small dataset, careful tuning of the learning rate is crucial. We conduct a grid search with CPT on synthetic transcripts training data from the Human–Computer Interaction (HCI) lecture, varying the learning rate from 3e-6 to 6e-5 and the number of warm-up steps among 0, 5, 10. The best configuration is a learning rate of 2e-5 with 5 warm-up steps, which is significantly lower than the original maximum pre-training learning rate of 3e-4 for LLaMA 3.1. We then apply these hyperparameters to the final setting, in which we combine the slides and the synthetic transcripts of all courses for CPT, given the limited dataset size. The training employs a replay ratio of 33%, i.e., 33% of the training tokens is Wikipedia data. CPT is run for a maximum of 15 epochs, but training is stopped early once validation domain perplexity ceases to improve. We employ training-to-validation split ratio of 90:10.

5.4. Evaluation

We employ automatic grading with an LLM to evaluate the proposed approaches on the university exam questions in SciEx. As candidate graders, we consider LLaMA 3.3, Qwen2.5, and Mistral Large 2. Among these, Mistral Large 2 performs best, achieving a Pearson correlation of 0.84 with the expert-provided grades in the SciEx dataset.

Adhering to SciEx, the exam-level scores range from 0 to 60 points. Question-level scores have different ranges, so we report the normalized scale, ranging from 0 to 100.

In all experiments using LLMs to solve questions in SciEx, we use vanilla prompting without incorporating course materials as the baseline for the

knowledge-enhanced approaches (RAG and CPT).

The prompt used to generate answers is in Appendix B. The prompt used to grade exams is in Appendix C.

5.5. Hardwares

Inferencing is performed on either an NVIDIA A100 (80 GB), an H100 (94 GB), or two V100s (32 GB). Training is performed on four NVIDIA H100s.

6. Results and Discussion

6.1. LLM Performance with RAG

To understand the effectiveness of Retrieval-Augmented Generation (RAG) in the educational context, we analyze how different factors influence its performance. We examine model size and architecture, course topics, question difficulty, the modality and type of course materials used.

6.1.1. Effect of Model Architecture and Size

RAG consistently improves model performance Table 2 shows the performance of each LLM with and without RAG using slide text. Overall, incorporating RAG with slide text improves the performance of all models. Among them, LLaMA 3.1 8B, the smallest model, shows the least improvement (+0.8 points), likely due to its more limited capacity to leverage additional context.

Model	Baseline	RAG Slide Text
Qwen2.5	45.27	**47.36**
LLaMA 3.3	43.10	**44.01**
Qwen2-VL	38.05	**40.87**
LLaMA 3.1	31.17	**31.97**
Average	39.40	**41.05**

Table 2: RAG with slide text: average performance across all exams for each LLM. (Exam-level scores, 0-60 range.)

6.1.2. Effect of Course Topics

RAG mitigates knowledge gaps Table 3 presents the average performance across exams from different topics. On certain topics, such as Human–Computer Interaction (HCI), RAG yields substantial improvements (+6.11 points). For others, performance either improves modestly or remains stable. As Dinh et al. (2024) observed, LLMs often lack context for HCI, which our results suggest can be partially addressed by retrieving course-specific materials. RAG thus appears to be the most useful for filling knowledge gaps while maintaining performance in areas where LLMs are already strong.

Exam	Baseline	RAG Slide Text
HCI[1]	41.91	**48.02**
DBS[2]	30.16	**32.56**
NLP[1]	43.76	**44.34**
Algo	**35.38**	35.13
TGI	**41.38**	41.25
DLNN[1]	**42.78**	42.15
Average	39.40	**41.05**

[1] Averaged over English and German Exams
[2] Averaged over 2022 and 2023 Exams

Table 3: RAG with slide text: average performance across all LLMs for each exam topic. (Exam-level scores, 0-60 range.)

6.1.3. Effect of Question Difficulty

RAG provides greater benefits on difficult questions Table 4 reports the impact of RAG when questions are grouped by difficulty levels (as defined in SciEx). RAG yields the largest gains on hard questions (+10.79%). In contrast, improvements on easy and medium questions are smaller.

Difficulty	Baseline	RAG Slide Text	Δ*
Easy	71.71	73.63	1.92
Medium	66.83	72.16	5.33
Hard	56.72	67.51	10.79

* RAG improvement over the baseline.

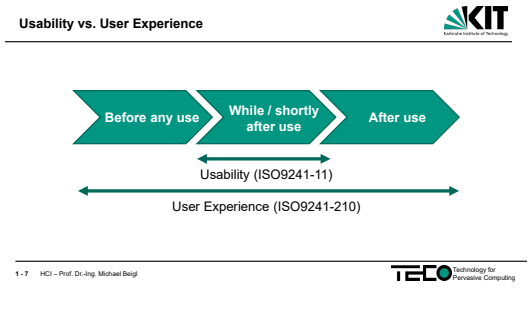
Table 4: RAG with slide text: average performance across questions of different difficulty levels. (Question-level scores, range 0-100.)

6.1.4. Effectiveness of Multimodal RAG

Slides as images outperform text extraction We examine whether visually capable LLMs can benefit from the non-textual aspects of slides under our multimodel RAG setting. Using Qwen2-VL, we compare slide text retrieval versus slide image input (Table 5). The results show that providing slide images leads to higher performance, confirming that visual information, such as images, layout, and formulas, contributes to better comprehension. Figure 3 shows an example of the effectiveness of multimodal RAG, where Qwen2-VL gives better answers by leveraging the visual information in the retrieved slide.

6.1.5. Effect of Course Material Types

Slides outperform lecture transcriptions for RAG Table 6 compares the effectiveness of using lecture transcriptions (both raw and LLM-polished) versus slides as materials for RAG. Unlike slides,



Question:
*What is the difference between "**usability**" and "**user experience**"?*

Ground-truth:
*Usability is assessed **while or shortly after use***
*User experience **extends to before and after use**.*

Qwen2-VL answer **using slide text**:
Usability refers to the ease of use of a product or system, while user experience refers to the overall satisfaction and enjoyment that a user gets from using a product or system.
→ **Graded 0.5 / 1.0: fail to accurately reflect the temporal aspect**

Qwen2-VL answer **using slide image**:
*… User Experience (UX):… broader than usability … and considers the entire journey of the user, … **including before, during, and after use** …*
→ **Graded 1.0 / 1.0**

Figure 3: Example of multimodal RAG effectiveness: Qwen2-VL produces a more accurate answer when provided with the slide as an image rather than as plain text. Viewing the slide image preserves the visual context, specifically, **the arrows illustrating the temporal scope of *usability* and *user experience***.

Exam	Base Model	RAG slide text	RAG slide image
HCI[1]	39.60	47.20	**50.75**
DBS[2]	23.03	26.15	**32.88**
NLP[1]	44.62	43.88	**44.50**
Algo	38.50	**42.00**	37.00
TGI	43.50	41.50	**48.00**
DLNN[1]	41.98	**45.38**	41.85
Average	38.05	40.87	**42.50**

[1] Averaged over English and German Exams
[2] Averaged over 2022 and 2023 Exams

Table 5: RAG slide text versus slide image: effect on Qwen2-VL performance. (Exam-level scores, range 0-60.)

transcriptions fail to provide improvements and, in many cases, even reduce performance compared to the baseline. A likely explanation is that slides contain concise and well-structured text, whereas transcriptions consist of spoken language which tends to be more more verbose and less organized, thus may distract the model from essential information. In the following experiments, when comparing CPT to RAG, we report the performance of RAG with slide text.

6.2. LLM Performance with CPT

We analyze how CPT on course materials effect LLaMA 3.1 8B performance. We examine the model's learning dynamics, overall performance, and sensitivity to different types of course materials. In the experiments, we perform CPT using combined slide text and polished transcripts from all courses, unless stated otherwise.

	Baseline	Slide Text	Transcripts	Polished Transcripts
HCI[1]	41.91	**48.02**	40.13	40.13
TGI	41.38	41.25	**41.38**	39.75
DLNN[1]	**42.78**	42.15	42.05	39.12
NLP[1]	43.76	**44.34**	42.89	42.96
Algo	**35.38**	35.13	34.25	32.50
Average	41.70	**43.17**	40.72	39.58

[1] Averaged over English and German Exams

Table 6: RAG slide text versus transcript: Effect on performance on each topic. DBS is excluded as there are no transcripts. (Exam-level scores, range 0-60.)

6.2.1. Adaptation Dynamics During CPT

LLaMA 3.1 8B can learn from course materials Figure 4 shows the validation losses (measured in perplexity) during CPT on course material data and Wikipedia data. The validation perplexity on the course materials drops sharply in the first few steps before stabilizing, whereas the Wikipedia validation perplexity gradually increases over time. By applying early stopping once the course material perplexity converges, we obtain a model that adapts effectively to the course materials while minimizing forgetting of its original knowledge.

6.2.2. Overall Performance

CPT does not bring consistent improvement Table 7 compares the performance of CPT with the baseline and RAG. On average, CPT performs worse than both the baseline and RAG. To further investigate, we break down the results by exam language, focusing on exams available in both English and German (Table 8). The results show that CPT yields improvements over the baseline in some cases (e.g., the HCI English exam, the NLP German exam, and the DLNN German exam), but

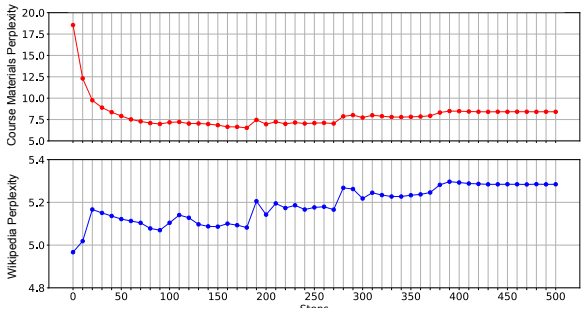


Figure 4: Course materials and Wikipedia (replay) validation perplexity for CPT with LLaMA 3.1 8B.

causes significant drops in others (notably the HCI German exam). The overall loss in performance is likely due to catastrophic forgetting, which is difficult to avoid when conducting CPT on the limited amount of course material data.

	LLaMA 3.1		
	Baseline	**RAG**	**CPT**
HCI[1]	39.40	**42.85**	37.85
DBS[2]	21.93	**26.35**	21.85
TGI	**29.00**	26.00	18.00
DLNN[1]	**34.12**	29.62	31.88
NLP[1]	34.75	**37.25**	34.25
Algo	**23.00**	22.00	19.00
All Exams	31.17	**31.97**	28.86

[1] Averaged over English and German Exams
[2] Averaged over 2022 and 2023 Exams

Table 7: CPT: LLaMA 3.1's performance across exam topics. (Exam-level scores, range 0-60.)

	LLaMA 3.1			
	HCI[1]	**NLP[2]**	**DLNN[2]**	**All Exams**
English				
Baseline	38.70	37.00	**34.30**	36.67
RAG	**42.40**	**42.00**	31.75	**38.72**
CPT	41.90	35.50	31.68	36.36
German				
Baseline	39.40	32.50	**33.95**	**35.28**
RAG	**42.85**	32.50	27.50	34.28
CPT	33.80	**33.00**	32.08	32.96

[1] Provided course material was English and German
[2] Provided course material was English

Table 8: CPT breakdown by language: LLaMA 3.1's performance on German and English exams. (Exam-level scores, range 0-60.)

6.2.3. Effect of Course Material Type

Polished transcripts yield the best CPT performancee We conduct an ablation study to examine the effect of different types of course materials for CPT on model performance. Due to computational constraints, this experiment is limited to the Human–Computer Interaction (HCI) topic. The results are shown in Table 9. We observe the same pattern: CPT in general performs worse than RAG. Among the CPT configurations, the best results are achieved when using polished transcripts. This outcome is expected, as the polished transcripts more closely resemble the natural text that LLMs are typically trained on.

	LLaMA 3.1		
	HCI EN	**HCI DE**	**Average**
Baseline	38.70	39.40	39.05
RAG w. Slide Text (EN)	42.40	**42.85**	**42.62**
CPT w. Slide Text (EN)	**42.90**	33.15	38.03
CPT w. Transcripts (DE)	40.00	33.40	36.70
CPT w. P. Transcripts (DE)	39.80	39.10	39.45

Table 9: CPT performance on the HCI course using different course material types. (Exam-level scores, range 0-60.)

7. Conclusion

This work investigates how university course materials can be leveraged to improve Large Language Models (LLMs) in educational settings. We compare two strategies, Retrieval-Augmented Generation (RAG) and Continual Pre-Training (CPT), to incorporate course-specific knowledge from materials such as lecture slides and transcripts. Our findings show that, given the limited size and multimodal nature of these materials, RAG is a more effective and efficient approach than CPT, which suffers from catastrophic forgetting. Furthermore, we demonstrate that incorporating slides as images in a multi-modal RAG setup yields additional performance gains over text-only retrieval, highlighting the importance of preserving visual and structural information in educational resources. Overall, our results suggest that lightweight retrieval-based methods offer a practical path toward developing AI assistants that can better support university-level learning.

Limitations

While our results demonstrate the potential of using university course materials to extend LLMs' scien-

tific knowledge, several limitations remain. First, our experiments are limited to computer science courses from a single institution. As course structure, teaching style, and content depth vary across universities, the observed improvements may not be the same as for other academic settings. Second, while we found RAG to be more robust than CPT for small specialized datasets, we did not perform CPT on models other than Llama 3.1 8B, on which CPT might have a more positive effect. Finally, all evaluations relied on automatic grading using an LLM, which, despite strong correlation with human scores, may introduce systematic biases or grading inconsistencies compared to human assessment.

Ethical Considerations

This work uses the publicly available SciEx benchmark and the explicitly authorized university course materials. No personally identifiable information or student data were processed. Although the goal of this research is to enhance educational accessibility and domain knowledge coverage, there are ethical concerns regarding the potential misuse of LLMs for generating unauthorized teaching material or student assessments. We discourage such uses and emphasize transparent documentation of data sources.

Acknowledgments

This work was supported by the Helmholtz Programme-oriented Funding, with project number 46.24.01, project name AI for Language Technologies. We acknowledge the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Wurttemberg and by the Federal Ministry of Education and Research. This work also received support from the Horizon Europe grant 101213369 DVPS.

Bibliographical References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, Zheng Liu, et al. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Ozan Gokdemir, Carlo Siebenschuh, Alexander Brace, Azton Wells, Brian Hsu, Kyle Hippe, Priyanka Setty, Aswathy Ajith, J Gregory Pauloski, Varuni Sastry, et al. 2025. Hiperrag: High-performance retrieval augmented generation for scientific insights. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–13.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Christian Huber, Tu Anh Dinh, Carlos Mullov, Ngoc-Quan Pham, Thai Binh Nguyen, Fabian Retkowski, Stefan Constantin, Enes Ugan, Danni Liu, Zhaolin Li, Sai Koneru, Jan Niehues, and Alexander Waibel. 2023. End-to-end evaluation for low-latency simultaneous speech translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 12–20, Singapore. Association for Computational Linguistics.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*.

Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. 2023. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11):btad651.

Ishan Jindal, Chandana Badrinath, Pranjal Bharti, Lakkidi Vinay, and Sachin Dev Sharma. 2024. Balancing continuous pre-training and instruction fine-tuning: Optimizing instruction-following in llms. *arXiv preprint arXiv:2410.10739*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation

for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. 2024. Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6):btae353.

Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. 2024. D-cpt law: Domain-specific continual pretraining scaling law for large language models. *Advances in Neural Information Processing Systems*, 37:90318–90354.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Jimeng Shi, Sizhe Zhou, Bowen Jin, Wei Hu, Shaowen Wang, Giri Narasimhan, and Jiawei Han. 2025. Hypercube-rag: Hypercube-based retrieval-augmented generation for in-domain scientific question-answering. *arXiv preprint arXiv:2505.19288*.

Ioannis Tsiamas, Gerard I Gállego, José AR Fonollosa, and Marta R Costa-jussà. 2022. Shas: Approaching optimal segmentation for end-to-end speech translation. *Interspeech 2022*, pages 106–110.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2024. Efficient continual pre-training for building domain specific large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10184–10201.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Çağatay Yıldız, Nishaanth Kanna Ravichandran, Nitin Sharma, Matthias Bethge, and Beyza Ermis. 2024. Investigating continual pretraining in large language models: Insights and implications. *arXiv preprint arXiv:2402.17400*.

Xianrui Zhong, Bowen Jin, Siru Ouyang, Yanzhen Shen, Qiao Jin, Yin Fang, Zhiyong Lu, and Jiawei Han. 2025. Benchmarking retrieval-augmented generation for chemistry. *arXiv preprint arXiv:2505.07671*.

Language Resource References

Dinh, Tu Anh and Mullov, Carlos and Bärmann, Leonard and Li, Zhaolin and Liu, Danni and Reiß, Simon and Lee, Jueun and Lerzer, Nathan and Gao, Jianfeng and Peller-Konrad, Fabian and Röddiger, Tobias and Waibel, Alexander and Asfour, Tamim and Beigl, Michael and Stiefelhagen, Rainer and Dachsbacher, Carsten and Böhm, Klemens and Niehues, Jan. 2024. *SciEx: Benchmarking Large Language Models on Scientific Exams with Human Expert Grading and Automatic Grading*. Association for Computational Linguistics. PID https://doi.org/10.18653/v1/2024.emnlp-main.647.

Lu, Pan and Mishra, Swaroop and Xia, Tony and Qiu, Liang and Chang, Kai-Wei and Zhu, Song-Chun and Tafjord, Oyvind and Clark, Peter and Ashwin Kalyan. 2022. *Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering*. PID https://huggingface.co/datasets/derek-thomas/ScienceQA.

Welbl, Johannes and Liu, Nelson F. and Gardner, Matt. 2017. *Crowdsourcing Multiple Choice Science Questions*. Association for Computational Linguistics. PID https://doi.org/10.18653/v1/W17-4413.

Zhang, Wenxuan and Aljunied, Mahani and Gao, Chang and Chia, Yew Ken and Bing, Lidong. 2024. *M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models*. PID https://cutt.ly/m3exam-data.

A. Prompt Used for Polishing Lecture Transcripts

Transcript Polishing Prompt

You are a professional editor for transcripts of the lecture `{lecture_name_en}` with a deep understanding of academic content.
First, you will see the previous, already cleaned section of the transcript. This serves as context to maintain a natural transition:
[Previous section]:
`{previous_synthetic_chunk}` or [No previous section available]
Now follows another unedited section of the transcript. Please edit **only this following section** by:

- removing filler words and pauses (e.g., "uh", "so", "exactly", "no", "so to speak"),
- correcting grammatical errors,
- making the language fluent, precise, and written,
- **removing all sentences** that **are not part of the technical content of the lecture**. This includes in particular:
  - Organizational information (e.g., about homework, exams, slides, Moodle, technical problems),
  - Comments on the room, time, breaks, sensitivities, or jokes,
  - Personal anecdotes or conversations with students,
  - Repetitions without added value, digressions, or small talk.

The **technical content** (e.g., definitions, examples, proofs, concepts) **must not be summarized or shortened.**
The structure and order of the content should be retained as far as possible.
Please only return the edited, subject-relevant section of text — **without introduction, explanations, or additional comments**.

Here is the new section:
`{new_section}`

The prompt is translated into German when polishing German transcripts.

B. Prompt Used for Generating Exam Answers

Main Prompt

You are a university student. Please answer the following JSON-formatted exam question.
The subquestions (if any) are indexed.
The provided figures (if any) each contains its path at the bottom, which matches the path provided in the JSON.

`{context_message}`

Please give the answers to the question and subquestions that were asked, and index them accordingly in your output.
You do not have to provide your output in the JSON format.
If you are asked to draw on the figure, then describe with words how you would draw it.
Please provide all answers in English.
Here is the question:

`{question}` or `{question_with_context}`

The following prompts are used for the RAG experiments. They are placed in the Main Prompt instead of the `{context_message}` field.

Context Message: Text Course Material

Additionally, you will be provided with some course materials
(can be found in 'Context').
You can use that additional context to answer the question if it is helpful.
If it is not helpful, you don't have to use it.

Context Message: Image Course Material

You will also be provided with images of course material.
These images each contain a path at the bottom, which corresponds to the path specified in the JSON under 'Context'.
You can use these images to answer the question if this is helpful.
If they are not helpful, you don't have to use them.

The prompt is translated into German when generating German answers for German questions.

C. Prompt Used for Grading Exams

The following prompt is used to grade exams for all experiments. They were originally used in the SciEx paper (Dinh et al., 2024).

Grading Prompts

You are a university professor. Please grade the following exam question.
The exam question, examinee's answer, correct answer and the maximum possible score are provided in the format:

```
[question] <exam_question>
[/question]
```

```
[answer] <answer> [/answer]
[correct_answer] <cor-
rect_answer> [/correct_answer]
[max_score] <max_score>
[/max_score]
```

The question is provided in JSON format, but the answer can be freeform text.
The provided figures in the question (if any) each contain their path at the bottom, which matches the path provided in the JSON.
The answer is text-only.
If the question asks to draw on the figure, then the answer should contain a text description of how the drawing should be.
Please provide the grade between `[0, <max_score>]`
Please provide the reasoning for your grade.
Please provide your output in the format:

```
[reason] <reasoning> [/reason]
[grade] <grade> [/grade]
```

Here is your input:

```
{question}, {answer}, {refer-
ence_answer}, {max_points}
```

The prompt is translated into German when grading German answers for German questions.

Generating Research Data Metadata from Their Accompanying README Files

Kotaro Sekido[1], **Yu Watanabe**[1], **Koichiro Ito**[1], **Shigeki Matsubara**[1,2]
[1]Graduate School of Informatics, Nagoya University
[2]Information Technology Center, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
{sekido.kotaro.h0,watanabe.yu.x3}@s.mail.nagoya-u.ac.jp
{ito.koichiro.z5,matsubara.shigeki.z8}@f.mail.nagoya-u.ac.jp

Abstract

Software repositories have conventionally been used for software development. Recently, they have also served as research data repositories. Research data published in such repositories are frequently accompanied by README files; however, the data frequently lack structured metadata. To address this issue, this paper investigates the feasibility of generating research data metadata from their accompanying README files. First, we analyze the occurrence patterns of metadata-related information in README files. The results of this analysis demonstrated that README files could serve as valuable resources for metadata generation. We then performed an experiment on extracting metadata-related information from README files using large language models (LLMs) and evaluated their performance. The experimental results demonstrated that LLMs could extract metadata-related information with high performance.

Keywords: open science, research data management, metadata generation, LLM

1. Introduction

Software repositories, e.g., GitHub[1], have been primarily used for software development. In addition to source code, they typically include README files that describe usage and other details. Recently, software repositories have been increasingly used as platforms to publish research data; thus, they also function as repositories for research data (Koesten et al., 2020). In such cases, README files are frequently provided to describe the overview and usage of the corresponding research data.

To facilitate smooth circulation of research data, the data should be described with rich metadata (Wilkinson et al., 2016) comprising structured fields and their values (as shown in Table 1). The registration of metadata in metadata repositories improves the searchability of research data. However, research data published in software repositories frequently lack metadata (Gesese et al., 2025), which makes it difficult for researchers to discover the research data, even if they are accessible to the public.

In this paper, we investigate the feasibility of generating metadata for research data from the accompanying README files. We expect that the searchability of research data published in software repositories can be improved if metadata can be generated from README files and registered in metadata repositories.

[1]https://github.co.jp/

Field	Value
Title	Helsinki Prosody Corpus
PublicationYear	2019
Creator	Aarne Talman, Antti Suni, ...
Subject	Prosody Prediction
Language	English

Table 1: Partial metadata of the Helsinki Prosody Corpus (Talman et al., 2019).

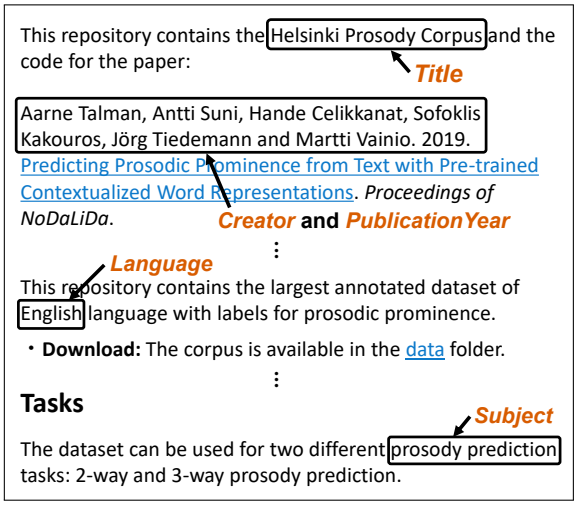


Figure 1: Excerpt from the README file of the Helsinki Prosody Corpus (Talman et al., 2019).

First, we analyzed the occurrence patterns of metadata-related information in README files. In this analysis, we targeted GitHub repositories as

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 180–185
12 May 2026.

one of the most widely used software repositories. The results demonstrated that although there were differences among the metadata fields, README files could serve as valuable resources for metadata generation. We then performed an experiment to evaluate the feasibility of extracting metadata-related information from README files. In this experiment, large language models (LLMs) were employed for information extraction, and we evaluated their performance. The results demonstrated that the LLMs could extract metadata-related information from the README files with high performance.

2. Metadata Generation for Research Data

2.1. Research Data Circulation and Metadata

Recently, with global trends toward open science, the importance of publishing and sharing research data has increased (UNESCO, 2021; OSWG, 2023). However, to facilitate the circulation of research data, it is essential to provide rich metadata (Wilkinson et al., 2016). Table 1 shows part of the metadata for the Helsinki Prosody Corpus, which is a dataset used to predict speech prosody, as an example of research data metadata. Research data metadata include information that characterizes the data, e.g., the name of the data (*Title*), year of publication (*PublicationYear*), creators (*Creator*), subject or keywords (*Subject*), and languages (*Language*).

2.2. Metadata Generation using README Files

A README file for research data is a document created to facilitate use of the corresponding data. In other words, README files support human understanding and reuse of the data. Figure 1 shows an excerpt from the README file of the Helsinki Prosody Corpus[2] as a representative example of a README file for research data. A research data README file contains descriptions of the data and may include metadata-related information. For example, the README file shown in Figure 1 includes information corresponding to the metadata presented in Table 1, e.g., the *Title* and *PublicationYear* of the research data. One approach to improving the discoverability of research data in software repositories is to generate metadata from README files and register them in metadata repositories.

[2] https://github.com/Helsinki-NLP/prosody

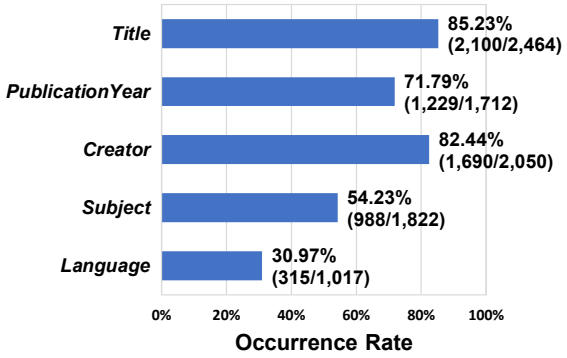


Figure 2: Occurrence rate for each metadata field.

2.3. Related Work

Previous studies have attempted to extract information about research data from documents that mention the data. Most of these studies targeted software or datasets, and they extracted metadata-related information from scholarly papers (Schindler et al., 2021; Stavropoulos et al., 2023; Watanabe et al., 2024; Alyafeai et al., 2025; Watanabe et al., 2025).

Several studies have attempted to acquire information about research data from README files (Kumar et al., 2024; Utrilla Guerrero et al., 2024). These studies focused on software and attempted to extract mentions related to functionality or usage. However, to the best of our knowledge, no previous studies have investigated the use of README files to generate research data metadata.

3. Analysis

In this study, we first evaluated whether README files can serve as valuable resources for metadata generation. Here, we analyzed the occurrence patterns of metadata-related information in README files.

Recently, an increasing number of datasets have been registered in software repositories (Koesten et al., 2020; Roman et al., 2023). Despite this trend, research focusing specifically on such datasets remains limited compared with studies focused on software and other artifacts. Thus, in this analysis, we focused on datasets as the target research data.

3.1. Analysis Data

The analysis required data comprising pairs of metadata and README files for research data. Thus, we constructed the data to be analyzed from Papers With Code Datasets (PWCD) (Papers With Code, 2019–25). PWCD includes dataset metadata and the URLs of their corresponding publication repositories. In this study, we targeted URLs

whose publication platform is GitHub, and we retrieved the README files using the GitHub API. Note that repositories containing multiple README files were excluded from the analysis because it was unclear which README file corresponded to the dataset. In addition, the metadata in PWCD are written in English; thus, we limited our collection to repositories with English README files.

Among the 2,737 collected pairs of metadata and README files, we used 2,464 pairs for analysis, excluding those whose repositories were created after 2023.

Among the metadata fields included in the collected data, we selected those corresponding to fields in the DataCite Metadata Schema (DataCite Metadata Working Group, 2024), which is a global standard metadata schema, as the targets of the analysis. As a result, the five fields shown in Table 1 were selected.

3.2. Analysis Method

The method employed to determine the occurrence of metadata values in the README files is described as follows. First, we applied text preprocessing to both the metadata and README files to reduce the effects of orthographic variations. Then, for each metadata field, we determined whether its value occurred in the README file. In PWCD metadata, the values for *Creator*, *Subject*, and *Language* are represented as lists. For these fields, if at least one element of the list occurred in the README file, we determined that the values for those fields occurred. Note that some metadata fields in PWCD contain missing values. For such cases, these entries were excluded from the calculation of the occurrence rate of the metadata values in README files.

3.3. Analysis Results

Figure 2 shows the calculated occurrence rate for each metadata field. As shown, the fields with the highest occurrence rate were *Title* and *Creator*, both of which exceeded 80%. Furthermore, the field with the lowest occurrence rate, i.e., the *Language* field, reached 30%. The macro-average occurrence rate across the five fields was 64.93%. From these results, although the occurrence rate varied among the fields, we confirmed that README files can serve as valuable resources for metadata generation.

4. Experiment

In this study, we performed an experiment to investigate the feasibility of extracting metadata-related information from README files using LLMs. As in Section 3, we targeted the README files for datasets in this experiment.

```
# System
You are an information extraction model.
Your task is to extract information about dataset from the provided README.
You must output the results strictly according to the following definitions.

Definitions
- name: The name of the dataset.
- introduced_year: The year when the dataset was introduced.
- creators: The creators of the dataset.
- tasks: The tasks that dataset is intended for.
- natural_languages: The natural languages of the dataset.

If a value is not stated in the README, set the value to null.

# User
README:
{readme_text}
```

Figure 3: LLMs prompt.

4.1. Experimental Data

Here, we split the analysis data described in Section 3 into training and development sets, and we allocated data that were not used in the analysis to the test set. The training, development, and test sets included 2,189, 275, and 273 samples, respectively. In this experiment, the development set was used for prompt design, and the test set was used to evaluate the extraction performance.

4.2. Implementation

In this study, we employed Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Ministral-8B-Instruct-2410 (Mistral AI Team, 2024), and Qwen3-8B (Yang et al., 2025) as open LLMs, and GPT-5 (Open AI, 2025) as a closed LLM. Here, we used vLLM[3] (Kwon et al., 2023) for the open LLMs and the OpenAI API[4] for GPT-5. For vLLM, the temperature was set to 0.0. Note that we limited the model responses to JSON format consisting of the five fields shown in Table 1 using Structured Outputs. Figure 3 shows the prompt. The prompt included instructions to extract values for each metadata field and to output "null" if a value could not be identified.

[3] https://github.com/vllm-project/vllm
[4] https://openai.com/ja-JP/api/

	Recall				Precision			
	Llama	**Ministral**	**Qwen**	**GPT**	**Llama**	**Ministral**	**Qwen**	**GPT**
Title	85.71 (198/231)	84.42 (195/231)	87.45 (202/231)	**91.34 (211/231)**	77.34 (198/256)	79.27 (195/246)	75.94 (202/266)	**83.40 (211/253)**
PublicationYear	80.20 (158/197)	76.14 (150/197)	**93.40 (184/197)**	87.82 (173/197)	88.76 (158/178)	**89.82 (150/167)**	81.78 (184/225)	86.07 (173/201)
Creator	**95.79 (182/190)**	93.68 (178/190)	95.26 (181/190)	87.37 (166/190)	98.38 (182/185)	97.80 (178/182)	89.60 (181/202)	**98.81 (166/168)**
Subject	80.19 (85/106)	75.47 (80/106)	69.81 (74/106)	**91.51 (97/106)**	**55.56 (85/153)**	48.78 (80/164)	40.22 (74/184)	55.43 (97/175)
Language	**74.19 (23/31)**	67.74 (21/31)	67.74 (21/31)	61.29 (19/31)	56.10 (23/41)	46.67 (21/45)	26.58 (21/79)	**67.86 (19/28)**

Table 2: Extraction performance (Recall and Precision).

	Llama	**Ministral**	**Qwen**	**GPT**
Title	81.31	81.76	81.29	**87.19**
PublicationYear	84.27	82.42	**87.20**	86.93
Creator	**97.07**	95.70	92.35	92.74
Subject	65.64	59.26	51.03	**69.04**
Language	63.89	55.26	38.18	**64.41**
Macro average	78.43	74.88	70.01	**80.06**

Table 3: Extraction performance (F1).

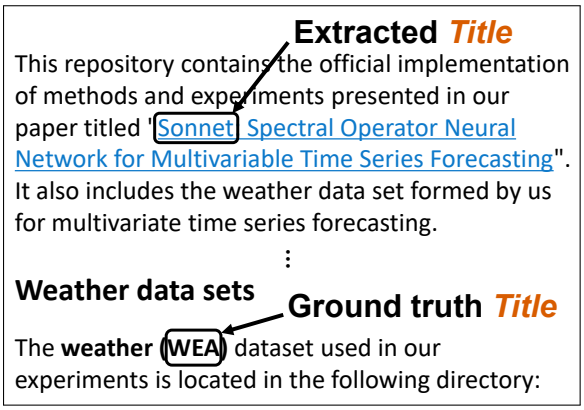


Figure 4: Excerpt from the README file of the WEA (Shu and Lampos, 2025) illustrating an extraction error caused by the presence of multiple research data.

4.3. Evaluation

In this experiment, the extraction performance of the LLMs for each metadata field was evaluated using the recall, precision, and F1-score, which are calculated as follows.

Recall. The proportion of correctly extracted metadata values among those in the README files.

Precision. The proportion of correctly extracted metadata values among non-null values output by the LLMs.

F1-score. The harmonic mean of recall and precision.

We determined whether the extracted values were correct by string matching with the ground truth. Here, partial match was employed for *Title* and *Subject*, and exact match was used for *PublicationYear*, *Creator*, and *Language*. Partial match was defined as a case where either string is a substring of the other. For list-type metadata values, i.e., *Creator*, *Subject*, and *Language*, the extraction was considered correct if at least one element matched. Note that metadata fields with missing values in the test set were excluded from the evaluation.

4.4. Experimental Results

The recall and precision results are shown in Table 2. As can be seen, for recall, although the scores for *Language* were relatively low, the best scores for the other metadata fields exceeded 90%, which indicates that the LLMs could extract metadata values from the README files with high coverage. In terms of precision, although the scores for *Subject* and *Language* were relatively low, the best scores for the other metadata fields exceeded 80%.

Table 3 shows the results for the F1-score. The macro-average for all LLMs falls in the range of approximately 70% to 80%. Among the LLMs, GPT-5 obtained the best performance. These results demonstrated that the LLMs were able to extract metadata-related information from the README files with high performance.

4.5. Error Analysis

Although the LLMs demonstrated high extraction performance, several errors were observed. Specifically, when research data of a different type from the dataset appeared in a README file, the LLMs sometimes incorrectly extracted information related to those nontarget research data. As a representative example, Figure 4 shows an excerpt from the README file of the WEA dataset (Shu and Lampos, 2025)[5] for weather forecasting. Here, al-

[5] https://github.com/ClaudiaShu/Sonnet

though the correct *Title* is “WEA,” “Sonnet,” which is a model trained using “WEA,” was incorrectly extracted as the *Title*.

To reduce such errors, LLMs must be able to appropriately identify the type of research data described in a README file, and improving prompts to enable such distinctions remains future work.

5. Conclusion and Future Work

In this study, we investigated the feasibility of generating research data metadata from their accompanying README files. The results of the analysis indicated that README files tend to contain values that corresponded to metadata fields for research data and could serve as valuable resources for metadata generation. Then, we performed an experiment using LLMs, and the results demonstrated that the LLMs could extract metadata-related information from the README files with high performance.

In the analysis and experiment, we assumed that each README file described a dataset. However, in practice, the type of research data described in a README file is not always explicit. Thus, in the future, we plan to address the identification of the type of research data described in a README file.

6. Acknowledgements

This research was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 25K03418) of JSPS and “Developing a Research Data Ecosystem for the Promotion of Data-Driven Science” of MEXT.

7. Bibliographical References

Zaid Alyafeai, Maged S. Al-shaibani, and Bernard Ghanem. 2025. MOLE: Metadata Extraction and Validation in Scientific Papers Using LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12236–12264.

DataCite Metadata Working Group. 2024. DataCite Metadata Schema Documentation for the Publication and Citation of Research Data and Other Research Outputs. Version 4.6.

Genet Asefa Gesese, Zongxiong Chen, Oussama Zoubia, and others. 2025. A Survey on Metadata for Machine Learning Models and Datasets: Standards, Practices, and Harmonization Challenges. In *Proceedings of 5th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment (Sci-K 2025)*, volume 4065, pages 57–71.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and others. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Laura Koesten, Pavlos Vougiouklis, Elena Simperl, and Paul Groth. 2020. Dataset Reuse: Toward Translating Principles to Practice. *Patterns*, 1(8).

Prince Kumar, Srikanth Tamilselvam, and Dinesh Garg. 2024. Read between the Lines - Functionality Extraction from READMEs. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3977–3990.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, and others. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP 2023)*, page 611–626.

Mistral AI Team. 2024. Un Ministral, des Ministraux.

Open AI. 2025. Introducing GPT-5.

G7 OSWG. 2023. Annex 1: G7 Open Science Working Group (OSWG).

Anthony Cintron Roman, Kevin Xu, Arfon Smith, and others. 2023. Open Data on GitHub: Unlocking the Potential of AI. *arXiv preprint arXiv:2306.06191*.

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 2021)*, pages 4574–4583.

Yuxuan Shu and Vasileios Lampos. 2025. Sonnet: Spectral Operator Neural Network for Multivariable Time Series Forecasting. *arXiv preprint arXiv:2505.15312*.

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. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 37–53.

Aarne Talman, Antti Suni, Hande Celikkanat, and others. 2019. Predicting Prosodic Prominence from Text with Pre-trained Contextualized Word Representations. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa 2019)*, pages 281–290.

UNESCO. 2021. UNESCO Recommendation on Open Science.

Carlos Utrilla Guerrero, Oscar Corcho, and Daniel Garijo. 2024. Automated Extraction of Research Software Installation Instructions from README Files: An Initial Analysis. In *Proceedings of the Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, pages 114–133.

Yu Watanabe, Koichiro Ito, and Shigeki Matsubara. 2024. Capabilities and Challenges of LLMs in Metadata Extraction from Scholarly Papers. In *Proceedings of the 26th International Conference on Asia-Pacific Digital Libraries (ICADL 2024)*, pages 280–287.

Yu Watanabe, Koichiro Ito, and Shigeki Matsubara. 2025. Metadata Generation for Research Data from URL Citation Contexts in Scholarly Papers: Task Definition and Dataset Construction. In *Proceedings of the Third Workshop for Artificial Intelligence for Scientific Publications (WASP 2025)*, pages 72–79.

Mark Wilkinson, Dumontier Michel, IJsbrand Aalbersberg, and others. 2016. The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Sci Data*, 3.

An Yang, Anfeng Li, Baosong Yang, and others. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

8. Language Resource References

Papers With Code. 2019–25. *Papers With Code Datasets*. PID https://huggingface.co/pwc-archive/datasets. Originally obtained via the Papers With Code GitHub repository. The dataset is not currently directly distributed from the repository (Last Accessed: 2025-08-08).

Identifying Implicit Research Data References in Paper Citations

Koshi Motegi[1], **Koichiro Ito**[1], **Shigeki Matsubara**[1,2]
[1]Graduate School of Informatics, Nagoya University, Japan
[2]Information Technology Center, Nagoya University, Japan
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
motegi.koshi.h9@s.mail.nagoya-u.ac.jp,
{ito.koichiro.z5, matsubara.shigeki.z8}@f.mail.nagoya-u.ac.jp

Abstract

To encourage the public release of research data under open science, it is beneficial to establish mechanisms for evaluating research data based on metrics such as citation counts. In scholarly papers, authors sometimes cite papers that report the creation or release of research data instead of citing the research data themselves. In this paper, as a step toward computing citation counts of research data, we investigate the feasibility of identifying paper citations that refer to research data. We conducted an identification experiment using large language models and evaluated their performance.

Keywords: research data, paper citations, scholarly papers

1. Introduction

In recent years, open science has been promoted (UNESCO, 2021; G7, 2023), and the sharing of research artifacts has been encouraged. Research artifacts include not only scholarly papers but also research data such as datasets and tools. Although open access to scholarly papers has advanced, the openness of research data remains limited. One reason for this limited openness is that evaluation metrics for research data are less established. In contrast, citation counts are widely used as a representative evaluation metric for scholarly papers, and researchers tend to focus on increasing them.

To address this situation, citation counts of research data could be considered as an evaluation metric. However, as shown in Figure 1, research data are cited in various ways in scholarly papers. Specifically, the following citation ways are considered:

(a) **Formal citation**: Citing research data by describing relevant information about the research data in the reference list, in the same way as citing scholarly papers

(b) **URL citation**: Citing research data by describing the URLs of research data in the body text or in the footnote

(c) **Paper citation**: Citing research data implicitly by citing scholarly papers that report the creation or release of research data

For the formal citation and URL citation, several initiatives have focused on aggregating citation counts of research data (Clarivate, 2023; Rosonovski et al., 2023). In contrast, for paper citations, little effort has been made to develop mechanisms for aggregating citation counts of research data. This is because paper citations that refer to research data have generally been counted as citations to the papers themselves without distinguishing whether the citations refer to research data.

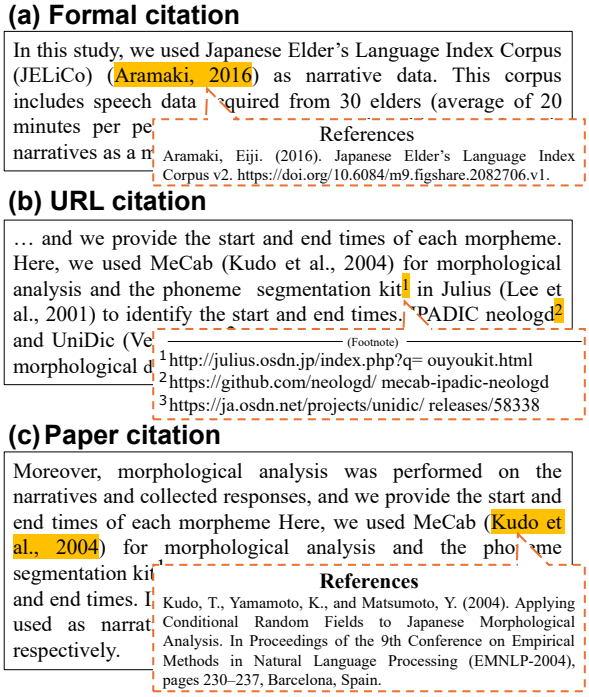


Figure 1: Ways to cite research data (examples are excerpted from Ito et al., 2022).

This study discusses the feasibility of identifying paper citations that refer to research data in scholarly papers to compute the citation counts of re-

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 186–192
12 May 2026.

search data. To this end, we attempted to automatically classify paper citations in scholarly papers as referring to research data or not, using texts surrounding citations (i.e., the citation contexts) and bibliographic information. We conducted a preliminary experiment to classify paper citations using large language models (LLMs) and evaluated their performance.

2. Citations and Evaluation of Research Data

Typically, research data are cited in one of the following three ways: formal citation, URL citation, or paper citation. In this section, we discuss existing research and efforts to compute citation counts of research data across different citation citation ways.

2.1. Formal Citation

Formal citation is a way to cite research data by providing its information, such as its title, creators, and persistent identifiers (e.g., DOI) in the reference list, in the same way as citing scholarly papers. Figure 1 (a) shows an example of a formal citation. This way has been recommended for citing research data in guidelines such as the Joint Declaration of Data Citation Principles (Group, 2014) and the Tromsø Recommendations (Andreassen et al., 2019). In addition, several conferences and journals treat formal citation as the standard to cite research data, including LREC[1], which is an international conference on language resources, and Scientific Data[2], an open-access journal that focuses on the creation and reuse of research data.

Furthermore, several systems have been developed to compute the citation counts of research data based on formal citations, such as Make Data Count[3], the Data Citation Index (Clarivate, 2023), and the Europe PMC API (Rosonovski et al., 2023). Using the citation counts obtained through these systems, research data and their creators have been evaluated in several indices, including the OmicsDI (Perez-Riverol et al., 2019) and the Data-Index (Hood and Sutherland, 2021).

2.2. URL Citation

URL citation is a way to cite research data by describing the URLs of research data in the body text or a footnote of a scholarly paper. Figure 1 (b) shows an example of URL citation. Computing URL citation counts requires identifying URLs of research data within scholarly papers.

Previous studies extracted URLs of research data from scholarly papers using regular expressions and pretrained language models (Yamamoto and Takagi, 2007; Tsunokake and Matsubara, 2022; Otto et al., 2023). In addition, some studies attempted to classify resource types referenced by URLs in scholarly papers (Zhao et al., 2019; Tsunokake and Matsubara, 2022; Wada et al., 2025).

(a) Research data

The FINECITE dataset was built from a subset of ACL Anthology Network Corpus (Radev et al., 2009). The ACL Anthology Network contains over 80K papers from several ACL conferences and other venues in computational …

(b) Concept

By embedding scientific progress and argumentation, citations serve a critical function that has been extensively examined—a research field known as citation context analysis (CCA) (Kunnath et al., 2022; Swales, 1986).

(c) Method

We considered four baselines for the classification task: (i) the scaffolding approach presented in Cohan et al.,(2019), (ii) the best-performing citation classification model from the …

: Citation tag, : Cited content

Figure 2: Examples of paper citations and cited contents (examples are excerpted from Jantsch et al., 2025).

2.3. Paper Citation

Paper citation of research data is a way to cite research data implicitly by citing a scholarly paper that reports the creation or release of research data. Figure 1 (c) shows an example of paper citation. Generally, as shown in Figure 2, paper citations are made to refer to the concepts, methods, or findings presented in the cited paper. That is, citations that refer to research data are not necessarily frequent. Therefore, to compute the citation counts of research data accurately, it is necessary to identify paper citations that refer to research data.

A previous study introduced metrics to evaluate research data that consider the citation counts of papers reporting the creation or release of the data (Callahan et al., 2018). However, in that study, the papers were identified using indexing terms from PubMed's thesaurus, MeSH (Medical Subject Headings)[4], which makes it challenging to apply the approach beyond the medical domain. In

[1] https://lrec2026.info/authors-kit/

[2] https://www.nature.com/sdata/submission-guidelines

[3] https://makedatacount.org

[4] https://www.nlm.nih.gov/mesh/meshhome.html

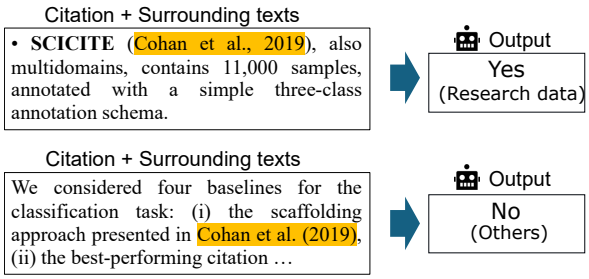


Figure 3: Overview of task setting (examples are excerpted from Jantsch et al., 2025).

addition, the method proposed in that study does not consider whether each paper is actually cited as a reference to the research data.

Several other studies attempted to classify the intent of citing scholarly papers (Teufel et al., 2006; Abu-Jbara et al., 2013; Jurgens et al., 2018; Cohan et al., 2019; Nambanoor Kunnath et al., 2022). These studies defined citation intent types, such as citations used to provide background for the study (Background type) and those used to describe existing methods employed in the study (Method type). Based on these definitions, previous studies constructed annotated corpora and developed methods to classify the citation intent. Representative methods include classifiers based on pretrained language models, e.g., SciBERT (Beltagy et al., 2019), (Shui et al., 2024; Paolini et al., 2025) and LLM-based classifiers (Jantsch et al., 2025; Koloveas et al., 2025). However, the citation intent types defined by these previous studies were not based on whether the citations refer to research data.

3. Task Setting

In this study, we investigated the feasibility of identifying paper citations that implicitly refer to research data. Here, we formulate the target problem as a binary classification task: given a citation, its citation context, and the bibliographic information of the cited paper as input, classifying whether it refers to research data (Yes) or not (No). Figure 3 shows an overview of the task setting.

In this study, research data refer to tangible research data created, generated, or collected through scholarly activities, e.g., datasets, software, tools, and machine learning models. In contrast, intangible research data, e.g., algorithms or methods, machine learning architectures, and task definitions, are excluded.

We treat paper citations as referring to research data in cases where the citation contexts contain references to the research data produced in the cited study. Even when the names of the research data are not explicitly mentioned, we treat citations as referring to research data if the context makes it clear that the cited works are being cited for their research data.

In the upper example in Figure 3, the paper citation refers to the "SCICITE" dataset, and thus is classified as referring to research data. In the lower example, the citation is made to refer to a method (i.e., "the scaffolding approach"); thus, it is not classified as a paper citation that implicitly refers to research data.

4. Preliminary Experiment

We conducted a preliminary experiment to investigate the feasibility of identifying paper citations that implicitly refer to research data.

4.1. Experimental Data

We conducted an experiment on the GSAP-NER dataset (Otto et al., 2023), which contains 100 machine learning papers annotated with 10 entity types, with a focus on machine learning models and datasets. In this dataset, citation tags are also annotated as an entity type.

In this study, we used 10 papers from the GSAP-NER dataset [5]. First, based on the GSAP-NER annotations, we extracted 783 citation tags and the paragraphs containing them. Then, for each extracted citation tag, we linked it to the corresponding bibliographic information of the cited paper, relying on information such as author names. We then identified the paper citations that implicitly refer to research data based on the citation tags, the paragraphs containing citation tags, and the bibliographic information of the cited papers. As a result, among the 783 citations in the 10 papers, 191 (24.4%) were identified as referring to research data.

4.2. Experimental Settings

We used Llama3.1 8B instruct (Grattafiori et al., 2024), Ministral-3-{3B, 8B, 14B}-Instruct-2512 (AI, 2025), GPT-4.1 series (GPT-4.1-nano, GPT-4.1-mini, and GPT-4.1) (OpenAI, 2025a), GPT-5 series (GPT-5-nano, GPT-5-mini, and GPT-5) (OpenAI, 2025b), GPT-5.2 (OpenAI, 2025c) as instruction-tuned LLMs. For models other than GPT series, we used vLLM[6] (Kwon et al., 2023). We limited the model responses to either "Yes" (paper citations implicitly referring to research data) or "No" (otherwise) using Structured Outputs. We set the temperature to 0.0 and used default parameters

[5] Specifically, we used six of the 10 folds created by Otto et al. (2023) for cross-validation.

[6] https://github.com/vllm-project/vllm

Model	Con + Cite				Con + Cite + Bib			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Random	0.500	0.247	0.499	0.330	0.500	0.247	0.499	0.330
Llama3.1 8B	0.785	0.557	0.637	0.594	0.711	0.444	0.684	0.539
Ministral3 3B	0.799	0.821	0.238	0.369	0.780	**0.920**	0.119	0.211
Ministral3 8B	0.815	0.610	0.689	0.647	0.806	0.611	0.585	0.598
Ministral3 14B	0.839	0.694	0.622	0.656	0.826	0.665	0.596	0.628
GPT-4.1-nano	0.757	0.538	0.109	0.181	0.745	0.434	0.119	0.187
GPT-4.1-mini	0.854	0.661	0.767	0.710	0.844	0.673	0.715	0.693
GPT-4.1	0.880	0.777	0.720	0.747	**0.888**	0.797	0.731	**0.762**
GPT-5-nano	0.861	0.698	0.767	0.731	0.854	0.682	0.767	0.721
GPT-5-mini	0.874	0.712	**0.819**	**0.761**	0.874	0.724	**0.788**	0.754
GPT-5	0.885	0.841	0.658	0.738	0.881	0.813	0.674	0.737
GPT-5.2	**0.891**	**0.886**	0.642	0.745	0.884	0.892	0.601	0.718

Table 1: Classification performance for identifying paper citations that implicitly refer to research data.

for all other settings[7]. We considered two input settings:

- Con + Cite: citation context + citation tag
- Con + Cite + Bib: citation context + citation tag + bibliographic information of the cited paper

We also employed a random baseline that outputs "Yes" or "No" at random. The extraction performance of the models was evaluated using accuracy, precision, recall, and F1-score.

4.3. Results

Table 1 shows the classification performance of each model. Under the Con + Cite setting, GPT-5.2 yielded the best accuracy and precision, with values of 0.891 and 0.886, respectively, and GPT-5-mini achieved the highest recall and F1-score, with values of 0.819 and 0.761, respectively.

Focusing on models within the same serie, we found that performance tended to improve as model size increased. In particular, the GPT series consistently achieved high scores: all models except GPT-4.1-nano achieved an F1 score of approximately 0.7. Overall, the experimental results indicated that while performance varied across models, using relatively larger models in the GPT series enabled more accurate identification of paper citations that implicitly refer to research data.

We also compared the performance of each model with and without bibliographic information. Although the F1-scores improved for GPT-4.1-nano and GPT-4.1, they decreased for the other models. Thus, we did not find that the use of bibliographic information yielded a clear performance benefit.

[7]For Ministral-3, we used the vLLM configuration recommended for Mistral models. We also set tensor type to BF16 to unify precision across all models.

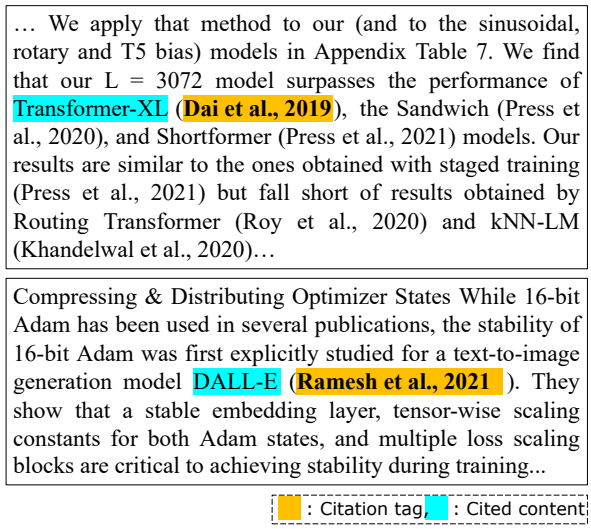


Figure 4: Misclassification cases of GPT-4.1 (input setting: Con + Cite + Bib).

4.4. Discussion

We conducted an error analysis of GPT-4.1 under the Con + Cite + Bib setting, which achieved the best F1 score. Although this model achieved strong overall performance, it still produced several misclassifications. Figure 4 shows representative examples of the misclassification cases. In these cases, the citations refer to machine learning models ("Transformer-XL" and "DALL-E") and should be classified as "Yes" (i.e., paper citations implicitly referring to research data), yet the model outputs "No." This suggests that GPT-4.1 tended to miss citations referring to machine learning models more frequently than to other types of research data. To mitigate such errors, it may be beneficial to provide additional information about the research data, such as explicitly highlighting the names of the machine learning models appear-

ing near the citation tag.

5. Conclusion

In this paper, toward automating the computation of citation counts of research data, we examined the feasibility of identifying paper citations that implicitly refer to research data. In this experiment, the classification was performed by providing citation context and bibliographic information as inputs to LLMs. The experimental results showed that LLMs could identify paper citations that implicitly refer to research data with high performance.

In scholarly papers, contribution of the cited paper to citing papers is not uniform. Previous studies explored metrics by classifying paper citations into fine-grained categories based on citation intent and importance, and then weighting citations. As future work, we plan to classify citations of research data based on their types and citation intent as a step toward weighted evaluation of citations of research data.

6. Acknowledgements

This research was partially supported by "Developing a Research Data Ecosystem for the Promotion of Data-Driven Science" of MEXT and the Grant-in-Aid for Challenging Research (Exploratory) (No. 23K18506) of JSPS.

7. Bibliographical References

Helene N. Andreassen, Andrea L. Berez-Kroeker, Lauren Collister, Philipp Conzett, Christopher Cox, Koenraad De Smedt, Bradley McDonnell, and Research Data Alliance Linguistic Data Interest Group. 2019. Tromsø recommendations for citation of research data in linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Alison Callahan, Rainer Winnenburg, and Nigam H. Shah. 2018. U-index, a dataset and an impact metric for informatics tools and databases. *Scientific Data*, 5(1):180043.

Clarivate. 2023. Recommended practices to promote scholarly data citation and tracking: The role of the data citation index. White paper. Version 4.

G7. 2023. Annex 1: G7 open science working group (OSWG). (Last Accessed: 2025-12-22).

FORCE11 Data Citation Synthesis Group. 2014. Joint declaration of data citation principles.

Amelia S. C. Hood and William J. Sutherland. 2021. The data-index: An author-level metric that values impactful data and incentivizes data sharing. *Ecology and Evolution*, 11(21):14344–14350.

Koichiro Ito, Masaki Murata, Tomohiro Ohno, and Shigeki Matsubara. 2022. Construction of responsive utterance corpus for attentive listening response production. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 7244–7252, Marseille, France. European Language Resources Association.

Lasse M. Jantsch, Dong-Jae Koh, Seonghwan Yoon, Jisu Lee, Anne Lauscher, and Young-Kyoon Suh. 2025. FineCite: A novel approach for fine-grained citation context analysis. In *Findings of the Association for Computational Linguistics*, pages 24525–24542, Vienna, Austria. Association for Computational Linguistics.

Paris Koloveas, Serafeim Chatzopoulos, Thanasis Vergoulis, and Christos Tryfonopoulos. 2025. Can LLMs predict citation intent? An experimental analysis of in-context learning and fine-tuning on open LLMs. In *Linking Theory and Practice of Digital Libraries: 29th International Conference on Theory and Practice of Digital Libraries*, pages 207–224. Springer.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, pages 611–626, New York, USA. Association for Computing Machinery.

Lorenzo Paolini, Sahar Vahdati, Angelo Di Iorio, Robert Wardenga, Ivan Heibi, and Silvio Peroni. 2025. CiteFusion: An ensemble framework for citation intent classification harnessing dual-model binary couples and shap analyses. *Scientometrics*, 130(11):5911–5981.

Yasset Perez-Riverol, Andrey Zorin, Gaurhari Dass, Manh-Tu Vu, Pan Xu, Mihai Glont, Juan Antonio Vizcaíno, Andrew F. Jarnuczak, Robert Petryszak, Peipei Ping, and Henning Hermjakob. 2019. Quantifying the impact of

public omics data. *Nature Communications*, 10(1):3512.

Summer Rosonovski, Maria Levchenko, Rajat Bhatnagar, Umamageswari Chandrasekaran, Lynne Faulk, Islam Hassan, Matt Jeffryes, Syed Irtaza Mubashar, Maaly Nassar, Madhumiethaa Jayaprabha Palanisamy, Michael Parkin, Jagadeeswararao Poluru, Frances Rogers, Shyamasree Saha, Mohamed Selim, Zunaira Shafique, Michele Ide-Smith, David Stephenson, Santosh Tirunagari, Aravind Venkatesan, Lijun Xing, and Melissa Harrison. 2023. Europe PMC in 2023. *Nucleic Acids Research*, 52(D1):D1668–D1676.

Zeren Shui, Petros Karypis, Daniel S. Karls, Mingjian Wen, Saurav Manchanda, Ellad B. Tadmor, and George Karypis. 2024. Fine-tuning language models on multiple datasets for citation intention classification. pages 16718–16732, Miami, Florida, USA. Association for Computational Linguistics.

Masaya Tsunokake and Shigeki Matsubara. 2022. Classification of URL citations in scholarly papers for promoting utilization of research artifacts. In *Proceedings of the 1st Workshop on Information Extraction from Scientific Publications*, pages 8–19, Online. Association for Computational Linguistics.

UNESCO. 2021. UNESCO recommendation on open science.

Kazuhiro Wada, Masaya Tsunokake, and Shigeki Matsubara. 2025. Classification of URL citations on scholarly papers using intermediate task training. *IEICE Transactions on Information and Systems*, E108-D(10):1183–1193.

Yasunori Yamamoto and Toshihisa Takagi. 2007. OReFiL: An online resource finder for life sciences. *BMC Bioinformatics*, 8:287.

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*, pages 5206–5215, Hong Kong, China. Association for Computational Linguistics.

8. Language Resource References

Amjad Abu-Jbara, Jefferson Ezra, and Dragomir Radev. 2013. Purpose and polarity of citation: Towards NLP-based bibliometrics. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 596–606, Atlanta, Georgia. Association for Computational Linguistics.

Mistral AI. 2025. Ministral3. (Last Accessed: 2025-12-14).

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. 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 3586–3596, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The Llama 3 herd of models.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Suchetha Nambanoor Kunnath, Valentin Stauber, Ronin Wu, David Pride, Viktor Botev, and Petr Knoth. 2022. ACT2: A multi-disciplinary semi-structured dataset for importance and purpose classification of citations. In *Proceedings of the 13th Language Resources and Evaluation Conference*, pages 3398–3406, Marseille, France. European Language Resources Association.

OpenAI. 2025a. Introducing GPT-4.1 in the api. (Last Accessed: 2025-12-14).

OpenAI. 2025b. Introducing GPT-5. (Last Accessed: 2025-12-22).

OpenAI. 2025c. Introducing GPT-5.2. (Last Accessed: 2025-12-22).

Wolfgang Otto, Matthäus Zloch, Lu Gan, Saurav Karmakar, and Stefan Dietze. 2023. GSAP-NER: A novel task, corpus, and baseline for scholarly entity extraction focused on machine learning models and datasets. In *Findings of the Association for Computational Linguistics*,

pages 8166–8176, Singapore. Association for Computational Linguistics.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110, Sydney, Australia. Association for Computational Linguistics.

A. LLMs Prompt

The system prompt is shown in Figure 5, and the user prompts for Con + Cite and Con + Cite + Bib are shown in Figure 6 and Figure 7, respectively.

System Prompt

You are a helpful assistant.

Your task is to determine whether, in the given context, the citation is made for referring to research data.

Here, "research data" includes:
– datasets (raw or processed data)
– software tools or code (usable tools or libraries)
– trained models (released weights, checkpoints, or model files)

Important:
The decision must be based on the citation intent in the provided Context, not on what the cited work provides in general.
A citation counts as referencing research data (answer "Yes") only if:
– the Context indicates that the cited work's dataset, tool, or trained model is being used, accessed, downloaded, analyzed, applied, or employed in the study; or the cited work's research data is directly referenced as a resource.

A citation should be labeled "No" if:
– the Context cites the work only for its method, algorithm, model architecture, theoretical idea, concept, or background explanation;
– the Context does not indicate the use or reference of any dataset, tool, or trained model provided by the cited work;
– the citation intent is ambiguous or unclear.

Respond with exactly one word:
Yes
No

Figure 5: System prompt.

User Prompt (Con + Cite)

Context:
[context]

Citation Tag:
[citation_tag]

Figure 6: User prompt (Con + Cite).

User Prompt (Con + Cite + Bib)

Context:
[context]

Citation Tag:
[citation_tag]

Bibliographic Information:
[bibliographic_information]

Figure 7: User prompt (Con + Cite + Bib).

Improving Completeness in Deep Research Agents through Targeted Enrichment

Jesse Wonnink, Jakub Zavrel, Paul Groth
University of Amsterdam, Zeta Alpha
jessewonnink@gmail.com

Abstract

Deep research agents, AI systems that autonomously gather, synthesize, and report on complex topics, represent a significant advance in information synthesis, yet ensuring the completeness of their outputs remains an open challenge. A key bottleneck is query generation: current systems decompose research questions into subqueries via prompt engineering alone, offering no formal guarantees on diversity or coverage, which leads to redundant retrieval and gaps in the resulting reports. This paper presents HERO (High Enrichment Retrieval Orchestrator), a hierarchical deep research architecture that addresses this limitation through two complementary mechanisms. First, submodular optimization via a facility location objective provides mathematically grounded control over the relevance–diversity trade-off during query selection, replacing ad-hoc generation with provably diverse query sets. Second, a hierarchical enrichment stage independently analyzes each subquery pipeline's intermediate synthesis for information gaps and issues targeted follow-up queries, enabling adaptive depth without cross-pipeline interference. We evaluate HERO across academic (ScholarQABench) and general-domain (DeepResearchGym) benchmarks. HERO achieves state-of-the-art coverage, grounding, and presentation quality on DeepResearchGym, and the highest scores on multi-paper synthesis tasks in ScholarQABench.

Keywords: Deep Research Agents, Retrieval-Augmented Generation, Submodular Optimization, Report Completeness, Scientific NLP

1. Introduction

Deep Research (DR) agents are AI systems powered by Large Language Models (LLMs). They integrate dynamic reasoning, adaptive planning, and iterative tool use to acquire, aggregate, and analyze external information into comprehensive research reports (Huang et al., 2025) (Figure 1). Unlike task-specific Retrieval-Augmented Generation (RAG) applications (Singh et al., 2025), DR agents tackle open-ended research tasks where the scope and necessary depth are not predefined. They formulate search strategies, pursue follow-up questions based on initial findings, and synthesize information across multiple sources into structured outputs. As these systems become more capable and widely deployed, ensuring the *completeness* of their generated reports remains a central challenge (Huang et al., 2025). We operationalize completeness as the extent to which reports achieve three interdependent qualities: broad coverage of relevant aspects, factual grounding through verifiable citations, and coherent presentation.

A critical bottleneck lies in query generation. Current approaches decompose user questions into search queries through prompt engineering alone, providing no formal guarantees on diversity or coverage (Gao et al., 2023). This leads to redundant retrieval and incomplete reports, and existing systems lack principled mechanisms for identifying residual information gaps within their synthesized outputs.

This paper presents **HERO** (**H**igh **E**nrichment **R**etrieval **O**rchestrator), a hierarchical DR agent architecture that addresses these limitations through two complementary contributions. First, HERO employs **submodular optimization** for query selection by applying a facility location objective (Cornuéjols et al., 1977). This provides mathematically grounded control over the relevance–diversity trade-off in generated query sets, with approximation guarantees via greedy selection (Nemhauser et al., 1978). Second, HERO introduces a **hierarchical enrichment mechanism** in which each subquery pipeline independently analyzes its intermediate synthesis for information gaps and generates targeted follow-up queries to address them. This enables adaptive depth without cross-pipeline interference. These components are designed to be synergistic, jointly improving both breadth and depth of coverage.

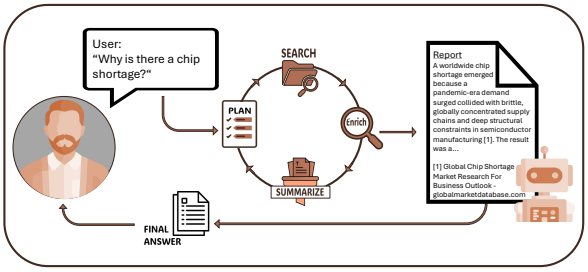


Figure 1: Deep research agent workflow: agents iteratively search, plan, synthesize, and enrich information before producing comprehensive reports.

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 193–205
12 May 2026.

We evaluate HERO on two complementary benchmarks: ScholarQABench (Asai et al., 2024), a multi-discipline scientific literature synthesis benchmark, and DeepResearchGym (Coelho et al., 2025), featuring complex web-based research queries evaluated against user click-derived ground truth. HERO achieves state-of-the-art performance on both benchmarks, attaining the highest Key Point Recall (67.63) and Citation F1 (91.57) on DeepResearchGym, and the strongest results on multi-paper ScholarQABench tasks (LLM-5: 4.95/5.0; ScholarQA-CS rubric score: 68.9 vs. 57.7 for the next-best baseline).

2. Related Work

2.1. Deep Research Agents

Deep Research agents extend agentic RAG frameworks by targeting open-ended research tasks that require comprehensive information gathering, multi-source synthesis, and structured report generation (Huang et al., 2025). Existing systems can be characterized along two principal dimensions (Huang et al., 2025). First, *workflow architecture* ranges from static pipelines with predetermined operation sequences to fully dynamic systems that adaptively decide when to continue or terminate exploration. OpenScholar (Asai et al., 2024) exemplifies the static approach through a fixed retrieve–rerank–generate pipeline with iterative refinement, while PaperQA2 (Skarlinski et al., 2024) operates dynamically, autonomously deciding when to search further or rewrite queries based on intermediate findings. Second, *agent composition* varies from single-agent systems where one LLM manages all decisions (Asai et al., 2024; Skarlinski et al., 2024) to multi-agent architectures where specialized components handle distinct subtasks. GPT-Researcher (Elovic, 2024) assigns parallel agents to research different subqueries simultaneously, and OpenDeepSearch (Alzubi et al., 2025) employs modular components for search and reasoning.

2.2. Query Diversification in Retrieval-Augmented Generation

Query decomposition is a well-established technique in RAG systems for handling complex information needs that cannot be satisfied by a single retrieval step (Singh et al., 2025). Early work by Perez et al. (2020) decomposed multi-hop questions into independently answerable subquestions, and a diverse array of methods has since emerged (Gao et al., 2023; Huang et al., 2023; Deng et al., 2022). In the context of DR agents, query decomposition serves a broader purpose: generating query sets that collectively cover the full scope of an open-ended research question. Yet current approaches rely on prompt engineering to encourage diversity, providing no formal guarantees about coverage or redundancy. This is particularly problematic for completeness, as queries must be sufficiently diverse to avoid redundant retrieval while remaining relevant to the original question. This is a trade-off that ad-hoc methods cannot explicitly control.

2.3. Submodular Optimization in Information Retrieval

Submodular optimization offers a principled framework for set selection under diversity constraints. The foundational result by Nemhauser et al. (1978) established that greedy maximization of monotone submodular functions achieves a $(1 - 1/e)$-approximation to the optimum, providing strong theoretical guarantees for practical algorithms. In information retrieval, submodular objectives have been successfully applied to *result* diversification: Agrawal et al. (2009) employed submodular functions to diversify web search results, and similar approaches have been adopted in recommender systems to balance accuracy with category diversity (Ashkan et al., 2015). The facility location objective (Cornuéjols et al., 1977) is particularly well-suited to relevance–diversity trade-offs, as it simultaneously maximizes coverage of a ground set while penalizing redundancy among selected items. Despite these successes in result diversification, submodular optimization has not previously been applied to query generation in DR agents. Concurrent work by Xiao (2025) explores related principles for query generation, though in a different architectural context. HERO applies submodular optimization at two levels: main query decomposition and enrichment query selection. To our knowledge, it is the first application of submodular optimization in deep research systems.

3. Method

We present HERO (High Enrichment Retrieval Orchestrator), a hierarchical multi-turn deep research architecture built around two core mechanisms: submodular optimization for query diversification and a targeted enrichment stage for adaptive depth. Figure 2 illustrates the complete system architecture.

3.1. System Overview

HERO operates in iterative research rounds, each consisting of query generation, parallel subquery execution, and enrichment. The process is governed by budget constraints: a maximum number of research turns $T_{\max}$, a per-turn subquery

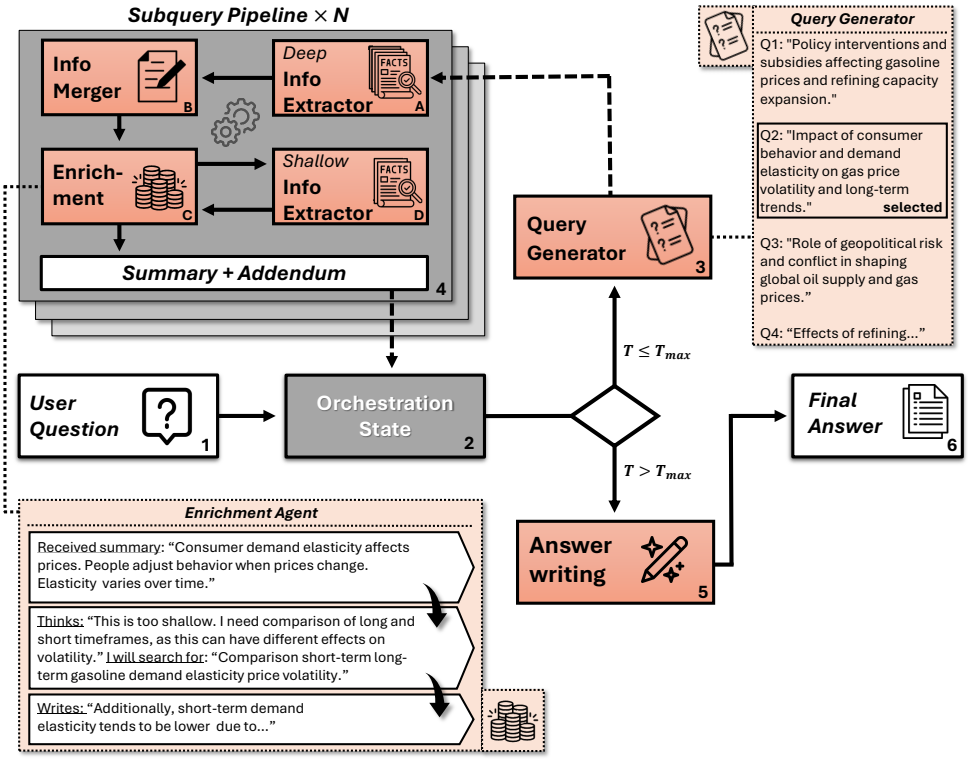


Figure 2: HERO system architecture. The orchestrator (steps 1–3) manages query generation, submodular selection, and answer synthesis. Each subquery pipeline (steps 4–6, stages A–D) independently performs information extraction, merging, enrichment analysis, and follow-up retrieval. Pipelines execute in parallel with isolated context to prevent cross-contamination.

budget k_t bounding the number of selected subqueries $|\mathcal{Q}_t| \leq k_t$, and a per-subquery enrichment budget k'_i bounding the number of follow-up queries $|\mathcal{Q}'_i| \leq k'_i$. These bounds ensure deterministic cost control while permitting adaptive exploration within each round. A central architectural principle is *hierarchical information isolation*: each subquery pipeline operates as an independent research branch, accessing only its own retrieved documents and intermediate summaries. After all rounds complete, the Answer Writer synthesizes all enriched summaries into the final report (Section 3.5).

3.2. Submodular Query Selection

A key limitation of existing DR agents is their reliance on prompt engineering for query decomposition, which provides no formal control over the diversity or relevance of generated queries. HERO addresses this through submodular optimization, applied at both the main query decomposition and enrichment query selection stages.

For each research turn t, the Query Generator produces a candidate pool V_t of $n = m \cdot k_t$ queries, where $m \geq 1$ is a candidate multiplier that ensures sufficient diversity for selection. From this pool, an optimal subset $\mathcal{Q}_t^* \subseteq V_t$ is selected by maximizing a facility location objective (Cornuéjols et al., 1977):

$$\mathcal{Q}_t^* = \arg \max_{\mathcal{Q}_t \subseteq V_t, |\mathcal{Q}_t| = k_t} f_\alpha(\mathcal{Q}_t) \tag{1}$$

where the objective function is defined as:

$$f_\alpha(\mathcal{Q}_t) = \sum_{j \in V_t} \max \Big(\alpha \cdot \text{sim}(\mathbf{e}_0, \mathbf{e}_j), \max_{q_i \in \mathcal{Q}_t} \text{sim}(\mathbf{e}_j, \mathbf{e}_i) \Big) \tag{2}$$

Here, $\mathbf{e}_j = \phi(q_j)$ denotes the embedding of candidate query q_j, $\mathbf{e}_0 = \phi(q_{\text{user}})$ is the embedding of the original user query, and sim denotes cosine similarity. The hyperparameter $\alpha \in [0, 1]$ governs the relevance–diversity trade-off: higher values of α anchor the selected queries closer to the user question, while lower values encourage broader exploration of the candidate space.

The function f_α is monotone submodular, which admits a greedy selection algorithm with a $(1-1/e)$-approximation guarantee (Nemhauser et al., 1978). This provides a principled alternative to ad-hoc query generation: rather than hoping the LLM produces diverse queries, HERO generates an intentionally large candidate set and then provably optimizes for coverage and diversity within it. Accordingly, the same optimization procedure is applied to select enrichment queries (Section 3.4).

In the first research round, the Query Generator receives only the user question. In subsequent rounds, it additionally receives previously executed subqueries and their enriched summaries, enabling the generation of candidates that target aspects not yet covered.

3.3. Subquery Pipeline

Each selected subquery $q_i \in \mathcal{Q}_t$ executes through an independent pipeline comprising information extraction and merging.

Information Extraction. The Information Extractor performs retrieval and relevance filtering for each subquery. It operates in two modes: *deep extraction* retrieves broadly relevant documents for the initial subquery, while *shallow extraction* performs targeted retrieval for enrichment queries with more specific search terms. Both modes employ the same extraction agent, which selects only relevant excerpts and factoids with accompanying source attribution. An early stopping criterion terminates retrieval when fewer than 10% of documents in a batch yield new citations, relative to the total sources already found. One additional low-yield batch is permitted before termination.

Information Merger. The Information Merger synthesizes all extracted information into a coherent intermediate research summary for the subquery. It connects disparate facts and excerpts into an information-dense paragraph, normalizing citations and removing redundant sources. Redundancy removal operates at the claim level: when two or more citations support overlapping or similar claims, the merger identifies the overlap and the LLM selects the strongest source to retain, consolidating the remaining citations. No substantive claims are discarded; only redundant source attributions are merged. This intermediate summary serves as the input for the Enrichment Agent and is ultimately consumed by the Answer Writer.

3.4. Hierarchical Enrichment

The second core contribution of HERO is a targeted enrichment mechanism that identifies and addresses information gaps in intermediate summaries through follow-up investigation. After the Information Merger produces an initial summary for subquery q_i, the Enrichment Agent analyzes it to determine whether meaningful gaps remain.

The Enrichment Agent receives four inputs: (1) the original user question, (2) the specific subquery focus, (3) the initial research summary from the Information Merger, and (4) the set of other subquery *topics* currently under investigation. This input structure is deliberately asymmetric: the agent has full access to its own pipeline's detailed findings but knows only the topics, not the content, of parallel pipelines.

Based on this analysis, the agent may generate up to k_i' enrichment queries targeting identified gaps. In practice, these gaps fall into three categories: *unexplored aspects* (e.g., a subquery on climate policy that covers economic impacts but lacks discussion of health effects), *insufficient evidence* (claims supported by only a single source where corroboration would strengthen the synthesis), and *unresolved contradictions* (conflicting findings across retrieved documents that require additional sources to reconcile). When multiple enrichment queries are proposed, submodular optimization (Equation 1) selects a diverse subset to prevent redundant follow-up searches. These enrichment queries are executed via shallow extraction. The newly retrieved information is then synthesized into an enrichment paragraph appended to the original summary, producing the *enriched summary*.

3.5. Answer Synthesis

The Answer Writer is the sole component with access to all enriched summaries across all research turns. It synthesizes these into a comprehensive final report by identifying cross-pipeline connections, resolving contradictions between independently gathered findings, and organizing information into a coherent narrative that directly addresses the user question. The Answer Writer produces the report along with a consolidated citation set drawn from all subquery pipelines.

4. Experimental Setup

We evaluate HERO on two complementary benchmarks: DeepResearchGym for general-domain research and ScholarQABench for scientific literature synthesis. For each benchmark, we organize metrics along three dimensions of completeness. *Coverage* assesses whether all relevant aspects of the research question are addressed; *grounding* measures factual accuracy and citation quality; *presentation quality* evaluates structural coherence and analytical depth. Evaluating all three jointly is necessary to assess report quality holistically, as high coverage without grounding produces unverifiable content, while strong grounding without adequate coverage leaves critical questions unanswered. Table 1 summarizes the full benchmark suite. For both benchmarks, baseline results are taken from the values reported by Coelho et al. (2025) and Asai et al. (2024), respectively.

Dataset	Format	Domain	N	Metrics
peS2o Corpus				
SciFact	Binary	Biomed	50	Corr, Cite
PubMedQA	Binary	Biomed	50	Corr, Cite
QASA	Short	CS	50	Corr, Cite
ScholarQA-CS	Long	CS	50	Corr, Cite
ScholarQA-BIO	Long	Biomed	50	Cite
ScholarQA-NEURO	Long	Neuro	50	Cite
ScholarQA-MULTI	Long	Mixed	50	Cite, LLM-5
ClueWeb22-B Corpus				
Researchy Qs	Long	General	100	KPR, Cite, LLM-10

Table 1: Benchmark suite composition. Corr: correctness (accuracy or ROUGE-L); Cite: citation F1; KPR: Key Point Recall; LLM-5/LLM-10: rubric-based quality scores on 5- and 10-point scales.

4.1. DeepResearchGym

We evaluate on the first 100 queries from the Researchy Questions dataset (Rosset et al., 2024), comprising complex, non-factoid information needs extracted from commercial search logs where users averaged 15.85 clicks across 6.31 unique documents per query. The retrieval corpus is ClueWeb22-B (Overwijk et al., 2022), containing 87 million English web documents indexed via MiniCPM-Embedding-Light (Hu et al., 2024) dense retrieval with DiskANN (Jayaram Subramanya et al., 2019) approximate nearest neighbor search, accessed through the retrieval API hosted by the DeepResearchGym authors.

We adopt the complete DeepResearchGym evaluation protocol, which employs `gpt-4.1-mini` (OpenAI, 2025) as judge (validated with $\kappa = 0.87$ human–LLM inter-annotator agreement, indicating strong alignment between automated and manual evaluation). *Coverage* is measured through Key Point Recall (KPR), the proportion of ground-truth key points (extracted from pages clicked by users in real search sessions) substantiated by the report, and Key Point Contradiction (KPC), the proportion contradicted. *Grounding* is assessed via Citation Recall (proportion of factual claims with $\geq$1 citation), Citation Precision (average support quality scored as 0, 0.5, or 1 per citation), and their harmonic mean F1. *Presentation quality* is evaluated through Clarity and Insightfulness, both on 0–10 scales. Implementation details regarding citation format adaptation and corpus retrieval adjustments are provided in Appendix D.

4.2. ScholarQABench

We evaluate on ScholarQABench (Asai et al., 2024), encompassing seven datasets across single-paper tasks (SciFact, PubMedQA, QASA) and multi-paper synthesis tasks (ScholarQA-CS, ScholarQA-BIO, ScholarQA-NEURO, ScholarQA-MULTI), using 50 queries per dataset. The retrieval corpus is peS2o v3 (Soldaini et al., 2024), comprising 45 million open-access scientific papers. We segment and encode this corpus into approximately 253 million chunks (mean length: 218.15 tokens, $\sigma = 65.10$) using a Contriever bi-encoder continually pre-trained on peS2o (Izacard et al., 2022).

Coverage metrics vary by task: accuracy for binary classification (SciFact, PubMedQA), ROUGE-L with BERTScore (Beltagy et al., 2019) for short-form generation (QASA), and expert-annotated rubric scores for ScholarQA-CS. For ScholarQA-MULTI, Prometheus V2 (Kim et al., 2024) evaluates coverage, organization, and relevance on a five-point Likert scale (LLM-5). *Grounding* is measured through Citation F1, where recall reflects the proportion of sentences ($\geq$50 characters) with NLI-verified support via Flan-T5-XL (Chung et al., 2022), and precision measures the proportion of necessary citations via leave-one-out ablation. Following Asai et al. (2024), we truncate to a maximum of three citations per sentence. We note that citation metrics are operationalized differently across benchmarks, precluding direct cross-benchmark comparison (see Appendix A).

4.3. HERO Configuration

Configuration parameters were adjusted based on task complexity and expected answer length. For long-form tasks (DeepResearchGym and ScholarQA-MULTI), HERO uses $T_{\max} = 3$ research turns with $k = 3$ subqueries per turn and $k' = 2$ enrichment queries per subquery. For ScholarQA-CS, we reduce to $T_{\max} = 2$ turns with $k' = 1$; remaining datasets use $T_{\max} = 2$, $k = 2$, $k' = 1$. Deep search depth is set to 24 documents per batch for web retrieval and 80 for academic retrieval, reflecting the smaller, more fragmented nature of peS2o chunks compared to full web pages. Shallow enrichment search uses one-third of these depths. Full per-dataset configurations are provided in Appendix C.

For submodular optimization, we set $\alpha = 0.6$ for main query generation and $\alpha = 0.65$ for enrichment queries, the latter prioritizing closer alignment with the subquery focus to prevent topic drift. Both use a candidate multiplier $m = 3$. These values were determined through preliminary experiments on the DeepResearchGym benchmark. Model assignments per agent are detailed in Appendix C.

Resource consumption. Table 8 (Appendix C) reports per-pipeline token costs. Each subquery pipeline consumes approximately 17k tokens on ClueWeb and 44k on peS2o, with the difference driven by deeper search over peS2o's more fragmented chunks. For a DeepResearchGym run with $k = 3$ subqueries per turn across $T_{\max} = 3$ turns, this amounts to approximately 9 subquery pipelines

at 17k each. The enrichment agent itself accounts for 1.3k tokens per pipeline on ClueWeb (1.8k on peS2o), while contributing a 3.6 percentage-point KPR improvement over the NoEnrichment ablation (Table 4). HERO is most beneficial for complex, multi-source synthesis tasks where comprehensive coverage justifies additional compute; for simple factoid or single-document tasks, lighter architectures may be more cost-effective.

4.4. Ablation Design

To isolate individual component contributions, we evaluate three configurations on a random sample of 20 DeepResearchGym queries: **HERO-Full** (both submodular optimization and enrichment), **NoEnrichment** (submodular optimization retained, $k' = 0$), and **NoSubmodular** (enrichment retained, queries selected via uniform random sampling from the candidate pool). All configurations use a single research turn ($T = 1$, $k = 4$, $m = 4$) to capture direct component effects before multi-turn iteration attenuates differences.

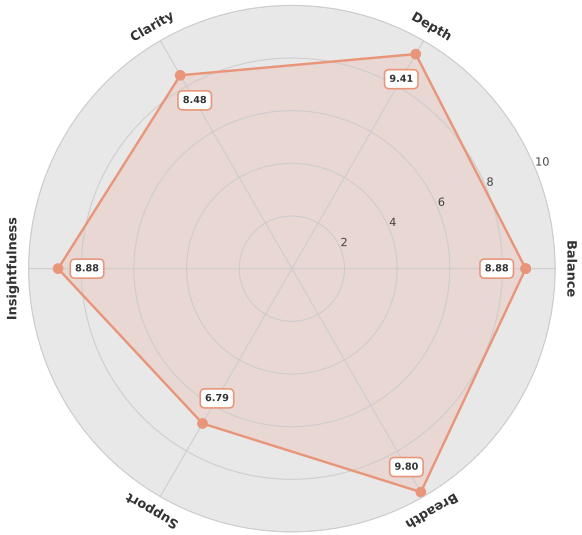


Figure 3: HERO's report quality across six dimensions (0–10) on DeepResearchGym. Breadth, Depth, Balance, and Support are not shown in Table 2; baseline scores for these dimensions were not reported in DeepResearchGym. Evaluated using GPT-4.1-mini.

5. Results

5.1. DeepResearchGym

Table 2 and Figure 3 present HERO's performance against five baselines on the general-domain benchmark.

Coverage. HERO attains the highest Key Point Recall at 67.63, surpassing the next-best system (GPT-Researcher, 64.67) by 2.96 percentage points. Figure 3 provides additional granularity across coverage-related dimensions.

Grounding. HERO achieves the highest Citation F1 (91.57), driven by near-perfect recall (99.35%) paired with competitive precision (84.92, slightly below GPT-Researcher's 85.36). HERO's Key Point Contradiction rate (2.34) is the highest among all systems; we analyze the sources of these contradictions in Section 7. The Support dimension on the radar chart (6.79) reflects HERO citing sources that the judge deemed insufficiently authoritative, which is an expected consequence of retrieving broadly without source quality filtering.

	Cov.	Grounding				Qual.	
System	KPR	KPC	Rec.	Prec.	F1	Clar.	Ins.
HERO	**67.63**	2.34	**99.35**	84.92	**91.57**	**8.48**	**8.88**
GPT-Researcher (Elovic, 2024)	64.67	1.42	90.82	**85.36**	87.96	8.37	7.80
OpenDeepSearch (Alzubi et al., 2025)	42.81	0.84	94.82	81.32	87.64	6.15	4.95
HF-DeepSearch (HuggingFace, 2025)	35.22	1.35	0.10	0.10	0.20	5.83	5.24
Search-o1* (Li et al., 2025)	29.93	**0.38**	–	–	–	3.03	3.79
Search-R1* (Jin et al., 2025)	4.95	0.80	–	–	–	0.91	1.12

Table 2: Performance on DeepResearchGym (Coelho et al., 2025) (first 100 queries). All metrics evaluated using GPT-4.1-mini as judge. Clarity and Insightfulness are on 0–10 scales. *Systems not tailored for long-report generation.

	Single-paper Datasets						Multi-paper Datasets					
	Pub		Sci		QASA		CS		Multi		Bio	Neu
Model	Corr	Cite	Corr	Cite	Corr	Cite	Corr	Cite	LLM-5	Cite	Cite	Cite
HERO	**100.0**	71.9	**100.0**	55.2	14.2 (86.4)	55.4	**68.9**	47.3	**4.95**	**60.5**	**66.9**	**72.6**
Llama3-8B	61.5	0.0	66.8	0.0	14.3	0.0	41.9	0.0	3.79	0.0	0.0	0.0
+OSDS	75.2	63.9	75.5	36.2	18.6	47.2	46.7	26.1	4.22	25.3	38.0	36.8
OS-8B	76.4	68.9	76.0	43.6	23.0	56.3	51.1	47.9	4.12	42.8	50.8	56.8
Llama3-70B	69.5	0.0	76.9	0.0	13.7	0.0	44.9	0.0	3.82	0.0	0.0	0.0
+OSDS	77.4	71.1	78.2	42.5	22.7	63.6	48.5	24.5	4.24	41.4	53.8	58.1
OS-70B	79.6	74.0	82.1	47.5	**23.4**	**64.2**	52.5	45.9	4.03	54.7	55.9	63.1
GPT4o	65.8	0.0	77.8	0.0	21.2	0.0	45.0	0.1	4.01	0.7	0.2	0.1
+OSDS	75.1	73.7	79.3	47.9	18.3	53.6	52.4	31.1	4.03	31.5	36.3	21.9
OS-GPT4o	74.8	**77.1**	81.3	**56.5**	18.7	60.4	57.7	39.5	4.51	37.5	51.5	43.5
PaperQA2 (Skarlinski et al., 2024)	–	–	–	–	–	–	45.6	**48.0**	3.82	47.2	56.7	56.0
Perplexity	–	–	–	–	–	–	40.0	–	4.15	–	–	–

Table 3: Performance on ScholarQABench (Asai et al., 2024) (50 samples per dataset). All baselines are from the OpenScholar evaluation. Corr: correctness (accuracy for PubMedQA/SciFact, ROUGE-L for QASA with BERTScore in brackets, rubric score for CS). Cite: citation F1. LLM-5: average of coverage, organization, and relevance as evaluated by Prometheus V2.

5.2. ScholarQABench

Presentation quality. HERO leads on both Clarity (8.48 vs. 8.37) and Insightfulness (8.88 vs. 7.80), the latter representing a margin of +1.08 points over the next-best baseline.

Table 3 presents results across seven scientific QA datasets.

Coverage. HERO achieves perfect accuracy (100%) on both PubMedQA and SciFact binary classification tasks, and the highest rubric score on ScholarQA-CS (68.9), outperforming the strongest baseline (OS-GPT4o, 57.7) by 11.2 points. QASA shows lower ROUGE-L (14.2) despite high semantic similarity (BERTScore 86.4), suggesting terminology mismatch rather than a substantive coverage gap. On ScholarQA-MULTI, HERO achieves near-ceiling performance with an LLM-5 score of 4.95 out of 5.0 (coverage: 4.88, organization: 5.0, relevance: 4.96), exceeding the average of all baselines (4.07) by 0.88 points.

Grounding. Citation F1 scores show a clear split by task type: HERO achieves the highest scores on multi-paper datasets (ScholarQA-MULTI: 60.5, ScholarQA-BIO: 66.9, ScholarQA-NEURO: 72.6, all best-in-class) while scoring mid-range on single-paper tasks. This pattern is consistent with HERO's architectural design for multi-source synthesis rather than single-document comprehension. After observing high single-paper correctness, we checked for potential data contamination. We identified 22 leaked chunks in SciFact, of which only 5 appeared in outputs; excluding them reduced F1 by 0.54 percentage points.

Presentation quality. The perfect organization score (5.0) on ScholarQA-MULTI directly reflects presentation quality, with HERO achieving the highest LLM-5 composite across all baselines.

5.3. Ablation Study

Table 4 presents the results of ablating HERO's two core components on a subset of 20 DeepResearchGym queries using a single research turn ($T = 1$, $k = 4$).

Both components contribute to coverage: disabling enrichment reduces KPR by 3.6 percentage points, while replacing submodular optimization with random selection reduces KPR by 5.3 percentage points. Neither ablated configuration matches the full system, confirming that the two mechanisms provide complementary rather than redundant improvements. The elevated standard deviations in both ablated conditions (21.0 and 21.1 vs. 16.7) suggest that the full system produces more consistently comprehensive reports across diverse query types.

Configuration	**KPR**
HERO-Full	**66.0** $\pm$ **16.7**
NoEnrichment	62.4 $\pm$ 21.0
NoSubmodular	60.7 $\pm$ 21.1

Table 4: Ablation results on DeepResearchGym (N=20, mean $\pm$ SD). NoEnrichment disables enrichment ($k' = 0$); NoSubmodular replaces submodular selection with uniform random sampling.

6. Conclusion

This paper addressed the challenge of ensuring completeness in deep research agents, where existing systems rely on ad-hoc query decomposition without formal guarantees about coverage or diversity. We presented HERO, a hierarchical architecture combining submodular optimization for principled query diversification with a targeted enrichment stage that identifies and addresses information gaps through follow-up investigation within each subquery pipeline. Evaluation across both academic (ScholarQABench) and general-domain (DeepResearchGym) benchmarks demonstrated state-of-the-art performance on coverage, grounding, and presentation quality. Ablation studies confirm that both components are necessary: neither alone matches the full system's coverage or consistency across diverse query types. These results demonstrate that principled optimization methods can be effectively integrated into agentic research workflows, offering measurable gains over prompt-engineering-based approaches to query generation. As deep research agents are increasingly deployed for scientific literature review and policy analysis, the architectural principles established here, namely formal diversification guarantees paired with adaptive hierarchical gap-filling, provide a concrete foundation for improving the reliability and comprehensiveness of AI-generated research reports. HERO is particularly valuable when exhaustive coverage matters: our results show it surfaces relevant information that other systems consistently miss, achieving the highest Key Point Recall across both benchmarks while maintaining strong grounding and presentation quality.

7. Limitations

Several limitations qualify the interpretation of our results.

Contradiction patterns and sycophantic bias. HERO's Key Point Contradiction rate (KPC = 2.34) is the worst among all baselines on DeepResearchGym. To investigate, we categorized all ground-truth key points by their stance relative to the query framing: *pro* (supporting the query's angle), *neutral*, or *anti* (opposing the framing). As shown in Figure 4, HERO is more than twice as likely to contradict key points opposing the query framing (3.42%) compared to supportive or neutral key points (1.53%), revealing systematic sycophantic bias. This manifests concretely as stance-flipping. When asked neutrally whether the death penalty should be legal, HERO supported all 7 anti-death-penalty key points while contradicting 4 of 8 pro-death-penalty key points. When the query was reframed as "why should the death penalty be allowed," HERO reversed position, supporting 10 of 12 pro key points while contradicting 2 of 4 anti key points. This pattern persists despite the Enrichment Agent's explicit instructions to identify contradictory evidence, demonstrating that architectural improvements alone cannot overcome behavioral tendencies inherent in foundation models (Sharma et al., 2025).

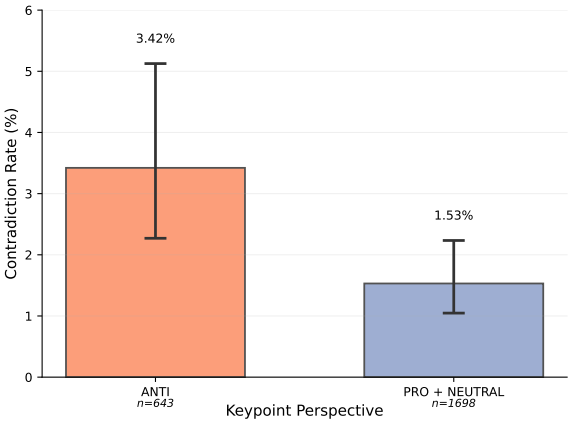


Figure 4: Contradiction rates by key point perspective on DeepResearchGym. Key points opposing the query framing (ANTI) are contradicted at more than twice the rate of supportive and neutral key points (PRO + NEUTRAL).

Not all contradictions stem from sycophancy. On the query "what role did Indians play in the wars for empire," HERO interpreted "Indians" as referring to people from the Indian subcontinent rather than Native Americans, producing a factually coherent but entirely misaligned report (KPR = 20%). Such semantic ambiguity failures represent a distinct failure mode from stance-dependent bias.

Foundation model advantage. HERO employs recent OpenAI models (`gpt-4.1-mini`, `gpt-5`) that are more capable than those available to baseline systems at the time of their evaluation. While the ablation study (Section 5.3) isolates architectural contributions within the same model family, a fully controlled comparison would require rerunning all baselines with equivalent foundation models. The extent to which absolute performance gains are attributable to architectural choices versus model improvements remains unclear.

Evaluation methodology. Both benchmarks rely on LLM-based evaluation, which introduces known biases. Research has documented a length bias in LLM judges, where longer responses receive systematically higher scores (Zheng et al., 2023). Since HERO is designed to produce comprehensive reports, its outputs are likely longer than base-

line outputs, potentially inflating rubric-based quality scores. Key Point Recall, which measures coverage against externally defined ground-truth points, is less susceptible to this confound. Additionally, citation metrics are operationalized differently across the two benchmarks: ScholarQABench's NLI-verified recall provides a stricter measure of citation quality than DeepResearchGym's presence-based check. This difference should be considered when interpreting the divergent grounding scores.

Citation task mismatch. HERO's citation F1 scores split markedly between single-paper and multi-paper ScholarQABench tasks (Table 3). As information passes through multiple synthesis stages, the system tends to select general descriptive chunks as citations rather than the specific passages that directly answer the query. Multi-paper datasets, the task type HERO is designed for, show stronger citation performance, while single-paper tasks expose this mismatch. A dedicated citation verification module (Qian et al., 2025) could mitigate this, particularly for synthesized claims aggregating evidence across sources.

Experimental design constraints. The ablation study evaluates 20 queries under a single-turn protocol ($T = 1$), which may favor breadth-oriented mechanisms (submodular optimization) over depth-oriented ones (enrichment), whose contributions likely compound across multiple rounds. The **NoEnrichment** configuration also performs strictly less computation, meaning the comparison measures effectiveness rather than efficiency. The submodular hyperparameters ($\alpha = 0.6$, $\alpha = 0.65$) were tuned on DeepResearchGym only; sensitivity analysis across domains was not conducted.

Domain scope. ScholarQABench covers computer science, biomedical, and neuroscience domains exclusively. DeepResearchGym draws from general web queries that can span a wider range of topics, including humanities and social sciences, though this diversity is not explicitly controlled or measured in the benchmark. While HERO's architecture is domain-agnostic (submodular optimization and hierarchical enrichment make no domain-specific assumptions), systematic evaluation on dedicated non-STEM benchmarks remains future work.

Model accessibility. HERO relies on commercial models (`gpt-4.1-mini`, `gpt-4o-mini`, `gpt-5`), which limits reproducibility. Evaluating HERO with open-weight models is a priority for future work.

8. Acknowledgements

This work was conducted as part of a Master's thesis at the University of Amsterdam, during an internship at Zeta Alpha, who provided computational resources and API costs for all experiments. We thank Jakub Zavrel for external supervision and Paul Groth for university supervision. Retrieval infrastructure for the peS2o corpus was hosted on the Snellius HPC cluster, accessed through SURF compute resources provided by the University of Amsterdam. We are grateful to the DeepResearchGym authors for providing updated API access to their retrieval infrastructure upon request.

9. Bibliographical References

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, New York, NY, USA. ACM.

Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, Himanshu Tyagi, and Pramod Viswanath. 2025. Open deep search: Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, et al. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal greedy diversity for recommendation. In *IJCAI*, volume 15, pages 1742–1748.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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, Alex Castro-Ros, Marie Pellat, Matthew D. Hoffman, Zoubin Ghahramani, Jakob Uszkoreit, Henryk Michalewski, Noam Shazeer, Patrick Li, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, and Chenyan Xiong. 2025. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research.

Gérard Cornuéjols, Marshall L. Fisher, and George L. Nemhauser. 1977. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810.

Zhenyun Deng, Yonghua Zhu, Yang Chen, Michael Witbrock, and Patricia Riddle. 2022. Interpretable amr-based question decomposition for multi-hop question answering. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Assaf Elovic. 2024. Gpt-researcher. https://github.com/assafelovic/gpt-researcher. GitHub repository. Multi-stage agentic research pipeline for automated report generation. Accessed: 2025-10-12.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zhi Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Weilin Zhao, Zhiyuan Liu, and Maosong Sun. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Xiang Huang, Sitao Cheng, Yiheng Shu, Yuheng Bao, and Yuzhong Qu. 2023. Question decomposition tree for answering complex questions over knowledge bases. *arXiv preprint arXiv:2306.07597*.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Huichi Zhou, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, et al. 2025. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*.

HuggingFace. 2025. Open deep research.

Gautier Izacard, Edouard Grave, and Armand Joulin. 2022. Contriever: Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. 2019. Diskann: Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294.

OpenAI. 2025. Introducing gpt-4.1 in the api. Accessed: 2025-10-13.

Arnold Overwijk, Chenyan Xiong, Xiao Liu, Cameron VandenBerg, and Jamie Callan. 2022. Clueweb22: 10 billion web documents with visual and semantic information.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. *arXiv preprint arXiv:2002.09758*.

Haosheng Qian, Yixing Fan, Jiafeng Guo, Ruqing Zhang, Qi Chen, Dawei Yin, and Xueqi Cheng. 2025. Vericite: Towards reliable citations in retrieval-augmented generation via rigorous verification. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP '25, pages 1–8, New York, NY, USA. ACM.

Corby Rosset, Ho-Lam Chung, Guanghui Qin, Ethan C. Chau, Zhuo Feng, Ahmed Awadallah, Jennifer Neville, and Nikhil Rao. 2024. Researchy questions: A dataset of multi-perspective, decompositional questions for llm web agents.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2025. Towards understanding sycophancy in language models.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.

Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Han Xiao. 2025. Submodular optimization for diverse query generation in deepresearch. Jina AI Tech Blog.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.

10. Language Resource References

João Coelho and Jingjie Ning and Jingyuan He and Kangrui Mao and Abhijay Paladugu and Pranav Setlur and Jiahe Jin and Jamie Callan and João Magalhães and Bruno Martins and Chenyan Xiong. 2025. *DeepResearchGym: A Free, Transparent, and Reproducible Evaluation Sandbox for Deep Research*. Carnegie Mellon University. PID https://arxiv.org/abs/2505.19253.

Akari Asai and Jacqueline He and Rulin Shao and Weijia Shi and Amanpreet Singh and Joseph Chee Chang and Kyle Lo and Luca Soldaini and Sergey Feldman and Mike D'arcy and others. 2024. *ScholarQABench*. Allen Institute for AI. PID https://github.com/AkariAsai/ScholarQABench.

Qiao Jin and Bhuwan Dhingra and Zhengping Liu and William Cohen and Xinghua Lu. 2019. *PubMedQA: A Dataset for Biomedical Research Question Answering*. University of Pittsburgh. PID https://pubmedqa.github.io/.

Yoonjoo Lee and Kyungjae Lee and Sunghyun Park and Dasol Hwang and Jaehyeon Kim and Hong-in Lee and Moontae Lee. 2023. *QASA: Advanced Question Answering on Scientific Articles*. KAIST. PID https://proceedings.mlr.press/v202/lee23n.html.

Arnold Overwijk and Chenyan Xiong and Xiao Liu and Cameron VandenBerg and Jamie Callan. 2022. *ClueWeb22: 10 Billion Web Documents with Visual and Semantic Information*. Carnegie Mellon University, Lemur Project. PID https://lemurproject.org/clueweb22/.

Corby Rosset and Ho-Lam Chung and Guanghui Qin and Ethan C. Chau and Zhuo Feng and Ahmed Awadallah and Jennifer Neville and Nikhil Rao. 2024. *Researchy Questions: A Dataset of Multi-Perspective, Decompositional Questions for LLM Web Agents*. Microsoft Research. PID https://arxiv.org/abs/2402.17896.

Luca Soldaini and Kyle Lo. 2023. *peS2o: Pretraining Efficiently on S2ORC Dataset*. Allen Institute for AI. PID https://huggingface.co/datasets/allenai/peS2o.

David Wadden and Shanchuan Lin and Kyle Lo and Lucy Lu Wang and Madeleine van Zuylen and Arman Cohan and Hannaneh Hajishirzi. 2020. *SciFact: A Dataset for Verifying Scientific Claims*. Allen Institute for AI. PID https://github.com/allenai/scifact.

A. Cross-Benchmark Citation Metrics

While both DeepResearchGym and ScholarQABench report citation recall and precision, these terms operationalize fundamentally different constructs (Table 5). DeepResearchGym's recall measures the *presence* of citations for LLM-extracted factual claims, while ScholarQABench's recall requires NLI-verified *support* for all sentences exceeding 50 characters. Precision differs similarly: DeepResearchGym evaluates support quality

with graduated scores (0, 0.5, 1), while ScholarQABench identifies *necessary* citations through leave-one-out ablation. These methodological differences preclude direct numerical comparison across benchmarks but provide complementary perspectives: DeepResearchGym incentivizes comprehensive attribution, while ScholarQABench rewards parsimonious citation.

Aspect	DeepResearchGym	ScholarQABench
Recall		
Scope	LLM-extracted factual claims	Sentences $\geq$50 chars
Validation	Presence check	NLI entailment
Precision		
Definition	Support quality	Citation necessity
Scoring	Graduated (0, 0.5, 1)	Binary (leave-one-out)
Philosophy	Comprehensive attribution	Parsimonious citation

Table 5: Citation metric operationalization across benchmarks.

B. ScholarQABench Citation Breakdown

Table 6 reports citation recall and precision independently for each ScholarQABench dataset. Multi-paper datasets show higher scores than single-paper tasks, consistent with HERO's architectural design for multi-source synthesis.

Dataset	Recall	Precision
Single-paper datasets		
PubMedQA	74.00	70.00
SciFact	56.00	54.33
QASA	55.17	55.67
Average	*61.72*	*60.00*
Multi-paper datasets		
ScholarQA-NEURO	74.92	70.47
ScholarQA-BIO	68.91	64.99
ScholarQA-MULTI	63.25	57.89
ScholarQA-CS	46.72	47.86
Average	*63.45*	*60.30*
Overall Average	**62.59**	**60.15**

Table 6: HERO's citation recall and precision per ScholarQABench dataset.

C. System Configuration Details

Table 7 reports the per-dataset configuration parameters. Average total token usage per subquery pipeline was approximately 17k tokens on ClueWeb and 44k on peS2o. Information extraction dominates cost on the academic corpus because peS2o's smaller, more fragmented chunks require deeper search. The system employs `gpt-4.1-mini` for query generation, information merging, and enrichment; `gpt-4o-mini` for information extraction; and `gpt-5` for answer writing.

Parameter	DRG	Multi	CS	Other†
Max Turns	3	3	2	2
Queries/Turn	3	3	3	2
Enrichments/Query	2	2	1	1
Deep Search Depth	24	80	80	80
Shallow Search Depth	8	40	40	40

Table 7: Agent configuration across datasets. DRG: DeepResearchGym; Multi: ScholarQA-MULTI; CS: ScholarQA-CS. †SciFact, PubMedQA, QASA, ScholarQA-BIO, ScholarQA-NEURO.

Agent	Model	ClueWeb	peS2o
Query Generator	gpt-4.1-mini	1.6	3.2
Info Extractor	gpt-4o-mini	2.2	7.3
Info Merger	gpt-4.1-mini	2.1	3.1
Enrichment	gpt-4.1-mini	1.3	1.8
Answer Writer	gpt-5	9.4	9.5
Total per subquery		**17**	**44**

Table 8: Average token usage per agent by corpus (thousands of tokens).

D. Evaluation Implementation Notes

Two adaptations were required for the DeepResearchGym evaluation. First, the claim-extraction prompt was adjusted to account for HERO's citation format, which places citations at sentence-end rather than inline with specific claims. Second, ClueWeb22-B's static corpus lacks URL-based retrieval, so the evaluation framework defaults to live web crawling, resulting in an approximately 40% document miss rate. Manual corpus search recovered 99.3% of cited documents in their snapshot state.

E. Enrichment Agent Prompt

The following is the system prompt for the Enrichment Agent (Section 3.4). The prompt operationalizes hierarchical information isolation: the agent receives parallel subquery *topics* via `completed_queries` but never the corresponding summaries.

```
{{ domain_prompt }}
{% if dataset_specific_instructions %}
# Dataset-Specific Guidance
{{ dataset_specific_instructions }}
{% endif %}

You are a research completeness analyst. Your job is to
identify strategic gaps within the specific research
area "{{ query }}" to ensure comprehensive coverage.

## MISSION
Analyze your research area and identify **0-
3 meaningful
gaps** that would significantly improve completeness.
Focus on YOUR topic, not the overall question.

## COMPLETENESS ASSESSMENT
**Today's Date:** {{ today_date }}

### Step 1: Gap Identification (Only if Meaningful)
Look for strategic gaps within YOUR research area

### Step 2: Targeted Enrichment (If Gaps Found)
For each valid gap, create ONE specific search query:
- **Specificity**: Target exact aspect missing
- **Searchability**: Use concrete terms likely in docs
- **Value**: High chance of substantial new information
- **Write relatively short and simpler queries that are
  easy to understand and execute, be more specific in
  the extraction instructions.**

## RESEARCH STATUS
**Overall Context:** {{ original_question }}
**Your Focus Area:** {{ query }}
{{ merged_information }}

{% if completed_queries %}
## PARALLEL RESEARCH
{% for completed_query in completed_queries %}
- "{{ completed_query }}"
{% endfor %}
Stay within your research area only.
{% endif %}

{% if round_num > 1 %}
## ROUND {{ round_num }} - REFINE, DON'T REPEAT
Previous round targeted gaps but some may persist.
If so:
- Generate MORE SPECIFIC queries (countries,
  mechanisms, timeframes, stakeholders)
**Refinement pattern**: "policy impact" ->
  "California EV policy impact 2024"
{% endif %}

## DECISION LOGIC
**STOP enrichment** if:
- Topic area is thoroughly covered from key angles
- Additional searches would likely be redundant
- No meaningful gaps remain within your scope

**CONTINUE enrichment** if:
- Clear strategic gaps exist within your research area
- Missing information would meaningfully improve
  topic completeness

## OUTPUT FORMAT (JSON)
**If meaningful gaps exist**: Generate EXACTLY
{{ max_candidate_queries }} enrichment tasks.
**If NO meaningful gaps exist**: Return empty array.

**Final Check**: Only propose enrichments that would add
meaningful, substantial content to your specific
research area.
```

MioFFAn: an Annotation Software for Formula Formalization with LLM Automation Capabilities

Nicolas Sibuet[*1], **Horacio Saggion**[2], **Riccardo Rossi**[1,3]
[1]Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[2]Universitat Pompeu Fabra (UPF), Barcelona, Spain
[3]International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain.
{nicolas.sibuet, riccardo.rossi}@upc.edu, horacio.saggion@upf.edu

Abstract

The automatic translation of mathematical expressions in scientific literature into executable symbolic code—a process we refer to as Formula Formalization—is hindered by a severe scarcity of high-quality, ground-truth datasets specialized for technical scientific domains. In this paper, we present MioFFAn, an open-source, document-centric, and customizable framework designed to facilitate rapid annotation for this task. Building upon the MioGatto architecture, we extend existing features to overcome structural limitations and pivot its scope by introducing specific functionalities for Formula Formalization, such as selection of equations of interest and aided symbolic code specification. By allowing users to configure custom taxonomies and properties for identified symbols, and compatible symbolic operators, we ensure the framework is adaptable to diverse specialized scientific fields. Furthermore, MioFFAn is designed to incorporate partial automation via Large Language Models. By defining a modular set of automated sub-tasks with strict output formats, we enable researchers to iteratively refine automation capabilities and evaluate competing strategies using standard NLP metrics. We specify the current automation methodology and perform a preliminary evaluation that demonstrates to efficacy of this human-in-the-loop approach.

Keywords: Symbolic Code Generation, Scientific Document Analysis, Dataset Curation

1. Introduction

Scientific literature serves as the primary vehicle for communicating complex discoveries across all disciplines. For many fields, these discoveries are expressed through mathematical formulations that are to be implemented computationally. This implementation process requires both precise semantic interpretation of mathematical notation and the technical ability to manipulate these elements within a programming language. While modern Optical Character Recognition (OCR) systems effectively extract presentational mathematical formats—such as LaTeX or MathML—from PDF documents, these lack the semantic and computational depth required for the extracted mathematical formulas to be implemented in code. As noted by Greiner-Petter (2023), converting these representations into unambiguous, computationally complete models would require systems capable of synthesizing relevant contextual information, such as explicit mentions of variables or operators and their case-specific properties. Although Large Language Models (LLMs) offer a promising mechanism for orchestrating this transformation, their progress is hindered by the scarcity of high-quality, case-specific datasets that can serve as benchmark or training corpora.

In this work, we present MioFFAn (Math identifier-oriented Formula Formalization Annotator)[1], an annotation tool designed for the fine-grained labeling of mathematical expressions within scientific documents to establish the ground-truth symbolic code required to model them in a Computer Algebra System (CAS). We hereafter refer to this task as Formula Formalization. We build upon the MioGatto framework (Asakura et al., 2021), pivoting from its original scope to introduce features tailored to our requirements. In particular, the original MioGatto software was designed for description alignment (Alexeeva et al., 2020) and coreference analysis of mathematical entities, focusing on the evolution of local notations throughout a document. Therefore, it addresses individual symbols rather than the way they are combined within formulas. In contrast, MioFFAn encompasses symbol characterization and grounding but extends to the generation of symbolic code that orchestrates these elements.

Our software provides domain-specific flexibility, allowing users to customize annotation parameters according to the unique constraints of different scientific fields, such as physics, economics and biological modeling, among others. The system is document-centric and interactive, enabling users to maintain full context by annotating directly onto the manuscript. Finally, the framework is designed for semi-automation in a modular way: by specifying certain tasks to be done automatically and providing corresponding interfaces and evaluation routines, developers may implement automation strategies in any way they see fit. While we incorpo-

[1]MioFFAn is open-source and available at https://github.com/NicolasSR/MioFFAn

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 206–217
12 May 2026.

rate baseline automation via LLM in this work, the system is designed to facilitate more sophisticated automated workflows in the future.

To demonstrate the applicability of our framework in realistic research and development scenarios, we present a use case centered on Finite Element Methods (FEM) (Zienkiewicz et al., 2024), cornerstone of numerical simulation. We illustrate how MioFFAn can be configured to characterize this domain's variable properties and operators and we use this use case within a proof-of-concept evaluation of current automation capabilities.

The remainder of this paper is organized as follows: Section 2 reviews related work in terms of similar NLP tasks, existing datasets and prior annotation tools. Section 3 thoroughly describes the MioFFAn framework, including a small review of the original MioGatto tool. Then, Section 4 describes the use case to which MioFFAn has been configured during this study. Following, Section 5 details the proof-of-concept evaluation for the current automation framework. Section 6 discusses possible integration of MioFFAn into other pipelines and Section 7 specifies the most important limitations of the software. Finally, Section 8 concludes the document and proposes future work directions.

2. Related work

2.1. Related NLP tasks

We review the tasks that relate the most to Formula Formalization:

Neural machine translation (NMT) for mathematics. The translation of mathematical notation from presentational formats (e.g., LaTeX) to computational formats can be framed as a particular NMT (Kalchbrenner and Blunsom, 2013; Tan et al., 2020) task. Petersen et al. (2023) use this terminology and demonstrate that end-to-end NMT can outperform rule-based systems in LaTeX-to-Mathematica translation; however, they highlight a significant bottleneck for further generalization caused by notation bias, lack of contextual integration and insufficient hand-crafted data. Prior rule-based pipeline LaCaST (Greiner-Petter et al., 2023) leverages external convention databases and entity linking to address ambiguity, though it lacks the flexibility of deep learning architectures.

Tasks on mathematical grounding and tagging. Relating specific mathematical identifiers with their in-document descriptions is attempted via LLM in (Dev et al., 2024) but also before, via rule-based systems (Schubotz et al., 2016; Alexeeva et al., 2020). Mathematical Entity Linking (Kristianto et al., 2016) strives to associate mathematical symbols to knowledge base entries corresponding to the concept they represent. Finally, Part-of-Math (PoM) Tagging (Youssef, 2017) classifies mathematical symbols according to the type of concept that they represent, according to a specific taxonomy. Zou et al. (2025) and Shan and Youssef (2024) approach the task by using LLMs and determine that further work is needed to recognize complex math concepts and that explicit human expert evaluation is needed for proper assessment.

Autoformalization. The Autoformalization task (Weng et al., 2025) refers to the translation of natural language mathematics into verifiable formal languages (e.g., Lean, Isabelle). Zhang et al. (2025) conclude that including well-defined contextual information and syntactic correction capabilities are determinant to achieve good results. While the task is generally focused on formal proofs, recent expansions have applied these techniques to physics lemmas and theorems (Soroco et al., 2025; Kabra et al., 2025).

Word math problem solving. Finally, research into solving word math problems has shown that offloading reasoning to symbolic solvers or CAS significantly improves accuracy (He-Yueya et al., 2023; Chen et al., 2023). These systems generate intermediate symbolic programs—comprising characterized variables and equations—that align closely with our proposed Formula Formalization output, despite differing in input modality and end goals.

2.2. Related Datasets

Ground-truth data related to Formula Formalization is notably scarce. MathMLben (Schubotz et al., 2018) provides ~300 LaTeX expressions in tree representation with Wikidata-linked variables, while MathAlign (Alexeeva et al., 2020) links identifiers to textual descriptions in arXiv snippets. However, these resources often omit complex expressions and lack the domain specificity required for real-world applications. Recent benchmark STEM-PoM (Zou et al., 2025) focuses exclusively on the PoM Tagging task. In contrast, Autoformalization benchmarks (e.g., MiniF2F (Zheng et al., 2022), LeanEuclid (Murphy et al., 2024)) are highly structured but strictly limited to proof assistants, failing to support the general-purpose CAS integration or numerical simulations required for engineering contexts. Schembera et al. (2025) present a graph knowledge base for applied mathematics and algorithms. It incorporates mathematical models and formulations for several scientific fields, indicating the nature of the quantities included in each formulation, but does not orchestrate corresponding symbolic codes or semantic trees. The knowledge base is collaborative, but limited by time-consuming manual input.

2.3. Mathematical Annotation

Existing frameworks for mathematical annotation vary significantly in input modality and granular focus. Early annotation tools compatible with mathematical notation, like KAT (Schmoll and Wiesing, 2016) and AnnoMathTeX (Scharpf et al., 2019) leverage HTML5 and LaTeX, respectively. The latter incorporates recommendation systems from Wikipedia and arXiv. For PDF-based workflows, Alexeeva et al. (2020) provide a specialized identifier annotator. MioGatto (Asakura et al., 2021) and AnnoTize (Panzer and Schaefer, 2023) both utilize MathML in HTML environments; while MioGatto emphasizes rapid annotation through mathematical identifier recognition and allows for contextual coreference resolution via text highlighting, AnnoTize prioritizes extensive user personalization for annotated object properties and marking style. VARAT (Kato and Kano, 2025) initiates a movement toward domain-specific workflows. It introduces industry-specific annotation for chemical manufacturing, although the annotation content is limited to textual definition and grounding on whole paragraphs. Unfortunately, none of these tools implement specific features for annotating the symbolic code representation of whole mathematical expressions.

The MathMLben suite (Schubotz et al., 2018) focuses on graph representations of complete equations and the STEM-PoM Labeler (Zou et al., 2025) focuses on taxonomic symbol classification. However, these frameworks process isolated equations, losing the document-level context essential for disambiguation.

Critically, while MioGatto and the STEM-PoM Labeler have been used in research involving LLM-assisted automation (Zou et al., 2025; Dev et al., 2024), the implementation of the frameworks used for LLM interaction are not publicly available.

3. The MioFFAn Framework

Designed as a document-centric suite utilizing HTML and MathML, the framework facilitates a multi-stage annotation pipeline that incrementally reduces the complexity of Formula Formalization. The main steps in this pipeline are:

1. Formula Selection: The user identifies a target mathematical expression within the manuscript.
2. Identifier Characterization: The user defines mathematical properties and attributes for the specific identifiers within that expression.
3. Contextual Grounding: These specifications are grounded by aligning identifiers with relevant context segments through document-level highlighting.
4. Symbolic Synthesis: The user writes the symbolic code for the formula, leveraging the previously grounded concepts and CAS operators.

MioFFAn utilizes a client-server model (TypeScript/Python) and operates locally via web browser. The choice of HTML+MathML over LaTeX, which is the preferred markup language for writing scientific corpora, is that they are specifically designed to be integrated into dynamic web applications, facilitating selection of elements and transformations of these via well-established web-app development suites. Nonetheless, conversion from LaTeX files to HTML and vice-versa may be added to the software at a later stage. Furthermore, there is increasing interest from scientific paper repositories such as arXiv towards using HTML as a more accessible visualization format (arXiv, n.d.).

A schematic of MioFFAn's architecture can be viewed in Figure 1, while a global view of the software's interface can be observed in Figure 2. The content of the sample shown in all the figures that contain MioFFAn's interface is sourced from (Hashemi et al., 2021).

As the framework builds upon the MioGatto architecture (Asakura et al., 2021), we first provide a preliminary analysis of the base software's characteristics before detailing our specific extensions and architectural shifts.

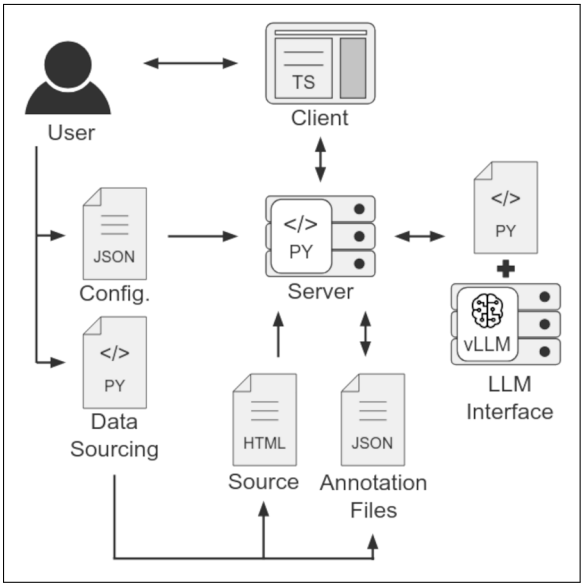


Figure 1: Architecture schematic of MioFFAn

3.1. Description of the Original MioGatto Software

We selected MioGatto as our foundational framework due to its MathML-native architecture, its document-centric approach, its interactivity and its

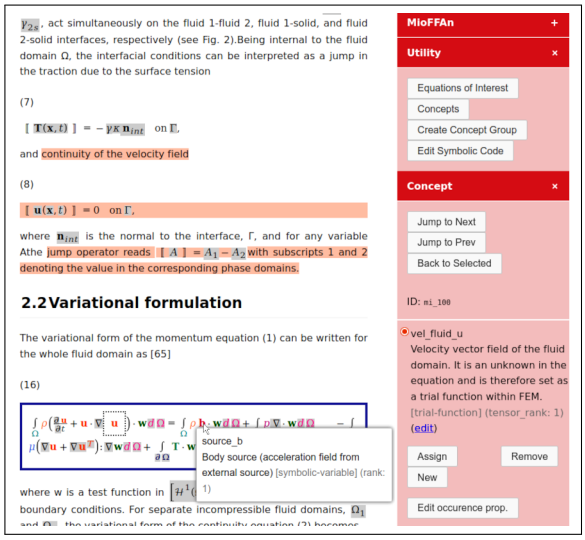


Figure 2: MioFFAn's interface with annotations. The EoI is surrounded by a blue frame. Colored symbols indicate Occurrences. SoGs are highlighted (orange) for the selected Occurrence ($\mathbf{u}$). Information is shown for the Occurrence under the mouse pointer ($\mathbf{b}$). Edition toolbar on the right, in red. The sample content is sourced from (Hashemi et al., 2021), as is in all subsequent figures.

approach to fast identifier selection. Most importantly, its features align with the identifier characterization and contextual grounding steps from our proposed pipeline at the beginning of Section 3.

The system implements three core functionalities:

- Math Identifier (MI) Management: Automated identification of <mi> MathML tags, enabling rapid selection via mouse or keyboard.
- Concept Assignment: Annotation of a given MI is done by assigning a Concept (description, arity, and notation affixes) to it. An MI associated with a Concept is termed an Occurrence[2]; all Occurrences of the same Concept share a distinct color for visual consistency and distinguishability.
- Contextual Grounding: Highlighting of text segments to serve as Sources of Grounding (SoG), linking descriptions or relevant information directly to specific Occurrences.

Despite its strengths, MioGatto presents several bottlenecks for its effective use in realistic annotation dataset creation. Architecturally, the system is constrained by a one-sample-per-session limit, requiring a server restart to switch documents. Datawise, its annotation schema pre-allocates entries for every MI regardless of activity. Functionally, we identify four critical limitations: (1) Annotations are strictly limited to single <mi> tags (generally single characters), preventing the grouping of complex symbols; (2) Concept properties are fixed and mix intrinsic characteristics (arity) with notation-specific ones (affixes); (3) SoG highlighting is restricted to individual nodes, preventing cross-node or in-line math selection; and (4) Concepts are identified by the Unicode characters they contain, making the underlying annotation structure cumbersome to manipulate.

[2]Throughout this paper, capitalized *Concept* and *Occurrence* refer specifically to these architectural entities.

3.2. Enhancement of Original Features

We implemented architectural enhancements that resolve the limitations identified in Section 3.1.

To allow for **hierarchical and compound MI management**: MioFFAn extends the definition of a Math Identifier beyond single <mi> tags. By default, the system now natively supports compound MathML structures, including sub-scripts (<msub>), super-scripts (<msup>), and their combinations (<msubsup>). The user may add more tags as they see fit via JSON configuration. All MIs are uniquely identified by tag IDs.

To capture complex notations like δu_i or $f(x)$ as single entities, we introduce Groups: user-defined MIs encompassing multiple contiguous sibling elements within the MathML tree. These are implemented via a temporary <mstyle class="custom-group"> wrapper in the server-side DOM, preserving the integrity of the original source file. They are permanently and uniquely identified in the annotation files by the first and last <mi> tags within their combined sub-trees in Depth-First Search (DFS) traversal order. The declaration process for Groups is intuitive, needing only the selection by mouse of start and end characters to include. An example of the process can be visualized in Figure 3. This way, the user does not need to deal with the explicit HTML code.

This architecture implicitly enables hierarchical annotation, where a Concept can be assigned to a holistic expression (e.g., a function $f(x)$) while its constituent symbols (e.g., x) retain their own specific Concept assignments. To navigate overlapping tags, users employ assigned keyboard keys to perform a DFS traversal of the MI tree, ensuring precise selection regardless of visual density.

To **decouple the property schema**: we allow both Concepts and Occurrences to have their own independent properties. This way, we differentiate:

- Invariant Semantic Properties (Concept): Defined once for a mathematical entity (e.g., a "velocity vector" (**u**) is always a vector quantity).

- Variant Properties (Occurrence): Specific to a single instance (e.g., whether the vector is currently transposed ($\mathbf{u}^T$)).

This disentanglement allows the same Concept to be reused across different implementation modalities and notations, instead of needing to create slightly different Concepts for each scenario. Users can customize these property sets via JSON configuration, enabling the tool to adapt to the specific requirements of different scientific disciplines. A view of the interface for defining new Concepts in MioFFAn can be seen in Figure 4.

To **improve SoG flexibility**: SoGs now support inter-span highlighting, allowing users to select text across multiple HTML nodes, including in-line mathematical expressions. It also allows for the highlighting of entire displayed formulas. SoGs are now directly associated to Concepts, instead of individual Occurrences.

Additionally, we allow the annotation schema to be dynamically populated as needed and the dependency to Unicode is replaced by unique IDs for Concepts. Finally, users can now navigate between document samples within a single session without requiring a server restart, facilitating large-scale dataset curation.

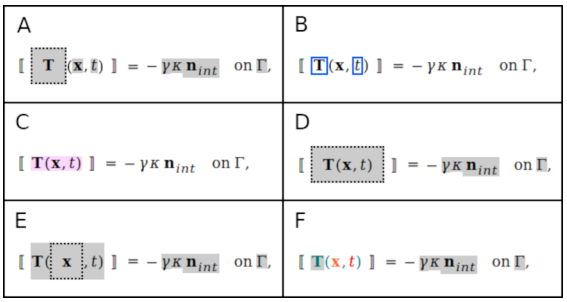


Figure 3: Group designation process and example. Sequentially: (A) Annotation options (shadowed symbols) prior to grouping. (B) Within Group creation tool: first and last Group elements selected by mouse click. (C) Within Group creation tool: complete group visualization. (D) Annotation options after grouping. (E) Demonstration of inner MI selection within Group. (F) Example view of hierarchically assigned concepts.

3.3. Formula Formalization-Specific Features

We introduce three core features that are vital to the Formula Formalization workflow.

First, we introduce **Equations of Interest (EoI)**. Rather than treating all document content equally, users designate specific EoIs, susceptible of formalization. This visually marks the expression and initializes a dedicated entry for that formula in the annotation schema. Each EoI is indexed via the HTML ID of its parent tag. The view of a marked EoI within the document can be seen within Figures 2 and 4.

Second, we introduce **canonical Variable Names**. To ensure that defined Concepts can be uniquely represented within the annotated symbolic code, the framework requires each Concept to be mapped to a Variable Name.

Third, we introduce **integrated symbolic code construction**. The final stage of the pipeline is the synthesis of the symbolic code representation. To minimize manual syntax errors and accelerate the translation process, MioFFAn provides an interactive construction environment with a palette containing the Variable Names for all Concepts defined in previous steps and a list of available operators specific to the target CAS. When a user selects an element from these lists, it is automatically injected into the text box for the code. However, these aids do not stop the user from defining their own intermediate variables and logic manually in order to maintain readability or handle complex local logic. A view of this functionality can be seen in Figure 5.

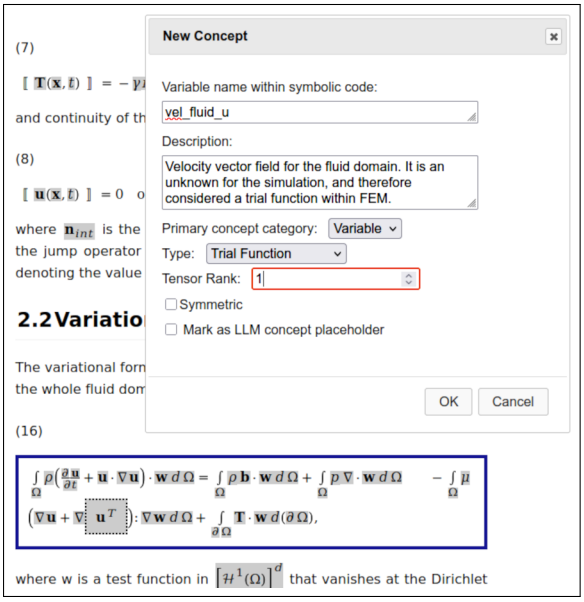


Figure 4: Menu for the registration of a new concept for the selected MI ($\mathbf{u}^T$). Fields according to taxonomy in Section 4. EoI visibly marked by a blue frame.

3.4. Data Sourcing, Management and Preprocessing

To ensure high-fidelity structural representation, we implement a preprocessing routine tailored for the ScienceDirect API[3], as it supports the retrieval of

[3] https://dev.elsevier.com/

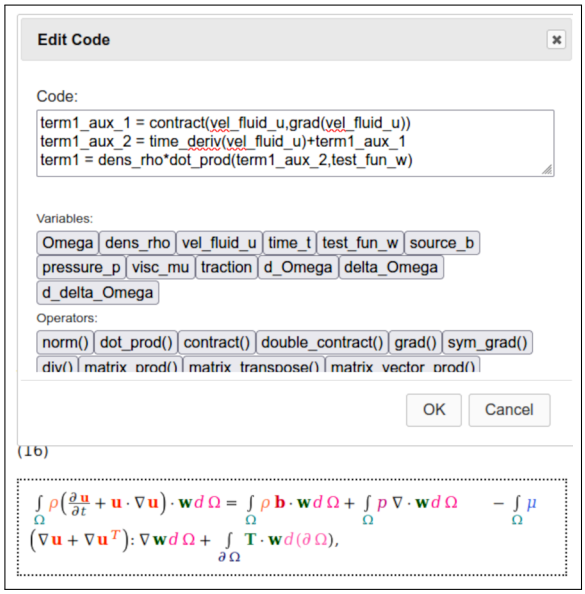


Figure 5: Menu for the specification of the symbolic code for an EoI. Shows palettes for both variables and operators.

research papers in XML format, bypassing the need for non-deterministic OCR and instead performing a direct transformation into MathML-compliant HTML.

During this transformation, we perform a deterministic ID injection phase:

- Mathematical Entities: All <mi> tags, designated compound MI tags and containers are assigned unique persistent identifiers if they are not already present in the source XML.
- Textual Context: To facilitate precise grounding, the text is segmented into <p> nodes and nodes within them, each receiving a unique ID.

This system allows the annotation schema to reference specific tags in the document. Said annotation system is based on two different JSON files: *anno.json*, used for Groups and *mcdict.json*, used for Concepts, EoIs and Occurrences. Both of them are generated during the preprocessing step.

3.5. Partial Automation via LLM

MioFFAn integrates infrastructure for LLM-assisted annotation. Rather than an end-to-end approach, we implement a modular, stage-wise pipeline that allows for human-in-the-loop verification and refinement. The automation logic is implemented in Python and is decoupled from the core application, enabling researchers to swap or evaluate different strategies without modifying the software's primary interface and functionalities.

In terms of interface, we utilize the OpenAI-compatible vLLM API to interact with local Large Language Models. This choice ensures data privacy—a critical requirement in scientific research.

The framework currently supports three automation tasks designed to populate the annotation schema:

- **Symbol Segmentation:** The model identifies which MathML elements or compound structures within an EoI should be grouped as unique Occurrences. For example, in $\delta u F(x)$, the system suggests segments for δu and $F(x)$. These are assigned placeholder Concepts to allow for easy visual verification and correction. The software enables the user to assign placeholder Concepts manually as well, in order to annotate ground-truth samples for this specific task, and in order to correct the automated results before continuing to the next steps. These placeholders do not require any description or Concept property specification.
- **Concept Assignment:** The system suggests Concepts, populated with at least their canonical Variable Names and a textual description. Within this same stage, the model maps these Concepts to existing placeholder Concepts.
- **SoG Identification:** The model identifies text spans or formulas in the HTML tree that serve as the SoG for each Concept. By providing the start and end node IDs, the system automatically generates document-level highlights.

By providing the HTML tree and the content within the annotation files as input for the routines and enforcing a fixed output format at each stage (lists of Groups, Concepts, Occurrences or SoGs to be added), MioFFAn enables methodology-agnostic evaluation. Predicted annotation outputs can be directly compared against gold-standard annotation files, and the annotation files can be processed together with the source HTML file to apply standard NLP metrics. The fact that MioFFAn annotations are based on pointers towards content included in the original HTML file, and not the textual content itself, makes it less susceptible to format-related bias during evaluation. We acknowledge that this limits the scores for systems that generate correct natural language content but are not able to locate it correctly within the original document. However, we believe this relation capability to be an integral necessity for the transparency of these systems towards the human annotator and therefore we consider this effect to be fair. In any case, the user may implement intermediate LLM logs within their automation framework, together with custom evaluation metrics, to directly evaluate natural language responses against the gold standard, bypassing this constraint.

3.5.1. Provisional Automation Methodology

The automation code and LLM prompts used for this paper are available within the examples directory of MioFFAn's GitHub repository. Here we describe the process followed for each task. For the **symbol segmentation**:

1. The LLM is tasked with recreating the MathML sub-structures for candidate symbols.
2. A Python routine using the lxml library performs a search within the source MathML tree to validate these candidates. Proposed structures that lack an exact match in the original source are discarded.
3. If overlapping structures are detected (e.g., x nested within $F(x)$), a second LLM call provides the conflict context and solves it, either removing the inner structure instance or maintaining it as an independent symbol.

For the **concepts assignment**:

1. The LLM proposes Concepts relevant to the EoI. For each one, it defines a unique Variable Name, a textual description, and a Justification for its choice (appended to the description).
2. The LLM performs a mapping from these new Concepts to the available segmented symbols. Concepts that fail to find a specific symbol occurrence are pruned.
3. For every Concept individually, the LLM determines appropriate domain-specific properties.

For the **SoG identification**:

1. For each Concept, the LLM identifies the IDs of specific , <p> or tags that serve as textual evidence for the Concept's meaning and properties.
2. The framework verifies that the returned IDs exist within the document before finalizing the SoG highlights in the annotation file.

4. Use Case

We configure the framework for its use in Finite Element Methods (FEM), a domain characterized by high-dimensional tensors and variational formulations that demand precise characterization. Specifically, we target a custom Sympy-based symbolic compiler[4] designed to translate certain variational forms of partial differential equations into simulation-ready C++ code for the KratosMultiphysics (Dadvand et al., 2010) framework.

[4] https://github.com/NicolasSR/KratosFECompiler

The configuration encompasses, on one side, the **domain taxonomy and properties schema**: We define a tiered schema (Table 1) that establishes primary categories (variables, functions, operators, domains and integration variables) and determines, for each one, invariant Concept properties and instance-specific Occurrence attributes. These can be categorical (multiple choice) or single-valued (boolean, numerical, textual). On the other side, it encompasses the **compiler grammar**: We define a symbolic grammar in terms of a list of operators such as `grad()` (gradient), `div()` (divergence) or `contract()` (tensor contraction), to be utilized from the Sympy-based compiler. These will be available within the MioFFAn symbolic code synthesis palette.

This characterization is the one applied within all figures of this document, and marks the setting for the evaluations in the next section. The full files containing the complete taxonomy and operators list can be found within the examples directory of the MioFFAn repository.

5. Automation Evaluation

We conduct a preliminary evaluation to assess the efficacy of the MioFFAn framework and its provisional LLM routines on a curated set of variational formulations from FEM literature.

5.1. Setup and Dataset

The framework was setup to interface a local vLLM server hosting a Qwen3-4B-Instruct-2507 (Yang et al., 2025) model with 65000 tokens of context length. This particular LLM model was chosen for its reported good performance in scientific content and mathematics, while being able to run on a consumer-grade GPU (RTX 4090 in our setup). A comparison against other models in the same parameters count range is shown in Appendix A.

The evaluation set consists of 6 samples sourced via the ScienceDirect API. The candidate samples collection was done in a greedy way by searching for keywords such as "finite element" or "variational" and keeping only those works published under a Creative Commons license. These were further filtered manually to ensure that they incorporate mathematical formulation aligned with the use case. Furthermore, we did a selective pruning of the original XML files, in order to remove irrelevant content and keep the HTML (in string format) within 20000 tokens long. We define the gold standard data through expert manual annotation via MioFFAn. We provide the data and processing scripts for this study within the examples directory of the MioFFAn repository.

Category	Level	Categorical Properties	Single-Value Properties
Variable	Concept	**Type**: [Trial, Test, Nodal, Symbolic, Numerical]	Tensor Rank, Symmetry
	Occurrence	—	Transpose, Voigt
Function	Concept	**Type**: [Defined, Undefined]	Tensor Rank, Symmetry, Dependencies
	Occurrence	—	Transpose, Voigt
Operator	Concept	**Type**: [Differential, Tensorial, Algebraic]	Linearity
Domain	Concept	**Type**: [Whole, Boundary]	Dimension
Integration var.	Occurrence	—	Domain Name Variable

Table 1: Taxonomy of mathematical concepts and occurrence-specific properties for the FEM use case.

	Symbol Segm.		Concepts Assign.	SoGs Ident.
	Coverage (%)	CoNLL (%)	CoNLL (%)	Avg. SDI (%)
Explicit Names	36.2	26.9	97.0	15.3
Original Characters	57.7	36.8	96.9	14.9

Table 2: Average results for each evaluation metric, comparing the unmodified automation workflow (Original Characters) with a modified workflow that substitutes non-standard Unicode characters by their explicit names (Explicit Names).

5.2. Evaluation Methodology

We evaluate the performance of the automation routines by comparing LLM-generated annotation files against their counterparts for the ground truth. To ensure valid comparisons, we setup the automated framework to perform the Concept Assignment and SoG Identification stages parting from the ground-truth outputs of the Symbol Segmentation stage; this ensures a one-to-one mapping of symbols and prevents error propagation from initial segmentation.

The following metrics are utilized within each automation phase:

- Symbol Segmentation. We assess Coverage, defined as the ratio of <mi> tags correctly identified relative to the gold standard. We define a tag as correctly identified if it is assigned to an Occurrence within both the predicted and ground-truth sets, regardless of the specific placeholder Concept. This does not account for the tags that are being identified in the automated results but not in the ground truth, however we consider these cases to be a minority (most symbols should be identified in the ground truth). To evaluate the accuracy of symbol clustering (distribution of <mi> tags among placeholder Concepts), we utilize the standard coreference metric: CoNLL score (average of MUC, B3, and CEAFe).
- Concept Assignment. The CoNLL score is reapplied here to determine if the segmented symbols are mapped to the correct final Concepts.
- SoG Identification. To measure the quality of the Sources of Grounding (SoG), we treat the identified text and formula IDs as sets and employ the Sørensen–Dice (SDI) index. This index measures the overlap between the LLM-predicted highlights and the expert ground truth for each Occurrence. The final score for a given sample is the average of Sørensen–Dice indexes from all Occurrences.

5.3. Results and Discussion

In this sub-section, we present a proof-of-concept evaluation of different automation methodologies. Specifically, we compare two slightly different variants of our task automation implementations:

- Original Characters (original): No modification is done to the HTML content given to the LLM. So non-standard Unicode characters will stay as they are, e.g "λ".
- Explicit Names (modification): The HTML given to the LLM is processed in order to replace all non-standard Unicode characters by the explicit names given by Python's unicodedata.name() function, e.g. "GREEK SMALL LETTER LAMBDA".

For each approach and metric, the scores for all samples are averaged. These results are shown in Table 2, where we see that the Original Characters variant yields superior performance in the Symbol Segmentation task, with an advantage of over 20% in terms of coverage and 10% in terms of clustering accuracy. Meanwhile, the Explicit Names approach

provides a marginal advantage in both Concept Assignment and SoGs Identification tasks. Such results may guide the developer towards choosing one strategy for the first task and another one for the two latter tasks. Therefore, proving the potential of the MioFFAn tool as an automation development playground.

Apart from this proof-of-concept comparison, we can take the Original Characters results as the current performance of the automation system. In this sense, the system is far from being ideal in the Symbol Segmentation task and, mostly, in the SoGs Identification task. Qualitative analysis for the latter one suggests that the model often identifies correct semantic context but struggles with the precise specification of tag IDs, sometimes highlighting entire paragraphs instead of a small sentence. These results reinforce our decision to prioritize a human-in-the-loop interface, as the automated annotations should serve only as a baseline that can be rapidly refined by a human expert, rather than a final result.

6. MioFFAn in External Pipelines

We briefly discuss how MioFFAn could fit within other tools and pipelines in the field of mathematical formula processing and annotation. In the context of OCR and document mining for scientific content, standard pipelines such as Mathpix[5] and Nougat (Blecher et al., 2023) focus on high-fidelity visual transcription (LaTeX or MathML) but typically lack semantic depth. MioFFAn could serve as a semantic post-processor for these outputs by appending explicit Concepts and symbolic codes to the formulas as textual metadata. Furthermore, within the Mathematical Information Retrieval field, reviewed by Malik et al. (2026), MioFFAn could enhance both text-based and tree-based formula retrieval systems. One could use annotated symbolic code as a precursor for generating semantic trees, or incorporate Concepts and SoGs as additional textual information to improve formula searchability and context-aware similarity comparisons. Finally, MioFFAn may be used as an input tool for knowledge bases such as the ones presented in (Schembera et al., 2025) with little modification efforts, as MioFFAn's data structure aligns closely with their framework: mathematical formulations correspond to EoIs, while quantities and their metadata map to Concepts and Concept properties.

7. Limitations

Arguably, the biggest limitation of the MioFFAn software at current time is the sourcing system's dependency on the ScienceDirect API. Apart from integrating other API's that directly work with XML/HTML formats, conversion from PDF or LaTeX files to MioFFAn-compatible HTML format should be enabled. Annotating web content would also be challenging because of its dynamic nature. This could be solved by developing an output modality that provides the annotations in an explicit way to be appended to the web content or by embedding annotation information directly onto corresponding MathML nodes. Other limitations include not being able to integrate links to external knowledge bases natively, not checking the applicability of CAS operators on given Concepts automatically and not allowing concurrent, collaborative annotation. All of these functionalities can be added progressively in the future.

8. Conclusions and Future Work

We have presented MioFFAn, a document-centric, interactive, and customizable software for the annotation of symbolic code corresponding to mathematical expressions. This software is proposed as a direct solution to the current scarcity of ground-truth data required for symbolic code generation in real-world research and development fields.

By extending the original MioGatto architecture, MioFFAn addresses critical challenges in structural flexibility through compound and hierarchical MI management, a decoupled properties system, and flexible source grounding. Furthermore, it incorporates the necessary features for the Formula Formalization task, specifically Equation of Interest selection and aided symbolic code specification. To demonstrate MioFFAn's adaptability to specialized scientific domains, we have detailed a use case in Finite Element Method by configuring an appropriate concept taxonomy and operator library.

A major feature of this framework is its modular partial automation pipeline via LLMs. By decomposing parts of the annotation process into well-defined, implementation-agnostic sub-tasks, we enable researchers to iteratively test and evaluate distinct automation strategies. Our preliminary evaluation confirms that even provisional LLM routines provide a significant baseline for informed refinement, validating the effectiveness of our human-in-the-loop design.

Moving forward, we aim to enhance MioFFAn through refining current automation strategies and specifying further automation tasks, including a wider range of APIs for source gathering and support for manual document uploads, and incorporating external knowledge bases to enrich the context of identified Concepts.

[5] https://mathpix.com/

9. Acknowledgements

Nicolas Sibuet acknowledges the Secretariat of Universities and Research of the Department of Research and Universities of the Generalitat of Catalonia, as well as the European Social Plus Fund for their financial support through the pre-doctoral scholarship AGAUR-FI (2024 FI-1 00089) Joan Oró.

10. Bibliographical References

Maria Alexeeva, Rebecca Sharp, Marco A. Valenzuela-Escárcega, Jennifer Kadowaki, Adarsh Pyarelal, and Clayton Morrison. 2020. MathAlign: Linking formula identifiers to their contextual natural language descriptions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2204–2212. European Language Resources Association.

arXiv. n.d. Accessible HTML - arXiv info. https://info.arxiv.org/about/accessible_HTML.html. Accessed: 2026-03-23.

Takuto Asakura, Yusuke Miyao, Akiko Aizawa, and Michael Kohlhase. 2021. MioGatto: A math identifier-oriented grounding annotation tool. In *Workshop Papers of the 14th Conference on Intelligent Computer Mathematics (CICM 2021)*, volume 3377 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural Optical Understanding for Academic Documents. ArXiv:2308.13418 [cs].

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. ArXiv:2211.12588 [cs].

Pooyan Dadvand, Riccardo Rossi, and Eugenio Oñate. 2010. An Object-oriented Environment for Developing Finite Element Codes for Multi-disciplinary Applications. *Archives of Computational Methods in Engineering*, 17(3):253–297.

Aamin Dev, Takuto Asakura, and Rune Sætre. 2024. An approach to co-reference resolution and formula grounding for mathematical identifiers using large language models. In *Proceedings of the 2nd Workshop on Mathematical Natural Language Processing @ LREC-COLING 2024*, pages 1–10. ELRA and ICCL.

André Greiner-Petter. 2023. *Making Presentation Math Computable: A Context-Sensitive Approach for Translating LaTeX to Computer Algebra Systems*. Springer Fachmedien, Wiesbaden.

André Greiner-Petter, Moritz Schubotz, Corinna Breitinger, Philipp Scharpf, Akiko Aizawa, and Bela Gipp. 2023. Do the Math: Making Mathematics in Wikipedia Computable. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4384–4395.

Mohammad R. Hashemi, Pavel B. Ryzhakov, and Riccardo Rossi. 2021. Three dimensional modeling of liquid droplet spreading on solid surface: An enriched finite element/level-set approach. *Journal of Computational Physics*, 442:110480.

Joy He-Yueya, Gabriel Poesia, Rose E. Wang, and Noah D. Goodman. 2023. Solving Math Word Problems by Combining Language Models With Symbolic Solvers. ArXiv:2304.09102 [cs].

Aditi Kabra, Jonathan Laurent, Sagar Bharadwaj, Ruben Martins, Stefan Mitsch, and André Platzer. 2025. Can Large Language Models Autoformalize Kinematics? ArXiv:2509.21840 [cs].

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Shota Kato and Manabu Kano. 2025. VARAT: Variable Annotation Tool for Documents on Manufacturing Processes. *Journal of Chemical Engineering of Japan*, 58(1):2454461.

Giovanni Yoko Kristianto, Goran Topić, and Akiko Aizawa. 2016. Entity Linking for Mathematical Expressions in Scientific Documents. In *Digital Libraries: Knowledge, Information, and Data in an Open Access Society*, pages 144–149, Cham. Springer International Publishing.

Ayushi Malik, Pankaj Dadure, and Sahinur Rahman Laskar. 2026. A Review of Mathematical Information Retrieval: Bridging Symbolic Representation and Intelligent Retrieval. *Archives of Computational Methods in Engineering*, 33(1):577–611.

Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. 2024. Autoformalizing Euclidean Geometry. ArXiv:2405.17216 [cs].

NVIDIA, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, et al. 2025. NVIDIA Nemotron 3: Efficient and Open Intelligence. ArXiv:2512.20856 [cs].

Lukas Panzer and Jan Frederik Schaefer. 2023. AnnoTize: A Flexible Annotation Tool for Documents with Mathematical Formulae. In *Proceedings of the 14th MathUI Workshop*, Cambridge, UK.

Felix Petersen, Moritz Schubotz, Andre Greiner-Petter, and Bela Gipp. 2023. Neural machine translation for mathematical formulae. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11534–11550. Association for Computational Linguistics.

Philipp Scharpf, Ian Mackerracher, Moritz Schubotz, Joeran Beel, Corinna Breitinger, and Bela Gipp. 2019. AnnoMathTeX - a formula identifier annotation recommender system for STEM documents. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 532–533, New York, NY, USA. Association for Computing Machinery.

Björn Schembera, Frank Wübbeling, Hendrik Kleikamp, Burkhard Schmidt, Aurela Shehu, Marco Reidelbach, Christine Biedinger, Jochen Fiedler, Thomas Koprucki, Dorothea Iglezakis, and Dominik Göddeke. 2025. Towards a knowledge graph for models and algorithms in applied mathematics. In *Metadata and Semantic Research*, pages 95–109. Springer Nature Switzerland.

Felix Schmoll and Tom Wiesing. 2016. KAT: Enabling the Semantification of STEM Documents. In *Joint Proceedings of the FM4M, MathUI, and ThEdu Workshops, Doctoral Program, and Work in Progress at the Conference on Intelligent Computer Mathematics 2016 (CICM 2016)*, volume 1785 of *CEUR Workshop Proceedings*, pages 66–72, Bialystok, Poland. CEUR-WS.org.

Moritz Schubotz, André Greiner-Petter, Philipp Scharpf, Norman Meuschke, Howard S. Cohl, and Bela Gipp. 2018. Improving the representation and conversion of mathematical formulae by considering their textual context. In *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, JCDL '18, pages 233–242. Association for Computing Machinery.

Moritz Schubotz, Alexey Grigorev, Marcus Leich, Howard S. Cohl, Norman Meuschke, Bela Gipp, Abdou S. Youssef, and Volker Markl. 2016. Semantification of identifiers in mathematics for better math information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, SIGIR '16, pages 135–144. Association for Computing Machinery.

Ruocheng Shan and Abdou Youssef. 2024. Using large language models to automate annotation and part-of-math tagging of math equations. In *Intelligent Computer Mathematics*, pages 3–20. Springer Nature Switzerland.

Mauricio Soroco, Jialin Song, Mengzhou Xia, Kye Emond, Weiran Sun, and Wuyang Chen. 2025. PDE-Controller: LLMs for Autoformalization and Reasoning of PDEs. ArXiv:2502.00963 [cs].

Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. 2020. Neural machine translation: A review of methods, resources, and tools. *AI Open*, 1:5–21.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, et al. 2025. Gemma 3 Technical Report. ArXiv:2503.19786 [cs].

Ke Weng, Lun Du, Sirui Li, Wangyue Lu, Haozhe Sun, Hengyu Liu, and Tiancheng Zhang. 2025. Autoformalization in the Era of Large Language Models: A Survey. ArXiv:2505.23486 [cs].

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. 2025. Qwen3 Technical Report. ArXiv:2505.09388 [cs].

Abdou Youssef. 2017. Part-of-Math Tagging and Applications. In *Intelligent Computer Mathematics*, volume 10383 of *Lecture Notes in Computer Science*, pages 356–374. Springer International Publishing, Cham.

Lan Zhang, Marco Valentino, and Andre Freitas. 2025. Autoformalization in the Wild: Assessing LLMs on Real-World Mathematical Definitions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1720–1738. Association for Computational Linguistics.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2022. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. ArXiv:2109.00110 [cs].

O.C. Zienkiewicz, R.L. Taylor, and S. Govindjee. 2024. *The Finite Element Method: Its Basis and Fundamentals*, 8th edition. Butterworth-Heinemann.

Jiaru Zou, Qing Wang, Pratyush Thakur, and Nickvash Kani. 2025. STEM-POM: Evaluating Language Models Math-Symbol Reasoning in Document Parsing. ArXiv:2411.00387 [cs].

A. LLM models comparison

	Symbol Segm.	
	Coverage (%)	CoNLL (%)
Nemotron	33.9	26.4
Gemma	28.2	18.8
Qwen	57.7	36.8

Table 3: Evaluation scores for the Symbol Segmentation task, comparing three different LLM models. For each model and metric, the results from all samples have been averaged to produce a unique score.

	Concepts Assign.	SoGs Ident.
	CoNLL (%)	Avg. SDI (%)
Nemotron	98.0	15.1
Gemma	95.6	18.3
Qwen	96.9	14.9

Table 4: Evaluation scores for the Concept Assignment and the SoG Identification tasks, comparing three different LLM models. For each model and metric, the results from all samples have been averaged to produce a unique score.

Three LLM models are chosen for comparison in the current implementation of the automation framework: NVIDIA's NVIDIA-Nemotron-3-Nano-4B-BF16 (NVIDIA et al., 2025), Google's gemma-3-4b-it (Team et al., 2025) and Alibaba's Qwen3-4B-Instruct-2507 (Yang et al., 2025).

The evaluation data is the same as described in Section 5.1 and the evaluation methodology is the one described in Section 5.2.

We run each model on all samples and compute the average scores for all of them. These results are listed in Table 3 for the Symbol Segmentation task and in Table 4 for the Concepts Assignment and SoG Identification tasks. The performance in the two latter tasks is similar for all models, with less than 2% of standard deviation. However, the variability of results for the Symbols Segmentation task is significant: the Qwen3-4B-Instruct-2507 model achieves an advantage of over 20% compared to the other models in terms of coverage, and of over 10% in terms of clustering accuracy. Given this significant difference, we choose Qwen3-4B-Instruct-2507 as the model to use in our proof-of-concept study in Section 5.3.

Normalizing section names and structure of scientific articles

Nicolau Duran-Silva[1,2], **César Parra-Rojas**[1],
Julian Moreno-Schneider[3], **Georg Rehm**[3]
[1]SIRIS Lab, Research Division of SIRIS Academic, Barcelona, Spain
[2]LaSTUS Lab, TALN Group, Universitat Pompeu Fabra, Barcelona, Spain
[3] Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Berlin, Germany
{nicolau.duransilva, cesar.parra}@sirisacademic.com,
{julian.moreno_schneider, georg.rehm}@dfki.de

Abstract

The growing amount of scientific literature has increased the need for automatic methods that can retrieve, process, and exploit scholarly content. In this work, we explore section name normalization and hierarchy prediction for scientific articles using a two-level taxonomy. We compare independent, sequential classification models, and generative large language models on the SASC dataset. Results show that classification approaches, particularly sequential models that employ document-level context, consistently outperform generative methods. Incorporating section content is essential for fine-grained classification, while generative models remain limited in zero-shot settings. Our experiments highlight the importance of structure-aware modelling for large-scale scholarly document processing, and the importance of section normalization for the development of advanced research mapping and research assessment tools.

Keywords: Section classification, LLM, Hierarchical text classification, Scholarly document processing

1. Introduction

The rapid growth of scientific literature (Nane et al., 2023) has intensified the need for automated systems that can effectively process, interpret, and reason over full-text research articles (Ghosal et al., 2025; Frommholz et al., 2025). Modern NLP applications increasingly operate on entire scientific papers rather than isolated sentences or abstracts, requiring representations that preserve discourse structure and semantic context (Wadden et al., 2022; Skarlinski et al., 2024; Duan et al., 2025; Keerthana and Gupta, 2025).

Scientific articles are inherently organized into sections that reflect the logical flow of the research process, such as background, methodology, experimental evidence, and interpretation (Sollaci and Pereira, 2004). Accurately identifying the semantic role of these sections is essential for a wide range of downstream tasks, including section-aware retrieval, scientific question answering, summarization, and knowledge graph construction (Accuosto and Saggion, 2019). However, in practice, many scientific parsing tools produce incomplete or inconsistent structural information. This issue is particularly pronounced for documents with complex layouts, such as two-column formats, dense figures, or non-standard templates, where document parsing tools may fail to correctly recover section boundaries and hierarchy. Approaches that infer section roles directly from textual content, as explored in this work, can help reconstruct document structure in such scenarios and provide a complementation to article parsing. Furthermore, LLMs ingest textual data, which can prefer the ingestion of documents as flat or markdown text (Team, 2024). This is particularly relevant with the emergence of retrieval-augmented generation (RAG) systems and research agents that rely on structured textual inputs, when section boundaries and their hierarchical relationships are missing or ambiguous, these systems risk losing critical contextual signals (Keerthana and Gupta, 2025). The problem is further exacerbated in non-normative papers, where section titles deviate from standard conventions, making cross-document alignment more complex.

Section-level structure plays an important role in the development of research assessment and scientometrics, supporting analysis of citations in context, distinguishing whether a reference supports background, methods, or results. Such applications depend on reliable section normalization and hierarchy reconstruction across heterogeneous scientific documents (Vergoulis et al., 2022).

In this work, we explore the possibility of recovering the structure of scientific articles from text at the section level. Beyond improving document parsing, section normalization enables a range of downstream applications that rely on structured scientific content (e.g. RAG, scientific question answering, or scientometric indicator calculation). We address section normalization as a hierarchical classification task under a two-level taxonomy and compare independent, sequential, and generative modeling paradigms. Our results highlight the importance of document-level context and show that

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 218–224
12 May 2026.

structure-aware discriminative models outperform generative approaches in zero-shot settings.

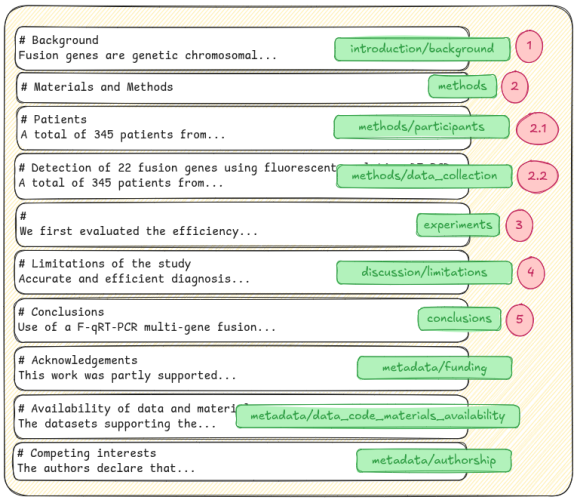


Figure 1: Example of a publication structure annotated with section label and hierarchical level from the SASC Dataset[1].

2. Tasks and Data

We study section-level structure in scientific articles through hierarchical section name normalisation, which aims to map raw section titles (e.g., "Experimental Setup") to a predefined set of canonical section labels. The task is formulated as multi-class classification at two levels of granularity:

- Level-1: coarse roles (e.g., Methods, Experiments);
- Level-2: fine-grained roles (e.g., Data Preprocessing, Evaluation).

Section hierarchy modelling captures the hierarchical organization of sections within a document. However, although the dataset includes explicit section numbering, we restrict our study to a two-level hierarchy (main sections and subsections) based on the 2-level section taxonomy. All deeper levels are collapsed into their immediate parent. Hierarchy is therefore evaluated through level prediction rather than explicit section number prediction.

2.1. Dataset

Experiments are conducted on the Scientific Article Section Classification (SASC) dataset[1] , a large-scale collection of scientific articles segmented by section. The dataset contains 4,896 full-text papers (approximately 117k sections) from different domain splits, including Energy, Cancer, Neuroscience, Transport, and a general science sample.

[1]Available at https://github.com/sirisacademic/sasc

Each section contains raw section title, raw section content, raw explicit section number (when available), semantic labels (level 1 and 2), and hierarchical number. The section taxonomy covers the following Level-1 section categories and their associated Level-2 subcategories are listed below:

- Abstract
- Introduction: background, problem statement, objectives, contributions, outline
- Related Work: literature review, theoretical framework, gaps identified
- Method: study design, participants, data collection, procedures, statistical analysis, ethical considerations, materials and instruments, data preprocessing, evaluation, case study
- Experiment: descriptive statistics, main findings, secondary findings, tables and figures, statistical tests, evaluation
- Discussion: interpretation, comparison with literature, implications, limitations, strengths, future work
- Conclusion: summary, key findings, contributions, final remarks, future directions and recommendations
- Appendix / Miscellaneous: supplementary data, additional analyses, technical details, code and materials, ethics and consent, compliance statement
- Metadata: acknowledgments, authorship, funding, license, data/code availability, publisher note
- References
- Other

3. Methods

We explore section name normalization under different paradigms that differ in how they encode section content and document context.

3.1. Independent classification

We formulate section name normalization as a multi-class classification task, where each section is processed independently. The input to the model consists of the section header concatenated with a truncated snippet of section content separated by a special token:

```
[CLS] sect_header [SEP] sect_content[:50]
```

To control input length and reduce noise, we use only the first 50 tokens of the section content. The resulting sequence is encoded using a language model pretrained on scholarly documents (SciBERT) (Beltagy et al., 2019), and the representation of the `[CLS]` token is passed to a linear classification head. We evaluate three variants: (i) header-only input, (ii) header plus content input, (iii) content

only (filtering only sections with content available). While this approach captures local lexical and semantic elements, it does not exploit dependencies between sections within the same document. The number of the section, for those available, could be a relevant input signal. However, we suspect this could introduce a bias due to the lack of variability and fixed correspondence for some predominant journal structures.

3.2. Sequential classification

To incorporate document-level structure explicitly, we extend the independent classifier into a sequential labelling framework. In this setting, all sections of a document are labelled jointly. Each section is first encoded independently using the same input format as above (header + truncated content) with SciBERT (Beltagy et al., 2019). The resulting section embeddings are then passed to a Transformer encoder (Vaswani et al., 2017) operating over the ordered sequence of sections in a document. This second-stage encoder enables each section representation to attend to other sections, capturing ordering and global patterns. A shared classification head is applied to each contextualized section embedding to predict the normalized label. This architecture mirrors token-level sequence labelling, with sections playing the role of tokens and documents acting as sequences.

3.3. Generative section classification

We additionally explore a generative formulation of section name prediction. In contrast to discriminative classifiers, generative models predict section labels as free-form text conditioned on natural language prompts, which are subsequently normalized to the predefined taxonomy. We explore here two generative settings: *document-level* and *section-level* settings. In the *document-level* setting, all sections of a scientific article are classified in a single model run. The ordered list of section headers, optionally augmented with short content snippets, is provided as input, and the model is prompted to output a sequence of normalized labels in the same order. This formulation enables the model to exploit global document context and implicit structural regularities. In the *section-level* setting, sections are classified sequentially, one at a time, using a local context window. For each section, the model receives the current section header and truncated content, together with the header of the previous and next sections, and—when available—the predicted label of the previous section. This setting approximates incremental document processing to prevent hallucinations in small models while retaining limited contextual information.

In both cases, prompts explicitly enumerate the allowed label taxonomy and enforce output constraints to reduce generation variability. Model outputs are post-processed through a normalization procedure that maps raw generations to valid joint labels (Level-1 and Level-2), with fallback rules applied in case of malformed or out-of-vocabulary predictions. We evaluate two instruction-tuned LLaMA models (Grattafiori et al., 2024), 1B and 8B, under identical prompts, reported in Appendix A, to assess the robustness of generative approaches relative to discriminative baselines. We focus on a zero-shot setting to provide a controlled comparison with discriminative models. However, exploring fine-tuning strategies for generative models is left as future work.

4. Experiments

4.1. Evaluation & Setup

Experiments are conducted on the SASC dataset using document-level train, validation, and test splits. We evaluate section name normalization at two levels of granularity. We additionally report joint correctness (ALL), which requires both Level-1 and Level-2 predictions to be correct. For classification-based models, we report Accuracy and Macro-F1. For generative models, we compute the same section-level metrics after parsing the generated outputs, enabling direct comparison across paradigms. All models are evaluated using identical test sets and label taxonomies.

4.2. Results

Table 1 summarizes the main results. Discriminative models outperform generative approaches across all settings, with sequential models consistently improving over independent classifiers, indicating the benefit of document-level context. Incorporating section content yields substantial gains, particularly for both levels, with content being potentially more relevant than header only. Generative models show lower overall performance under the zero-shot setting, particularly on fine-grained labels, although the LLaMA 8B model performs substantially better than the 1B variant. Especially in the section-level setting, the larger model performs better than the smaller one; however, in the *document-level* setting, the smaller model seems to work better. Unfortunately, these lower perfromance of generative experiments appears to be related to limitations of their model size, such as output processing, taxonomy retention, and hallucinations. Appendix B reports detailed performance by category.

We also analyse performance across scientific domains, reported in Table 2. This indicates that

Model	Paradigm	Input	Accuracy			Macro-F1		
			L1	L2	ALL	L1	L2	ALL
SciBERT	Independent	Header only	0.650	0.491	0.481	0.524	0.339	0.297
SciBERT	Independent	Header + Content	0.777	0.594	0.579	0.672	0.436	0.392
SciBERT	Independent	Content*	0.744	0.521	0.515	0.621	0.338	0.300
SciBERT	Sequential	Header	0.762	0.559	0.547	0.621	0.428	0.396
SciBERT	Sequential	Header + Content	0.797	0.602	0.593	0.683	0.469	0.438
LLaMA-1B	Generative	Header + Content (full)	0.178	0.023	0.089	0.118	0.031	0.036
LLaMA-1B	Generative	Header + Content (section)	0.073	0.001	0.072	0.014	0.001	0.003
LLaMA-8B	Generative	Header + Content (full)	0.460	0.138	0.132	0.311	0.109	0.101
LLaMA-8B	Generative	Header + Content (section)	0.538	0.291	0.251	0.476	0.262	0.226

Table 1: Section Name Normalisation results on the SASC test set. We report Accuracy and Macro-F1 for Level-1 (L1), Level-2 (L2), and joint predictions (ALL). Sequential models exploit document-level context, while generative models perform list-to-list normalization. From *Content-only* we remove all those sections without content, assuming that results are not fully comparable because some challenging cases are removed.

Domain	Accuracy			Macro-F1		
	L1	L2	ALL	L1	L2	ALL
cancer	0.918	0.739	0.715	0.509	0.708	0.485
energy	0.862	0.681	0.680	0.532	0.668	0.499
neuroscience	0.862	0.681	0.680	0.532	0.668	0.499
transport	0.668	0.615	0.450	0.414	0.426	0.390
general	0.693	0.648	0.515	0.443	0.501	0.409

Table 2: Domain-wise performance of the sequential SciBERT model with header and content input.

domains with more standardized article structures and section structure and naming conventions (e.g., cancer and neuroscience) achieve higher accuracy and F1 scores, while domains with greater variability in paper structure (e.g., general domain and transport) are more challenging.

5. Discussion

This work highlights that predicting the semantic structure of scientific articles remains a challenging and unresolved task, even within relatively close scientific domains. While the datasets considered related disciplines, substantial variability in section naming and organization persists, particularly at finer levels of granularity. As a result, current models struggle to generalize consistently across domains with less standardized writing conventions.

Our results show that Level-1 classification is considerably easier than Level-2, reflecting the higher level of abstraction and semantic overlap among fine-grained sections. Models relying only on section headers as input are prone to lexical bias, often learning canonical titles without sufficient contextual grounding. Incorporating section content substantially improves performance, confirming that semantic information beyond headings is critical for reliable classification, as well as the use of signals from document structure.

Generative models exhibit a clear performance gap between coarse- and fine-grained predictions, suggesting limitations in handling with a large number of labels in a zero/few-shot manner. While much larger models could mitigate this issue, their computational cost for processing millions of articles raises practical concerns. This could indicate that lightweight fine-tuning of smaller generative models could bring considerable gains, which will be included as future work.

Overall, this study provides a comparison of architectural paradigms for two-level section classification. Future work includes exploring transfer learning across disciplines, improving efficiency for large-scale processing, and balancing performance gains against computational and environmental costs when deploying larger language models.

5.1. Error analysis

As part of the error analysis, we report in Figure 2 the level-1 confusion matrix for the independent classification model using section headers and content. The results show that most errors arise from confusions with the “Other” and “Abstract” categories, and that sections are frequently misclassified into adjacent categories in the taxonomy, such as “Related Work” vs. “Introduction” or “Method”, and “Discussion” vs. “Experiment”. This pattern suggests that improving the separation between neighbouring categories in the taxonomy during training could represent a promising direction for improvement.

While sequential models may be sensitive to error propagation from earlier sections, our results indicate that the benefits of incorporating document-level context outweigh this limitation. A more detailed analysis of error propagation is left for future

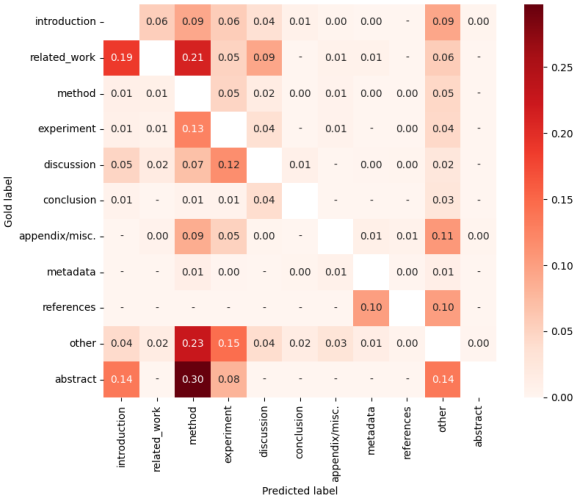


Figure 2: Confusion matrix at Level 1 for the independent classifier based on header + content.

work.

6. Conclusions

In this work, we address section name normalization and hierarchy prediction for scientific articles, comparing independent, sequential, and generative modelling paradigms under a two-level taxonomy. Our experiments show that discriminative approaches based on SciBERT outperform generative models, with sequential document-level modelling providing clear benefits by exploiting structural information across sections. Incorporating section content improves performance, while header-only representations suffer from lexical ambiguity and overlap between adjacent categories. Generative models, despite their flexibility, remain limited in zero-shot settings, particularly for detailed labels, and are sensitive to hallucinations and output normalization. However, differences across scientific domains remain a challenge. Overall, these findings highlight the importance of structure-aware discriminative models for large-scale scholarly document processing and point to future directions in improving taxonomy separation, domain transfer, and computational efficiency.

7. Acknowledgements

Supported by the Industrial Doctorates Plan of the Department of Research and Universities of the Generalitat de Catalunya, by Departament de Recerca i Universitats de la Generalitat de Catalunya (grant reference 2022/DI /00017). This work was co-funded by the EU HORIZON project SciLake (Grant Agreement 101058573) and the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[2] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.). The consortium is funded by the Federal Republic of Germany and its states through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the funders. Neither the European Union nor the granting authority can be held responsible for them.

We thank the anonymous reviewers for their constructive feedback and suggestions, which improved the clarity and quality of this work.

8. Bibliographical References

Pablo Accuosto and Horacio Saggion. 2019. Discourse-driven argument mining in scientific abstracts. In *International Conference on Applications of Natural Language to Information Systems*, pages 182–194. Springer.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Decheng Duan, Jitong Peng, Yingyi Zhang, and Chengzhi Zhang. 2025. SciNLP: A domain-specific benchmark for full-text scientific entity and relation extraction in NLP. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14473–14486, Suzhou, China. Association for Computational Linguistics.

Ingo Frommholz, Philipp Mayr, Guillaume Cabanac, Suzan Verberne, and Christin Katharina Kreutz. 2025. The first workshop on scholarly information access (scolia). In *European Conference on Information Retrieval*, pages 326–331. Springer.

Tirthankar Ghosal, Philipp Mayr, Anita De Waard, Aakanksha Naik, Amanpreet Singh, Dayne Freitag, Georg Rehm, Sonja Schimmler, and Dan Li. 2025. Overview of the fifth workshop on scholarly document processing. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 1–6.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur,

[2] https://www.nfdi4datascience.de

Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Garapati Keerthana and Manik Gupta. 2025. Clirag: A retrieval-augmented framework for clinically structured and context aware text generation with llms. *arXiv preprint arXiv:2507.06715*.

Gabriela F Nane, Nicolas Robinson-Garcia, François van Schalkwyk, and Daniel Torres-Salinas. 2023. Covid-19 and the scientific publishing system: growth, open access and scientific fields. *Scientometrics*, 128(1):345–362.

Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*.

Luciana B Sollaci and Mauricio G Pereira. 2004. The introduction, methods, results, and discussion (imrad) structure: a fifty-year survey. *Journal of the medical library association*, 92(3):364.

Deep Search Team. 2024. Docling technical report. Technical report.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thanasis Vergoulis, Serafeim Chatzopoulos, Kleanthis Vichos, Ilias Kanellos, Andrea Mannocci, Natalia Manola, and Paolo Manghi. 2022. Bip! scholar: a service to facilitate fair researcher assessment. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, pages 1–5.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76, Seattle, United States. Association for Computational Linguistics.

Appendix A. Prompt Templates for Generative Section Classification

We employ structured prompt templates to guide large language models in performing section name normalization under explicit output constraints. Prompts enumerate the allowed section taxonomy and enforce deterministic, schema-conform outputs.

Document-Level Section Classification Prompt

```
You are an expert scientific document
     section normalizer.

TASK:
For each section below, assign exactly ONE
     label from the ALLOWED LABELS list.

ALLOWED LABELS:
{L1 | L2 taxonomy listing}

RULES:
- Use "L1|L2" when a specific subtype
     applies.
- Use only "L1" when no subtype is
     appropriate.
- Output exactly N labels, one per section,
     in order.
- Output ONLY a JSON array of strings. No
     explanations.

DOCUMENT:
[1] HEADER: {SECTION_HEADER_1} | CONTENT: {
     CONTENT_SNIPPET_1}
[2] HEADER: {SECTION_HEADER_2} | CONTENT: {
     CONTENT_SNIPPET_2}
...
[N] HEADER: {SECTION_HEADER_N} | CONTENT: {
     CONTENT_SNIPPET_N}

OUTPUT:
["label_1", "label_2", ..., "label_N"]
```

Section-Level Prompt with Local Context

```
You are an expert scientific document
     section normalizer.

TASK:
Classify the CURRENT section into exactly
     ONE label from the ALLOWED LABELS.

ALLOWED LABELS:
{L1 | L2 taxonomy listing}

SURROUNDING CONTEXT:
PREVIOUS SECTION: {PREV_HEADER} (label: {
     PREV_LABEL})
NEXT SECTION: {NEXT_HEADER}

CURRENT SECTION:
HEADER: {CURRENT_HEADER}
CONTENT: {CONTENT_SNIPPET}

OUTPUT:
Return ONLY a single label string (e.g., "
     method|data_collection",
"introduction|background", or "references").
```

9. Appendix B. Performance by category

Table 3 reports per-category performance, showing higher accuracy for frequent and structurally well-defined sections (e.g., Method, Conclusion, Metadata) and lower performance for heterogeneous

categories.

Section	P	R	F1
Introduction	0.80	0.65	0.72
– Background	0.97	0.68	0.80
– Contributions	1.00	0.05	0.09
– Objectives	0.49	0.30	0.38
– Outline	0.00	0.00	0.00
– Problem Statement	0.37	0.23	0.29
Related Work	0.31	0.39	0.34
– Gaps Identified	0.00	0.00	0.00
– Literature Review	0.80	0.38	0.51
– Theoretical Framework	0.50	0.14	0.21
Method	0.80	0.86	0.83
– Case Study	0.91	0.68	0.78
– Data Collection	0.59	0.55	0.57
– Data Preprocessing	0.59	0.49	0.53
– Ethical Considerations	0.95	0.50	0.66
– Evaluation	0.51	0.39	0.44
– Materials And Instruments	0.57	0.50	0.53
– Participants	0.82	0.78	0.80
– Procedures	0.75	0.63	0.68
– Statistical Analysis	0.90	0.74	0.81
– Study Design	0.64	0.38	0.48
Experiment	0.79	0.77	0.78
– Descriptive Statistics	0.47	0.43	0.45
– Evaluation	0.38	0.37	0.38
– Main Findings	0.68	0.49	0.57
– Secondary Findings	0.47	0.32	0.39
– Statistical Tests	0.64	0.18	0.28
– Tables Figures	0.78	0.45	0.57
Discussion	0.70	0.71	0.71
– Comparison Literature	0.49	0.29	0.36
– Future Work	0.58	0.38	0.46
– Implications	0.58	0.26	0.36
– Interpretation	0.90	0.63	0.74
– Limitations	0.92	0.85	0.89
– Strengths	0.00	0.00	0.00
Conclusion	0.91	0.92	0.91
– Contributions	0.00	0.00	0.00
– Final Remarks	0.89	0.39	0.54
– Future Directions Recommendations	0.67	0.62	0.65
– Key Findings	1.00	0.14	0.25
– Summary	0.85	0.90	0.87
Appendix/Misc.	0.86	0.73	0.79
– Additional Analyses	0.00	0.00	0.00
– Code And Materials	1.00	0.43	0.60
– Compliance Statement	1.00	0.25	0.40
– Ethics And Consent	1.00	0.93	0.96
– Supplementary Data	0.92	0.71	0.80
– Technical Details	0.61	0.23	0.34
Metadata	0.98	0.96	0.97
– Acknowledgments	1.00	0.99	0.99
– Authorship	1.00	0.98	0.99
– Data Code Materials Availability	1.00	0.98	0.99
– Funding	1.00	0.94	0.97
– License	0.33	0.17	0.22
– Publisher Note	0.99	0.85	0.92
References	0.35	0.80	0.48
Other	0.44	0.46	0.45
Abstract	0.68	0.35	0.46

Table 3: Per-category precision (P), recall (R) and F1-score, using the independent classification model with header + content, reporting results at level 1 and 2.

Retrieval-Augmented LLMs and Encoder Models for Multi-Label Climate Disinformation Narrative Classification

Neda Foroutan[1*], Alexandra Tsiakalou[1*], Vera Schmitt[1,2,3,4]
[1]Technische Universität Berlin
[2]German Research Center for Artificial Intelligence (DFKI)
[3]BIFOLD – Berlin Institute for the Foundations of Learning and Data
[4]Centre for European Research in Trusted AI (CERTAIN)
Berlin, Germany
{neda.foroutan, tsiakalou, vera.schmitt}@tu-berlin.de
[*]These authors contributed equally as first authors.

Abstract

The detection of climate disinformation narratives remains challenging due to label imbalance, hierarchical taxonomies, and the multi-label nature of real-world claims. Developing models that can reliably assign fine-grained narrative categories is therefore essential for scalable analysis of climate disinformation. We present our approach to multi-label climate disinformation narrative classification for ClimateCheck@NSLP 2026 Task 2. The task requires assigning one or more narrative categories, defined by the hierarchical CARDS taxonomy, to climate-related claims. We investigate both encoder-based transformers and decoder-only large language models (LLMs), comparing fine-tuning BERT-based models with prompt-based and retrieval-augmented instruction tuning strategies with Qwen3 model. To address data scarcity and label imbalance, we explore targeted augmentation using external CARDS-based resources as well as semantic similarity filtering. Our experiments show that augmentation improves encoder-based models, with ModernBERT achieving competitive performance at low computational cost. However, the strongest results are obtained using retrieval-augmented instruction tuning with Qwen3, which narrows the candidate narrative space prior to prediction. This approach achieves a Macro-F1 score of 59.72% on the official test set, securing second place on the leaderboard. These findings demonstrate the effectiveness of retrieval-guided LLM adaptation for structured multi-label narrative classification while highlighting the continued relevance of efficient encoder-based models.

Keywords: climate change disinformation, narrative classification, multi-label classification, large language models

1. Introduction

Climate change has become a central topic in online and social media discourse. At the same time, digital platforms enable the rapid dissemination of false or misleading narratives about climate science, which can undermine public trust in scientific institutions and influence policy decisions. Automatically detecting and categorizing climate disinformation narratives has therefore emerged as a critical challenge for natural language processing.

The ClimateCheck@NSLP 2026 shared task (Abu Ahmad et al., 2026) addresses this challenge through two sub-tasks: scientific fact-checking and disinformation narrative detection. In this work, we focus on Task 2, which targets the classification of climate-related claims according to the CARDS taxonomy (Coan et al., 2021). Given a claim, systems must identify whether it expresses one or more disinformation narratives or none making the task a multi-label classification problem in which each claim may be associated with zero, one, or multiple narrative categories.

To tackle this problem, we explore both encoder-based transformer models and decoder-only large language models (LLMs). For encoder models, we fine-tune BERT-based architectures and apply targeted data augmentation to mitigate class imbalance and enhance generalization. For decoder-only models, we investigated instruction tuning with prompt-enhanced, hierarchical instruction tuning, and retrieval-augmented instruction tuning, leveraging the structured nature of the CARDS taxonomy to guide prediction.

Our results demonstrate that retrieval-augmented instruction tuning achieves the highest Macro-F1 scores, outperforming both the official ClimateCheck 2026 baseline and our fine-tuned BERT-based systems. Smaller transformer models remain competitive while requiring substantially lower computational cost and environmental footprint. All scripts and resources are publicly available at: https://github.com/XplaiNLP/ClimateCheck-NSLP-2026.

2. Related Works

Coan et al. (2021) presented one of the earliest efforts to systematically categorize climate change contrarian claims. The authors introduced the CARDS dataset by collecting climate-related text from blogs and conservative think-tank websites and developed a comprehensive contrarian climate

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 225–234
12 May 2026.

taxonomy, structured across three hierarchical levels, known as the CARDS taxonomy. Using this framework, they applied supervised machine learning approaches to classify contrarian climate claims, including logistic regression, linear Support Vector Machines (SVM), a pretrained language model, and RoBERTa-large pretrained transformer language model.

Building on this foundation, Rojas et al. (2024) extended the CARDS framework to the social media domain. They proposed the Augmented CARDS framework, a two-stage hierarchical model designed to detect climate disinformation on Twitter, a domain not covered in the original CARDS study. The authors collected contrarian climate-related tweets over a six-month period in 2022 and additionally incorporated the Climate Change Twitter Dataset released by the University of Waterloo. The combined dataset was annotated according to the CARDS taxonomy, with a refinement that separates category 5.3 from category 5.2. Their modeling approach employed DeBERTa in two sequential stages: first, a binary classifier distinguished contrarian claims from non-narrative content; second, identified contrarian claims were assigned fine-grained CARDS taxonomy labels. Their results demonstrated the effectiveness of hierarchical modeling for large-scale disinformation detection in noisy social media environments.

In addition to taxonomy-driven and supervised approaches, prior work has also explored data augmentation strategies for climate disinformation detection. J et al. (2022) investigated the impact of synthetic data generation techniques to improve the classification of climate change denial narratives. The authors examined augmentation methods such as paraphrasing and controlled text generation to mitigate class imbalance in CARDS dataset, demonstrating that augmented data can improve SVM and RoBERTa-large model robustness in low-resource settings. Their findings highlight the importance of data diversity and label distribution when training disinformation detection models.

Beyond CARDS-based studies, prior work has also examined climate-related narratives in other domains. (Rowlands et al., 2024) studied predicting narratives of Climate Obstruction in Social Media Advertising employs a RoBERTa-large classifier to analyze narrative framing in social media advertisements, using a distinct label schema that differs from the CARDS taxonomy. This illustrates domain-specific variations in climate obstruction narratives.

3. Datasets

The ClimateCheck 2026 dataset consists of 939 unique English claims related to climate change. Each claim is annotated with one or more disinformation narrative labels following the CARDS taxonomy (Coan et al., 2021), which defines a structured hierarchy of climate disinformation narratives. The labels span two levels of the taxonomy: super-claims (top-level categories) and sub-claims (fine-grained categories). As the annotation scheme is multi-label, a single claim may be associated with multiple narratives. The dataset is divided into a training set of 763 claims and a test set of 176 claims. It is available at: https://huggingface.co/datasets/rabuahmad/climatecheck.

Due to the relatively small size of the ClimateCheck training set, we further incorporated two additional resources in our experiments to enhance model performance: the original CARDS dataset (Coan et al., 2021) and the Augmented-CARDS dataset (Rojas et al., 2024), containing 9,067 and 10,661 instances, respectively. These supplementary datasets provide additional annotated climate-related disinformation claims, obtained from contrarian blogs, conservative think tank websites, and Twitter, and annotated under the same CARDS taxonomy. However, in contrast to ClimateCheck, these datasets provide single-label annotations, where each claim is assigned exactly one narrative category.

3.1. Augmented_CC26: Data Augmentation with External Datasets

In order to mitigate the label imbalance present in the ClimateCheck dataset, as well as its limited size, we augmented the dataset using external resources aligned with the CARDS taxonomy. Specifically, we used the Augmented CARDS dataset to enrich underrepresented classes. For every class other than 0_0 (no disinformation narrative present), which was the majority class, we randomly sampled up to 300 examples from Augmented CARDS and added them to the original training data. However, not all rare classes were able to be augmented, as many were also underrepresented or entirely missing from the external dataset, as well. We experimented with different augmentation scales, including sampling maximum 100 examples per class, and fully merging Augmented CARDS with the ClimateCheck dataset. However, for ModernBERT, both strategies resulted in lower Macro F1 validation scores, and were therefore not pursued further. The resulting dataset [1] consists of 6030 claims.

[1] https://huggingface.co/datasets/alextsiak/augmented_cc26

3.2. Exorde: Semantic Similarity-Based Augmentation

Additionally, in some experiments we used a subset of a dataset by Exordia Labs (Exorde Labs, 2024) to further augment the ClimateCheck2026 data according to the CARDS taxonomy. This dataset includes over 260 million social media posts, blog posts, and news articles, captured over a one month period from November to December 2024. Each text includes metadata, such as thematic categorization and a set of English keywords representing the core content of the text.

For our experiments, we first filtered the dataset by thematic categories, selecting posts in English labeled 'Environment', 'Health', 'Science', or 'Politics' that contained the keyword 'climate'. We then computed the semantic similarity of these filtered posts to examples in the Augmented CARDS dataset, using the e5-large-v2 pre-trained embedding model (Wang et al., 2024) and cosine similarity. Only the posts above a similarity threshold of 0.86 were retained. This threshold was selected by inspecting the distribution of cosine similarities across a sample of posts, and manually reviewing high-scoring candidates to identify the point at which retrieved posts were consistently similar in content to known climate disinformation claims. Since many of the retrieved texts were only thematically similar to the Augmented CARDS claims, but did not contain any disinformation, we applied the binary disinformation classifier by Rojas et al. (2024) and kept only posts predicted as disinformation. The resulting dataset contained 1690 posts.

3.3. Label Space Alignment

The ClimateCheck dataset follows a multi-label annotation scheme while the CARDS and Augmented-CARDS datasets provide single-label annotations. To enable joint training across datasets, we unified the label space under a common multi-label representation aligned with the CARDS taxonomy.

For encoder-based fine-tuning, all annotations were converted into multi-hot vectors. Single-label instances from CARDS and Augmented-CARDS were treated as one-hot vectors within this multi-label framework. For decoder-based instruction tuning, we maintained a consistent multi-label output format in the prompt design and included examples demonstrating both single-label and multi-label cases.

This unified representation allows seamless integration of the datasets while preserving taxonomy consistency. However, it implicitly assumes that single-label annotations are exhaustive, which may introduce mild noise if additional implicit narratives are present but unannotated.

4. Methodology

To address the multi-label narrative classification, we explored both encoder-only models (e.g., BERT-based models) and decoder-only large language models, including LLaMA and Qwen3. For all experiments involving Qwen3, we used the `unsloth/Qwen3-8B`. In the rest of this paper, we refer to this model simply as Qwen3 for brevity. Our approaches are described in detail below.

4.1. BERT-based models

We experimented with fine-tuning a variety of BERT-based transformer models for multi-label classification. To handle the label imbalance in the the training data, we additionally explored data augmentation strategies using external datasets.

The main transformer-based model used for our experiments was `ModernBERT-large` (Warner et al., 2024). Moreover, we experimented with `RoBERTa-large` (Liu et al., 2019), `DistilBERT-base-uncased` (Sanh et al., 2020), and `Llama-3.1-8B-Instruct`[2] using LoRA.

Prior to training, narrative labels were converted into multi-hot vectors to support multi-label classification. Input texts were tokenized using the respective model tokenizers with a maximum sequence length of 100 tokens, applying padding and truncation. This relatively short maximum length was chosen because of the concise nature of social media posts and the claims in the ClimateCheck dataset.

We finetuned ModernBERT, RoBERTa and DistilBERT only on the official ClimateCheck 2026 training dataset, utilizing binary cross-entropy loss for all. During training, Macro-F1 score on the validation set was monitored, and early stopping was applied with a patience of two epochs.

We also finetuned ModernBERT on the Augmented_CC26 dataset, as well as Llama-3.1-8B-Instruct using LoRA, to evaluate the effect of targeted augmentation on multi-label narrative classification.

4.2. Prompt Enhancement for Qwen3

We built upon the official ClimateCheck 2026 baseline pipeline [3] and enhanced the original prompt through targeted prompt engineering. Additionally, we augmented the enhanced prompt with three additional in-context examples to better guide the model's predictions. The added examples represent the following scenarios: (i) a claim associated

[2] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[3] https://github.com/XplaiNLP/ClimateCheck-2026-Baseline

with multiple narrative IDs (multi-label case), (ii) a claim assigned a single narrative ID, and (iii) a claim labeled as containing no disinformation narrative. This design ensures that the model is exposed to the full range of possible output structures. The final prompt template is shown in Figure 1 (Appendix A.1).

Following the baseline pipeline, we fine-tuned the Qwen3 chat model using supervised instruction tuning with LoRA. Fine-tuning was performed with the final prompt and the data sets described in Section 3.

4.3. Hierarchical Instruction Tuning

We further explored a hierarchical instruction tuning strategy to better leverage the structure of the CARDS taxonomy. In the first stage, we instruction-tuned Qwen3 to predict the top-level narrative category among five coarse-grained classes for the input claim. The prompt followed the same structure as described in the previous approach, with the output format adapted to reflect only the top-level category label. To enable this setup, the original narrative annotations were transformed by extracting their corresponding top-level category. The examples included in the prompt were also updated accordingly, as illustrated in Figure 2 (Appendix A.2).

In the second stage, we instruction-tuned Qwen3 again to predict the fine-grained (sub-level) narrative label. In this setting, the model was provided with (i) the input claim, (ii) the predicted top-level narrative category from the first stage, and (iii) the complete narrative taxonomy list. The prompt examples were modified to reflect this updated input structure.

This two-stage hierarchical approach aims to decompose the prediction task into coarse-to-fine steps, potentially reducing label confusion and improving classification consistency.

4.4. Instruction Tuning with Retrieval Augmentation

To incorporate retrieval-based guidance, we employed the pretrained `cross-encoder/nli-deberta-v3-base` CrossEncoder model to compute semantic similarity scores between each claim and the full list of narratives descriptions defined in the CARDS taxonomy. Based on these scores, we selected the top-10 most similar narratives for each claim, as preliminary experiments showed that this configuration yields better performance than using either top-5 or top-15 retrieved narratives.

The retrieved narratives were then incorporated into a new modified prompt provided to Qwen3. Specifically, the prompt included (i) the input claim, (ii) the top-10 retrieved narrative descriptions, and (iii) the complete narrative taxonomy list. In contrast to the previous prompt-based approaches, no in-context examples were included in this setting, as preliminary experiments showed that adding examples did not improve performance. The prompt template used in this retrieval-augmented setup is illustrated in Figure 3 (Appendix A.3) .

5. Evaluation

5.1. Experimental Setup

All experiments were conducted using two NVIDIA T4 GPUs on the Kaggle platform. Training the Qwen3 model required approximately 2.5 hours per experiment. For training and evaluation of the BERT-based models, we used batch sizes of 16 and finetuned them using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 2e-5 and a weight decay of 0.01. Models were trained for up to 15 epochs, with early stopping based on Macro-F1 on the validation set, and with each training run completing in approximately 20 minutes.

Due to hardware constraints, the `unsloth/Qwen3-8B` model was loaded in 4-bit precision (`load_in_4bit=True`), which reduces memory consumption while maintaining performance. The rest of parameters are kept the same as ClimateCheck 2026 baseline pipeline.

5.2. Evaluation Metrics

The task is evaluated using Macro-Precision, Macro-Recall, and Macro-, micro-, and weighted-F1 scores, where Macro-F1 is considered as the score for ranking in ClimateCheck 2026.

Models' performance are assessed using Macro-Precision, Macro-Recall, and F1 scores computed under macro, micro, and weighted averaging schemes. Among these metrics, Macro-F1 serves as the primary evaluation criterion and is used for system ranking in ClimateCheck 2026.

5.3. Results

Table 1 summarizes the performance of our main approaches and data augmentation strategies. For experiments involving CARDS and Augmented-CARDS, non-narrative and duplicated samples were removed, and only narrative-labeled claims were retained for training.

5.3.1. BERT-Based Models

Among the encoder-only models, ModernBERT achieved the strongest overall performance on the ClimateCheck 2026 (CC26) test set, reaching a Macro-F1 score of 41.36% and outperforming both

Model	Macro-Precision	Macro-Recall	Micro-F1	Weighted-F1	Macro-F1
DistilBERT + CC26	0.3194	0.4353	0.7324	0.7212	0.3505
RoBERTa + CC26	0.3512	0.5096	0.7428	0.7524	0.3832
ModernBERT + CC26	0.3860	0.5155	0.7700	0.7653	0.4136
ModernBERT + Aug_CC26 + Exorde	0.4196	0.6283	0.7341	0.7668	0.4583
Qwen3 & prompt + CC26	0.5146	0.5214	0.8333	0.8067	0.4926
Hierarchical + CC26	0.5574	0.5140	0.8189	0.7890	0.4936
ClimateCheck 2026 Baseline	0.5298	0.5736	0.7977	0.7843	0.5136
ModernBERT + Aug_CC26	0.5537	0.5741	0.8235	0.8171	0.5247
Qwen3 & prompt + Aug_CC26	0.5374	0.5918	0.8264	0.8104	0.5431
Qwen3 & Retrieval + CC26	0.5848	0.5699	0.8474	0.8123	0.5450
Llama-8B + Aug_CC26	0.5738	0.5864	0.8441	0.8369	0.5475
Qwen3 & prompt + CC26 + Aug_CARDS	0.5446	0.6135	0.8296	0.8133	0.5547
Qwen3 & prompt + CC26 + Full-CARDS	0.5474	0.6197	0.8351	0.8192	0.5587
Qwen3 & Retrieval + CC26 + Full-CARDS	**0.6249**	**0.6398**	**0.8011**	**0.8061**	**0.5972**

Table 1: Model performance on the ClimateCheck 2026 test set. CC26 is ClimateCheck 2026 dataset. "Full-CARDS" denotes the combination of the CARDS and Augmented-CARDS datasets. Model variants are defined as follows: Qwen3 & prompt: Prompt Enhancement for Qwen3, Hierarchical: Hierarchical Instruction Tuning, Qwen3 & Retrieval: Instruction Tuning with Retrieval Augmentation

RoBERTa and DistilBERT. As a more recently developed transformer model trained on substantially larger and more diverse corpora, ModernBERT appears better suited for complex multi-label classification tasks.

We additionally experimented with a full merge of the CC26 and Augmented-CARDS datasets; however, this approach resulted in decreased validation performance for the BERT-based models. In contrast, targeted augmentation using the Augmented_CC26 dataset improved performance across all encoder models. This suggests that carefully selected additional data can partially mitigate class imbalance in the original training set. Notably, ModernBERT trained with Augmented_CC26 data surpassed the shared-task baseline, demonstrating that compact transformer models can achieve competitive performance without the computational overhead of large language models.

In contrast to the above augmentation strategy, the semantic similarity-based augmentation (Exorde) did not lead to improved performance, decreasing the Macro-F1 score substantially. Manual inspection of a subset of this dataset revealed that, while the binary disinformation classifier effectively filtered non-disinformation posts, the fine-grained narrative labels assigned through semantic similarity to the Augmented CARDS data were often noisy. We observed that 60.4% of claims were classified as 5_1, 5_2, or 5_1. While these labels, along with the 2_0 category and its sub-narratives, seem to be fairly accurate, other narratives (particularly under 1_0 and 4_0) are more frequently misclassified. Table 2 presents examples of such misclassifications. In addition, some misclassifications seem to be due to posts in the Exorde dataset frequently containing multiple core claims. In contrast, the ClimateCheck dataset contains atomic claims; each claim represents a single statement, even if it can be associated with multiple narratives.

This indicates that semantic similarity alone might be insufficient for reliable narrative labeling, though can be more effective when used as contextual guidance, as explored in our LLM-based approaches.

5.3.2. Instruction Tuning with Qwen3

For our instruction-tuned Qwen3 models, prompt enhancement on CC26 data set achieved the Macro-F1 score of 49.26%, which did not surpass the baseline. However, when we incorporated the enhanced prompt approach with augmented data the performance improved up to 6%. The best performance for the Prompt Enhancement approach (55.87% Macro-F1) was achieved when training on the combined CC26, CARDS, and Augmented-CARDS datasets.

The Hierarchical Instruction Tuning approach on CC26 yielded performance comparable to Prompt Enhancement on the same data. This suggests that decomposing the prediction into two stages, first predicting the top-level category and then the sub-level label, did not provide additional benefits compared to directly predicting fine-grained labels in a single step.

The Retrieval Augmented model achieved the strongest performance among all our approaches. When Qwen3 was instruction-tuned with retrieval augmentation using only the CC26 dataset, it reached a Macro-F1 score of 54.50%. Perfor-

Claim	Assigned Label	Suggested Label
'Man-made climate change' IS a thing - it's just not what 'they' tell you it is. They tell you that natural climate change is man-made, which it isn't, and cannot it be controlled. Weather modification however IS man-made, controllable and has been weaponised.	4_5	2_1
As a Canadian and a mom to three I can't believe that our PM @JustinTrudeau would put feeding my family and keeping a roof over our heads a lower than keeping his climate BS going. He's a parent, I doubt his kids go without anything. Absolutely disgusting comments. He taxes us.	4_4	4_2
There is no climate crisis.	3_0	1_0
CLIMATE TERRORISTS...NO CLIMATE CHANGE YES CLIMATE SCAM..	1_8	5_3
THERE IS ZERO EVIDENCE TO LINK CARBON DIOXIDE WITH THESE CLIMATE EVENTS. IT IS SCARE MONGERING AT ITS WORST.	1_7	2_3

Table 2: Examples from error analysis of the Exorde dataset, showing misclassifications by semantic similarity-based labeling and suggested correct labels.

Team	Macro-Precision	Macro-Recall	Macro-F1
ahilbert	0.7071	0.6310	0.6247
XplaiNLP	0.6249	0.6398	0.5972
ClimateSense	0.6700	0.5678	0.5834

Table 3: Comparison of the top three teams on the ClimateCheck 2026 leaderboard, evaluated using Macro-F1 on the official test set.

mance further improved when training was conducted on the combined CC26, CARDS, and Augmented-CARDS datasets, achieving the highest overall Macro-F1 score of 59.72%. These findings suggest that incorporating a retrieval step effectively narrows the candidate narrative space and provides semantically relevant contextual information prior to prediction, thereby substantially enhancing Qwen3's classification performance.

5.3.3. Leaderboard results

On the official ClimateCheck 2026 leaderboard, our best-performing model, instruction tuning of Qwen3 with retrieval augmentation, achieved a Macro-F1 score of 59.72%, securing second place among all participating teams. The *ahilbert* team obtained the highest Macro-F1 score of 62.47%. Table 3 presents a comparison of Macro-Precision, Macro-Recall, and Macro-F1 scores across the teams. While our team, *XplaiNLP*, achieved the second-highest Macro-F1 score, it obtained the highest Macro-Recall.

6. Conclusion

In this work, we investigated multiple approaches for multi-label climate disinformation narrative classification in the context of ClimateCheck 2026 Task 2. We evaluated both fine-tuning encoder-based transformer models and instruction-tuned decoder-only language models, exploring prompt enhancement, retrieval augmentation, and hierarchical prediction strategies. Our findings demonstrate that targeted data augmentation consistently improves model performance and helps mitigate label imbalance in the original dataset. Among all approaches, instruction tuning of Qwen3 with retrieval augmentation achieved the strongest overall results, outperforming both the shared-task baseline and fine-tuned BERT-based models. At the same time, ModernBERT combined with carefully selected augmented data achieved competitive performance while requiring substantially lower computational resources. These results highlight the effectiveness of retrieval-guided instruction tuning of decoder large language models for complex multi-label narrative classification, while also demonstrating that smaller encoder models remain strong and efficient alternatives.

Limitations

Our work is constrained by computational and memory limitations, which restricted the size of the models and the volume of training data that could be used during instruction tuning. In particular, hardware constraints limited our ability to experiment with larger language models, longer training schedules, and more extensive hyperparameter optimiza-

tion.

Acknowledgments

This research was carried out as part of the *VeraXtract* (reference: 16IS24066) and *news-polygraph* (reference: 03RU2U151C) projects, both supported by funding from the German Federal Ministry for Research, Technology and Aeronautics (BMFTR).

7. References

Raia Abu Ahmad, Max Upravitelev, Aida Usmanova, Veronika Solopova, and Georg Rehm. 2026. ClimateCheck 2026: Scientific Fact-Checking and Disinformation Narrative Classification of Climate-related Claims. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Travis G Coan, Constantine Boussalis, John Cook, P. Nanko, and K.M. O'Connor. 2021. Computer-assisted classification of contrarian claims about climate change. *Scientific Reports*, 11(1):22320.

Exorde Labs. 2024. Multi-source, multi-language social media dataset. https://www.exordelabs.com/. [Data set].

Piskorski J, Nikolaidis N, Stefanovitch N, Kotseva B, Vianini I, Kharazi S, and Linge J. 2022. Exploring data augmentation for classification of climate change denial: Preliminary study. *CEUR Workshop Proceedings*, 3117:97–109.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Cristian Rojas, Frank Algra-Maschio, Mark Andrejevic, Travis Coan, John Cook, and Yuan-Fang Li. 2024. Hierarchical machine learning models can identify stimuli of climate change misinformation on social media. *Communications Earth & Environment*, 5:436.

Harri Rowlands, Gaku Morio, Dylan Tanner, and Christopher Manning. 2024. Predicting narratives of climate obstruction in social media advertising. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5547–5558, Bangkok, Thailand. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.

A. Prompt templates for Qwen3 Instruction Tuning

A.1. Prompt Enhancement for Qwen3

Figure 1 depicts the enhanced prompt template used for instruction tuning, as described in Section 4.2. The prompt retains the general classification structure but incorporates refined instructions and in-context examples to better guide the model's predictions. The examples are designed to cover different output scenarios, including multi-label assignments, single-label predictions, and cases with no disinformation narrative.

A.2. Hierarchical Instruction Tuning

Figure 2 illustrates the prompt template used for top-level narrative classification in the first stage of the hierarchical instruction tuning approach described in Section 4.3. The prompt follows the same general structure as the baseline setup, but the output space is restricted to coarse-grained narrative categories. The in-context examples are adapted accordingly to reflect top-level labels.

A.3. Instruction Tuning with Retrieval Augmentation

Figure 3 presents the full prompt template used in the retrieval-augmented instruction tuning described in Section 4.4. The prompt incorporates the input claim, the top-10 retrieved narratives based on semantic similarity, and the complete narrative taxonomy list.

```
f"""You are an expert system for detecting climate change
   disinformation.
You will be given a single claim. Your task is to classify the claim
   by assigning the most appropriate narrative ID(s) from the
   Narratives_list below.

Instructions:
- Select the narrative ID(s) whose description best matches the claim.
- If the claim clearly aligns with more than one narrative, return all
    applicable narrative IDs separated by a semicolon (;).
- If the claim does NOT contain climate change disinformation or does
   not match any listed narrative, return exactly: 0_0.
- Do NOT explain your reasoning.
- Do NOT output anything other than the narrative ID(s).

Narratives_list:
{Narratives_list_str}

Examples:
Example 1
Claim: "According to recent research, we haven't seen any glacier
   melting or sea level rise since 1995."
Narrative IDs: 1_1;1_6

Example 2
Claim: "According to a recent study, there's no link between CO2
   emissions and temperature changes in our lifetime. #ClimateChange #
   Science"
Narrative IDs: 2_3

Example 3
Claim: "Turns out, species that can adapt easily to different
   environments are often the ones that can survive in a wide variety
   of places."
Narrative IDs: 0_0

Now classify the following claim:
Claim: "{claim}"
Narrative IDs: """
```

Figure 1: Enhanced prompt template for Qwen3 instruction tuning.

```
f"""You are an expert system for detecting climate change
   disinformation.

You will be given a single claim. Your task is to classify the claim
   by assigning the most appropriate narrative ID(s) from the
   Narratives_list below.

Instructions:
- Select the narrative ID(s) whose description best matches the claim.
- If the claim clearly aligns with more than one narrative, return all
    applicable narrative IDs separated by a semicolon (;).
- If the claim does NOT contain climate change disinformation or does
   not match any listed narrative, return exactly: 0.
- Do NOT explain your reasoning.
- Do NOT output anything other than the narrative ID(s).

Narratives_list:
{Narratives_list_str}

Examples:

Example 1
Claim: "Over the past century, the Earth's temperature has risen by
   about 0.1$^\circ$C due to CO2."
Narrative IDs: 1;3

Example 2
Claim: "Interesting fact:  Way back when, CO2 levels were high, but
   the sun wasn't as strong. Makes you wonder about the link between
   solar activity and climate, right?"
Narrative IDs: 2

Example 3
Claim: "Turns out, species that can adapt easily to different
   environments are often the ones that can survive in a wide variety
   of places."
Narrative IDs: 0

Now classify the following claim:
Claim: "{claim}"
Narrative IDs:"""
```

Figure 2: Prompt template for top-level narrative classification in hierarchical instruction tuning.

```
f"""You are an expert system for detecting climate change
    disinformation.
You will be given a single claim and a list of 10 narrative labels
    selected from the full narrative inventory because they are most
    semantically similar to the claim.

Your task:
1. Carefully read the claim.
2. Review the provided Similar_Narratives list.
3. Classify the claim by assigning the most appropriate narrative ID(s
    ) from the provided Similar_Narratives list and Full Narrative
    Inventory.

Instructions:
- Select the narrative ID(s) whose description best matches the claim.
- You MUST choose ONLY from the provided Similar_Narratives.
- If the claim clearly aligns with more than one narrative, return all
     applicable narrative IDs separated by a semicolon (;).
- If the claim does NOT contain climate change disinformation or does
    not match any listed narrative, return exactly: 0_0.
- Do NOT explain your reasoning.
- Do NOT output anything other than the narrative ID(s).

Full Narrative Inventory:
{Narratives_list_str}

Now classify the following claim:
Claim: "{claim}"
Similar_Narratives: {Similar_Narratives_str}
Narrative IDs: """
```

Figure 3: Prompt template for Qwen3 instruction tuning given claim and its top-10 similarity with narratives Taxonomy.

The Linguist's Lie Detector: Linguistic Knowledge in Large Language Models

Lucía Catalán Gris[1], **Kim Gerdes**[1], **John S. Y. Lee**[2]
[1] Université Paris-Saclay, Lisn, CNRS, Orsay, France
[2] City University of Hong Kong, Department of Linguistics and Translation, Hong Kong SAR, China
lucia.catalan-gris@lisn.fr, kim.gerdes@lisn.fr, jsylee@cityu.edu.hk

Abstract

We present a benchmark and evaluation pipeline for assessing how well large language models (LLMs) handle linguistic knowledge. Starting from a curated subcorpus of 11 syntax-focused articles published in *Glossa: A Journal of General Linguistics* (2016–2026), we design a pipeline that (1) segments article text into sentences, (2) extracts atomic, verifiable statements, and (3) classifies them into linguistic categories (language-specific, typological, theoretical, citation, or structural). Each stage is evaluated against human gold annotations produced by three annotators, with inter-annotator agreement measured via Krippendorff's α and Cohen's κ. We compare several LLMs on extraction and classification, using BERTScore-style similarity for extraction and macro F1 for classification. Finally, we generate contradictions of the true linguistic statements and test whether LLMs can distinguish true from false claims. On a challenge set of 705 linguistic statements, we compare eight LLMs, with Gemini 3 Flash achieving the highest F1 score of 0.66, indicating that current models possess limited but non-trivial linguistic knowledge.

Keywords: fact-checking, linguistics, LLM evaluation, statement extraction, Glossa

1. Introduction

Large language models (LLMs) are trained on vast amounts of linguistic data and perform well on many language tasks. However, whether this success reflects genuine linguistic knowledge remains an open question. Linguistics provides a useful test domain for exploring this question: LLMs are *trained on* language, but are they knowledgeable *about* language? Can they distinguish a true claim about the Warlpiri sentence structure from a plausible but false one?

To illustrate the challenge, consider the following example from our corpus, which includes one true statement alongside two LLM-generated contradictions:

(1) a. In Warlpiri, all intransitive verbs combine with an absolutive subject. [true]
b. In Warlpiri, some intransitive verbs combine with an ergative subject.[1] [false; GPT-5.2]
c. In Warlpiri, some intransitive verbs combine with a non-absolutive subject. [false; Gemini 2.5 Flash]

Both false variants avoid explicit verbal negation yet reverse the truth conditions, creating minimal contrasts that are difficult to detect without relevant linguistic knowledge—exactly the kind of challenge a linguistically competent system should be able to resolve.

By evaluating models on such metalinguistic statements, we test their ability to assess claims about language itself rather than simply generate well-formed text. This provides a controlled way to examine whether LLMs truly understand linguistic rules, rather than just mimicking patterns they've seen before.

In this paper, we present a pipeline for checking LLMs' linguistic knowledge. Our contributions are:[2]

1. A curated dataset of 342 statements from open-access syntax articles from *Glossa: A Journal of General Linguistics*, spanning both high- and low-resource languages.

2. A three-stage NLP pipeline—Sentence Segmentation, Statement Extraction, Statement Classification—with detailed prompts and multi-model comparisons.

3. Gold annotations by three human annotators for both extraction and classification, with measured inter-annotator agreement.

4. A linguistic knowledge evaluation in which LLMs determine whether 705 linguistic statements are true or false; one-third are correct, and two-thirds are deliberately falsified.

[1] While technically false for Warlpiri, this statement is typologically plausible. In many split-intransitive (active-stative) languages, agentive intransitive subjects (*unergatives*) receive ergative marking. Furthermore, Warlpiri "middle" verbs take an ergative subject and a dative object; to an LLM, these might be misclassified as intransitives due to the lack of an absolutive/accusative argument.

[2] All data, prompts, and code are publicly available under CC-BY 4.0 at https://github.com/linguistic-fact-checking/nslp26

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 235–246
12 May 2026.

2. Related Work

This work sits at the intersection of several research areas: automated fact-checking, scientific claim verification, information extraction from scholarly articles, and automatic negation generation. In this section, we briefly review these research areas and discuss how our pipeline complements existing approaches.

2.1. General-Domain Fact-Checking

Automated fact-checking has received considerable attention in the NLP community. Thorne and Vlachos (2018) provided an early survey unifying task formulations across domains—focusing on claim inputs (such as triples or text), evidence retrieval (from structured and unstructured sources), and veracity outputs—noting that scientific journal text and encyclopedia articles are among the most common evidence sources. Guo et al. (2022) extended this survey with a three-stage NLP framework (claim detection, evidence retrieval, claim verification) and identified academic papers as a frequently used textual evidence source. More recently, Chen et al. (2024) developed a realistic pipeline for complex claim verification that retrieves raw web evidence and decomposes claims into sub-tasks: question generation, retrieval, answer extraction, and verdict synthesis.

A key dataset in this task is AVeriTeC (Schlichtkrull et al., 2023), which contains 4,568 real-world claims annotated with question/answer pairs, web evidence, and verdicts (Supported, Refuted, Not Enough Evidence, or Conflicting Evidence/Cherrypicking). Putta et al. (2025) introduced ClaimCheck, a fact-checking system evaluated on AVeriTeC that integrates claim-matching with novel claim processing using smaller open-source LLMs, demonstrating 62.6% verdict prediction accuracy.

2.2. Scientific Claim Verification

While general fact-checking typically accepts one or more sentences from a web page (e.g., Wikipedia) as sufficient evidence, scientific fact-checking imposes more rigorous evidentiary standards. Vladika and Matthes (2023) surveyed scientific fact-checking resources and approaches. They highlighted the need for automated tools that can handle the complex language found in research publications and assist scientists in verifying hypotheses and discovering evidence. Pradeep et al. (2021) proposed VerT5erini, a T5-based system for scientific claim verification in the biomedical domain, evaluating on SciFACT and demonstrating generalization to COVID-19 claims using the CORD-19 corpus. Sarrouti et al. (2021) introduced HEALTHVER, a dataset for evidence-based fact-checking of health-related claims against scientific articles, with three verdict types (Supports, Refutes, Neutral).

To our knowledge, **no prior work targets linguistic claims** extracted from peer-reviewed linguistics articles.

2.3. Claim Extraction from Scientific Text

Our Statement Extraction pipeline is motivated by research on identifying and formalizing claims in scientific publications. Early work by Nasar et al. (2018) surveyed computational approaches for automatically extracting structured information from scientific literature, including rule-based methods, conditional random fields (CRFs), and deep learning techniques. Building on this paradigm, Bekoulis et al. (2021) examined the Fact Extraction and VERification (FEVER) framework, including the SciFACT dataset, which validates scientific claims against a corpus of abstracts through abstract retrieval, rationale selection, and label prediction.

Several efforts have targeted the extraction of atomic claims specifically. Blake (2010) introduced the Claim Framework for identifying explicit, implicit, and under-specified claims in biomedical articles, differentiating levels of evidence. Jansen and Kuhn (2016) presented an approach for extracting core claims and representing them as AIDA sentences—atomic, independent, declarative, absolute English statements—providing a normalized format for scientific findings. Bleuze (2024) reported on defining claim types and training models for automated claim detection in NLP research papers, creating an annotated corpus of claim categories. At a larger scale, Dessì et al. (2022) introduced CS-KG, a knowledge graph of 41 million statements automatically extracted from 6.7 million computer science articles.

More recently, LLMs have been applied to structured extraction. Dagdelen et al. (2024) demonstrated that fine-tuned LLMs (GPT-3, Llama-2) can extract complex structured knowledge from scientific text and return it in JSON format, a paradigm closely related to our prompt-based extraction pipeline. Our work differs from these approaches in that we extract *linguistic* claims from full articles (not abstracts), enforce self-containedness and atomicity through detailed prompting, and evaluate against human gold annotations with measured inter-annotator agreement.

A related line of research uses NLP to predict whether scientific claims are replicable (Youyou et al., 2023); our work addresses the complementary question of whether claims are *factually correct* with respect to linguistic data.

2.4. Automatic Negation and Contradiction Generation

A distinctive feature of our evaluation is the use of LLM-generated contradictions to create false linguistic statements.

Bilu et al. (2015) addressed automatic claim negation in the context of argumentation mining, proposing rule-based and statistical methods for generating negations of natural-language claims. While Bilu et al. (2015) demonstrated that rule-based methods involving explicit negation markers (e.g., 'not', 'no') can successfully negate 75% of claims, they identified 'Usability'—the plausibility of a negation in context—as the primary bottleneck. Our approach extends this line of work by using LLMs with a carefully designed prompt that enforces strict logical contradiction while *prohibiting* explicit negation markers ("no", "not"), making the resulting false statements harder to detect through surface cues alone. Early experiments confirmed that this approach produces higher-quality contradictions than rule-based methods, consistent with the general advantage of LLMs on generation tasks.

3. Tasks Definition

3.1. Task 1: Statement Extraction

Given a paragraph, the goal is to extract all the atomic, self-contained, verifiable statements within the text, adhering to these guidelines:

- Each statement must be **atomic**, meaning that coordinated elements are separated whenever each conjunct expresses a distinct, independently meaningful proposition.
- Each statement must be **self-contained**, requiring the resolution of all pronouns and the expansion of abbreviations to ensure that they can be understood without reference to the surrounding context.
- Each statement must **preserve technical precision**. They should retain the technical terminology and explicitly include the relevant language studied when contextually clear. Moreover, linguistic examples should be incorporated directly into the statement text, with example numbers, glosses, and translations removed.
- Pure conditionals should be retained as single statements, whereas causal or inferential chains should yield both the premise and the resulting implication.
- Citation attribution should be preserved, so that if an author endorses a cited claim, both the attributed version and the general version are extracted.

3.2. Task 2: Statement Classification

Given a statement, the task is to assign one of the following labels:

- **Language-Specific statements (L-Spec)**: empirical claims about a specific language that are verifiable on language data.
- **Language-Typological statements (L-Typo)**: cross-linguistic or comparative claims.
- **Language-Theoretical statements (L-Theo)**: theory-internal generalizations that are not directly verifiable on raw data.
- **Citations (C)**: statements attributed to another author or paper.
- **Structural (S)**: structural or methodological statements that do not contain any linguistic claim, such as section headers, definitions, methodology descriptions, meta-discourse.

3.3. Task 3: Linguistic Knowledge Evaluation

The final task analyzes whether LLMs possess genuine linguistic knowledge beyond surface-level pattern matching.

The input consists of a claim previously classified as a linguistic statement (L-Spec, L-Typo, and L-Theo). For each statement, the model is expected to provide:

- A **verdict** (True or False): an assessment of the statement's factual accuracy.
- A **justification**: a concise explanation supporting the verdict.
- A **confidence score**: a numerical estimate of the model's certainty in its verdict. In Section **??** we show that these scores are positively associated with correctness, suggesting that the model is reasonably well calibrated on this task. Therefore, they are useful for selective prediction in a semi-automated workflow: high-confidence judgments can be prioritized for automatic acceptance, whereas low-confidence or uncertain cases can be flagged for human review.

The distinction between L-Spec, L-Typo, and L-Theo, while not strictly necessary for the LLM-based verification methods evaluated in this paper, will facilitate future work in selecting appropriate

linguistic resources for claim verification. For example, L-Spec claims may be verified with UD treebanks for the relevant language, and L-Typo claims can be checked in typological databases such as WALS, Grambank, or APiCS.

4. Dataset Construction

To enable a reliable evaluation of LLM performance, we created a manually curated gold dataset. This section outlines the methodology for producing the gold standard, including corpus collection and sampling, data preparation, and the human annotation process.

4.1. Article Collection and Sampling

We collected 1,139 scientific articles published in *Glossa: A Journal of General Linguistics* between its creation in 2016 and February 2026. Among the most prestigious journals for general linguistics, *Glossa* publishes under a Creative Commons Attribution 4.0 license, making it ideal for open-science research.

From these, we selected the articles that meet three criteria:

- The article has an **XML (JATS) file** available. Out of the 1,139 articles collected, 1,025 fulfilled this constraint.
- The article belongs to the domain of **syntax**, as determined by its title and keywords (693 of the 1,025 articles). Our focus on syntax is motivated by our planned future work on claim verification on real linguistic data, such as syntactic treebanks, to supplement the LLM-based methods presented in this paper.
- The language(s) studied in the article should span the spectrum of high- to low-resource languages, and they should also be represented in **Universal Dependencies** (UD) treebanks, enabling future cross-validation of linguistic claims on treebank data.

This selection yielded **11 articles** written in English, that study both high-resource and low/mid-resource languages.

4.2. Automatic Data Preparation

To prepare for human annotation, we built a small subset of statements using the pipeline presented in Figure 1.

First, we parsed 11 JATS-XML files–the previously sampled articles in the previous section–into sections and paragraphs, resulting in 1,082 unique paragraphs. The linguistic examples cited in the text were included within their enclosing paragraph rather than treated as separate units.

After, we selected a subset of 63 paragraphs, aiming to obtain at least 10 statements per article. Each paragraph was segmented into sentences using an LLM (GPT-5.2). The abstract was treated as a paragraph, and we added the preceding paragraph as context to resolve cross-references. From these 63 paragraphs, we ended up with 108 unique sentences. Each paragraph contained between 6 and 32 sentences, with an average of 12 sentences per paragraph.

For each of the previous extracted sentences, an LLM (Gemini 2.5 Flash) was prompted to do a first approximation to Statement Extraction (Section 3.1) and extract all the atomic, self-contained, verifiable statements. A total of 342 atomic statements were extracted from the 108 unique sentences, averaging 3.17 statements per sentence.

Then, a first approximation of Statement Classification (Section 3.2) was run with Gemini 2.5 Flash. The 342 atomic statements were classified into the categories L-Spec, L-Typo, L-Theo, C, or S. For statements labelled as L-Spec, we tasked the model to identify the language referred to in each statement. For example, the statement "in Warlpiri, all intransitive verbs combine with an absolutive subject" was classified as L-Spec, and the language identified was Warlpiri.

All data preparation stages used the provider default temperature setting (1.0) and the prompts provided in the Appendix A. Adhering to the FAIR Guiding Principles for scientific data management and stewardship, each statement can be explicitly traced back to the original paragraph and article in which it appears.

4.3. Human Annotation Process

For the human annotation process, we developed detailed annotation guidelines, and we employed three annotators. For each step, annotators first completed their annotations independently and then consolidated their results, discussing any disagreements until reaching consensus. We assessed the annotation reliability using Krippendorff's α, Cohen's κ, and the % agreement.

4.3.1. Human Annotation of Statement Extraction

For each LLM-proposed statement, annotators judged whether it was acceptable (*ok*) or problematic (*problem*), following the annotation guidelines requiring that each statement be:

- **Atomic**: expressing a single verifiable proposition.

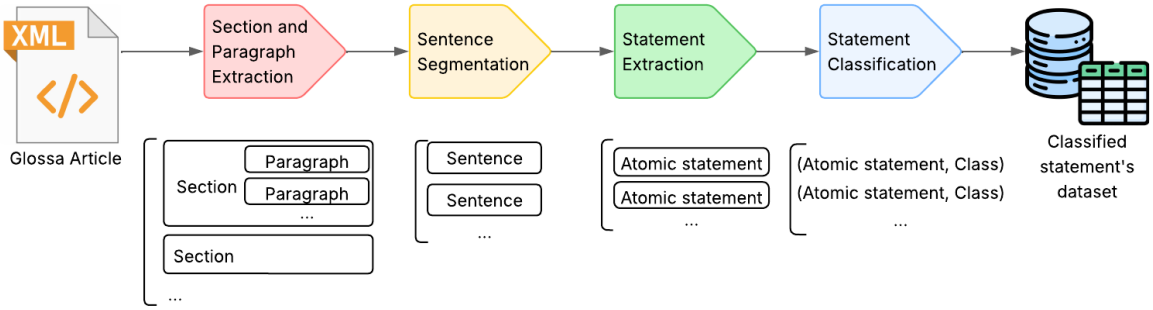


Figure 1: Pipeline used for extracting and classifying atomic statements from XML-encoded articles. The top row shows processing steps, while the bottom row shows the corresponding intermediate representations.

- **Independently verifiable**: understandable without the source paragraph.
- **Inferable from the source text**: not introducing information beyond the paragraph.

Metric	Value
Krippendorff's α (3 annotators)	0.299
Cohen's κ (Ann. 1 & Ann. 2)	0.397
Cohen's κ (Ann. 1 & Ann. 3)	0.208
Cohen's κ (Ann. 2 & Ann. 3)	0.301
% agreement (Ann. 1 & Ann. 2)	81.3%
% agreement (Ann. 1 & Ann. 3)	73.1%
% agreement (Ann. 2 & Ann. 3)	78.4%

Table 1: Inter-annotator agreement for Statement Extraction between the three annotators.

Table 1 reports the inter-annotator agreement. The relatively low values ($\alpha = 0.299$) reflect the inherent difficulty of the task. Although raw percentage agreement is comparatively higher (73.1%–81.3%), this likely overestimates true agreement due to chance effects and the uneven distribution between the *ok* and *problem* categories.

The annotators reported several challenges during the annotation process. First, the scope of an in-text citation can be ambiguous, especially when the citation is preceded or followed by multiple claims. Although annotators were asked to err on the side of caution and select the smaller scope when in doubt, the difference in judgment on scope remained a considerable source of disagreement. Furthermore, there can be significant subjective variation in defining what constitutes an atomic statement, particularly when there is ambiguity in coordination scope. The disagreement between annotators mostly did not involve the content of the claims themselves, but rather alternative interpretations of scope and atomicity. In the vast majority of cases, these interpretations may all arguably serve as gold statements without affecting their veracity value. Based on these observations, the annotation guidelines were subsequently refined.

Following annotation, the three annotators held a discussion meeting to reconcile disagreements and produced **342 human-curated gold statements** extracted from the 63 paragraphs.

4.3.2. Human Annotation of Statement Classification

The same three annotators independently classified the curated statements into the categories defined in Section 3.2. For statements labelled as L-Spec, they were also asked to provide the language referred to in each statement. Agreement was substantially higher than for the extraction task (Table 2).

Metric	Value
Krippendorff's α (3 annotators)	0.750
Cohen's κ (Ann. 2 & Ann. 1)	0.725
Cohen's κ (Ann. 2 & Ann. 3)	0.825
Cohen's κ (Ann. 1 & Ann. 3)	0.700
% agreement (Ann. 2 & Ann. 1)	81.4%
% agreement (Ann. 2 & Ann. 3)	87.6%
% agreement (Ann. 1 & Ann. 3)	79.4%

Table 2: Inter-annotator agreement for Statement Classification between the three annotators.

The most frequently identified languages in the L-Spec statements were: English, North Sámi, Yorùbá, Spanish, Basque, Polish, German, and Russian. Other languages appearing in the annotated dataset (each with 1 or 2 statements) include Mauritian Creole, Greek, Bulgarian, Saudi

Arabic, Brazilian Portuguese, Indonesian, Warlpiri, Samoan, Niuean, and Japanese. Different annotators showed high agreement on these language assignments, with English and North Sámi consistently appearing as the most common subjects of study. The low number of language-specific tags relative to the total number of statements indicates that the dataset primarily consists of general linguistic claims (L-Theo or L-Spec) rather than specific language data points (L-Typo).

A gold classification for all 342 statements was produced based on the individual annotations. If the three humans agreed, their label was kept; disagreements were resolved by manual review. The final dataset contains: 115 L-Spec, 110 L-Theo, 10 L-Typo, 91 C, and 15 S.

5. Experimental Setup

Evaluated Models. We ran the evaluation on eight LLMs spanning four provider families: GPT-5.2, GPT-5 Mini, and GPT-5 Nano (OpenAI); Gemini 3 Flash and Gemini 2.5 Flash (Google); DeepSeek V3.1 (DeepSeek/Fireworks); and Kimi K2.5 and Kimi K2 Instruct (Moonshot/Fireworks). The selection includes both open- and closed-source models with varying capabilities, enabling a more comprehensive evaluation.

Prompts settings. The models were evaluated using zero-shot prompts and tasked with returning only JSON arrays to facilitate post-processing of the results. The complete text of the prompts used is in Appendix A.

Challenge Dataset for Task 3. To evaluate whether LLMs possess genuine linguistic knowledge, we focused on the 235 L-type statements—115 L-Spec, 110 L-Theo, 10 L-Typo—since the S statements do not make a verifiable scientific claim, and the authors do not own the C statements. We constructed a challenge set of true and false statements. The 235 linguistic gold statements served as the true set. For each true statement, we generated two false variants: one using GPT-5.2 and one using Gemini 2.5 Flash. We used a contradiction prompt that requires:

> *Write a statement that is a strict logical contradiction of the claim below: both statements cannot be true at the same time. Keep the wording and structure as close as possible. The change must reverse the truth conditions of the claim. Do not use "no", "not", or explicit negation. Keep about the same length.*

Human annotators reviewed the automatically generated false statements to ensure that they contradicted the original version. The quality was generally very high and only a few statements required revision.

6. Results and Discussion

6.1. Task 1: Statement Extraction

We evaluated how well different LLMs extract statements by comparing their predictions against the 342 gold statements. Given the paraphrastic nature of the task—a model may express the same fact in different words—we use a semantic similarity approach rather than an exact match.

Evaluation metrics. We embedded both gold and predicted statements using the `all-MiniLM-L6-v2` model from the Sentence-Transformers library (Reimers and Gurevych, 2019), chosen for its strong performance on short-sentence similarity and paraphrase detection. Following Hanna and Bojar (2021), we computed the cosine similarity matrix between gold and predicted embeddings. **Recall** is the mean of the maximum similarity for each gold statement (how well does the model cover the gold?). **Precision** is the mean of the maximum similarity for each predicted statement (how relevant are the predictions?). **F1** is the harmonic mean of precision and recall.

Model	Prec.	Rec.	F1
Gemini 3 Flash	0.907	**0.883**	**0.894**
GPT-5.2	0.906	0.871	0.888
GPT-5 Mini	0.916	0.856	0.885
Gemini 2.5 Flash	0.917	0.853	0.884
Kimi K2.5	0.904	0.854	0.878
Kimi K2 Instruct	0.911	0.772	0.836
GPT-5 Nano	0.873	0.743	0.803
DeepSeek V3.1	**0.965**	0.494	0.653

Table 3: Statement Extraction evaluation.

Results. Table 3 presents the performance of all evaluated models on task 1. Most models achieve F1 scores between 0.80 and 0.89. Gemini 3 Flash leads with an F1 of 0.894, closely followed by GPT-5.2 (0.888). Gemini 2.5 Flash and Kimi K2.5 also perform competitively, though slightly below the top tier. GPT-5 Nano, the smallest model tested, still achieves a respectable 0.803, suggesting that a strong performance on this task does not necessarily require large models. DeepSeek V3.1 is an outlier with the highest precision (0.965), but the lowest recall (0.494), indicating that it extracts very few, but highly accurate statements. In contrast, the best-performing models achieve a more balanced trade-off between precision and recall, resulting in higher overall effectiveness.

Discussion. The results highlight the importance of balancing precision and recall in Statement

Extraction. Models such as GPT-5.2 and Gemini 3 Flash achieve strong F1 scores by maintaining consistency across both metrics, rather than optimizing for one at the expense of the other. On the other hand, the behavior of DeepSeek V3.1 illustrates a precision-oriented strategy that may be useful in applications where false positives are particularly costly, but less suitable for comprehensive information extraction. Overall, these findings indicate that recent compact models (e.g., GPT-5 Mini) can rival or outperform larger systems.

A closer qualitative analysis revealed additional challenges beyond overall performance. In particular, resolving citation scope remains highly difficult for LLMs—consistent with the difficulties observed for human annotators in Section 4.3.1. Consider the sentence "Broadly speaking, agentive verbs usually occur with an ergative subject and the auxiliary edun 'have', whereas patientive verbs combine with an absolutive subject and the auxiliary izan 'be' (Levin 1983)" from Pineda and Berro (2020). In the human annotation, the citation is applied to both coordinated statements: the statement on agentive verbs and the statement on patientive verbs. However, the automatically extracted statement, which applies the citation only to the latter, is also a plausible interpretation.

6.2. Task 2: Statement Classification

We evaluated how well LLMs classify the 342 gold statements into the categories defined in Section 3.2.

Evaluation metrics. For each model, we compute macro-averaged precision, recall, and F1-score against the gold labels.

Model	Prec.	Rec.	F1
Gemini 3 Flash	**0.824**	**0.890**	**0.840**
GPT-5.2	0.789	0.883	0.811
DeepSeek V3.1	0.774	0.870	0.805
GPT-5 Nano	0.747	0.799	0.767
Gemini 2.5 Flash	0.715	0.846	0.750
Kimi K2 Instruct	0.724	0.788	0.744
Kimi K2.5	0.733	0.770	0.742
GPT-5 Mini	0.679	0.683	0.680

Table 4: Statement Classification evaluation.

Results. Table 4 reports the performance of all evaluated models on task 2. Gemini 3 Flash achieves the best overall performance (0.840), consistent with its strong extraction performance. GPT-5.2 (0.811) and DeepSeek V3.1 (0.805) follow closely. While Gemini 2.5 Flash and the Kimi models obtain competitive results, they remain below the top-performing systems. GPT-5 Nano again demonstrates strong efficiency, achieving an F1 of 0.767 despite its smaller size. In contrast, GPT-5 Mini records the lowest performance (0.680), indicating greater variability across models for this task. In terms of individual metrics, most models maintain a relatively balanced precision–recall profile, although some variation is observed.

Model	Acc.	Prec.	Rec.	F1
Gemini 3 Flash	0.718	0.566	0.783	**0.657**
Kimi K2.5	0.680	0.526	0.740	0.615
Kimi K2 Instruct	0.643	0.490	**0.804**	0.609
DeepSeek V3.1	0.646	0.492	0.762	0.598
Gemini 2.5 Flash	0.649	0.494	0.736	0.591
GPT-5.2	**0.750**	**0.684**	0.515	0.587
GPT-5 Mini	0.703	0.567	0.596	0.581
GPT-5 Nano	0.696	0.575	0.455	0.508

Table 5: Overall Linguistics Knowledge Evaluation.

Discussion. Unlike task 1, where several models achieve similar performance, classification has more divergence across systems. Top-performing models such as Gemini 3 Flash and GPT-5.2 maintain a strong balance between precision and recall, which is crucial for achieving robust macro F1 performance across classes.

6.3. Task 3: Linguistic Knowledge Evaluation

As stated in Section 5, we used a challenge dataset of 705 statements—235 gold statements and 470 LLM-generated negations.

Evaluation metrics. We used macro-averaged accuracy, precision, recall, and F1-score against the gold labels. Beyond the binary verdict, we also analyzed the self-assessed confidence score[3] of each model.

Results. Table 5 presents the overall results of the Linguistics Knowledge Evaluation. Gemini 3 Flash achieves the highest F1 score (0.657) alongside strong recall (0.783), indicating robust coverage of linguistic phenomena. Across the eight models, accuracy ranges from 64.3% (Kimi K2 Instruct) to 75% (GPT-5.2), a spread of 11 percentage points. Within model families, a scaling effect is visible for OpenAI: GPT-5.2 (75%) $>$ GPT-5 Mini (70.3%) $>$ GPT-5 Nano (69.6%). For the Kimi family, Kimi K2.5 maintains a more balanced precision–recall profile, whereas Kimi K2 Instruct favors higher recall (0.804) at the expense of precision.

We examined whether the confidence score is a reliable indicator of accuracy by analyzing GPT-5.2, the best-performing model. Figure 2 plots accuracy

[3]All API calls use the provider default temperature (1.0 for OpenAI, Gemini, and Fireworks), since we are interested in the models' calibrated confidence rather than deterministic output.

and false-statement detection precision as a function of the model's own confidence. Statements are grouped by confidence threshold: each point shows the metric computed on the subset of statements for which the model reported confidence $\geq t$. The relationship is monotonic and substantial: at confidence ≥ 0.9, GPT-5.2 achieves 92.7% accuracy and 97.2% precision for false-statement detection ($n = 55$), compared with 73.4% and 83.0% over all 705 statements. A point-biserial correlation confirms the pattern (r=0.13, p<0.001); when restricted to true statements, the correlation between confidence and correct acceptance rises to r=0.39 (p<10^{-5}). Gemini 3 Flash, by contrast, reports confidence ≥ 0.8 for all the statements (mean = 0.93), leaving almost no room for threshold-based filtering. Its confidence score is therefore less informative as a triage signal, despite the model's competitive overall accuracy (71.8%).

A more extensive analysis of the models' performance by linguistic statement class is included in Table 6. Six out of eight models perform best on theoretical statements (L-Theo). Language-specific statements (L-Spec) are the most challenging, likely because they require knowledge of particular language data that may be underrepresented in training corpora. The L-Theo advantage over L-Spec is statistically significant for most models (chi-squared test: p<0.01 for Gemini 3 Flash, GPT-5 Mini, GPT-5 Nano, and Kimi K2.5; p<0.05 for Gemini 2.5 Flash and Kimi K2 Instruct) but not for GPT-5.2 (p=0.15) or DeepSeek V3.1 (p=0.12), suggesting that the best and worst models are less affected by statement type. Typological statements (L-Typo, n=30) show high variance due to the small sample size. The overall error rate (26–38%) indicates that current LLMs have non-trivial but insufficient linguistic knowledge for reliable fact-checking, reinforcing the need for human oversight or retrieval-augmented approaches.

Discussion. The results reveal a clear distinction between high-recall and high-precision modeling strategies. Models such as Gemini 3 Flash and Kimi variants prioritize recall, suggesting they are better suited for tasks requiring broad linguistic coverage, such as error detection or phenomenon identification. However, this often comes with reduced precision, potentially introducing more false positives. GPT-5 models exhibit a precision-oriented behavior, which may be advantageous in applications where correctness is critical (e.g., annotation pipelines), but their lower recall indicates limited coverage of diverse linguistic cases.

6.4. Tasks Comparison

Figure 3 highlights substantial heterogeneity across the three evaluation dimensions.

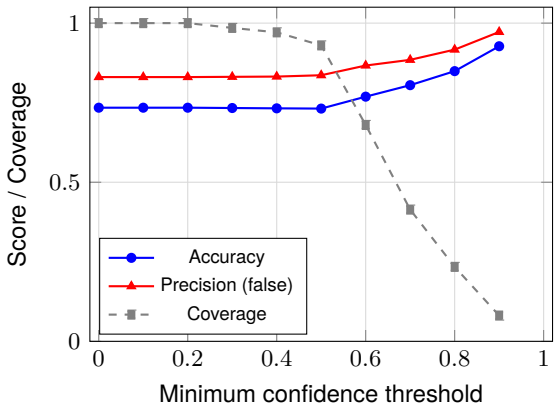


Figure 2: GPT-5.2 veracity performance as a function of minimum confidence threshold.

Model	Class	Acc.	Prec.	Rec.	F1
Gem. 3	L-Spec	0.67	0.51	0.76	**0.61**
	L-Theo	**0.78**	0.64	**0.80**	**0.71**
	L-Typo	0.67	0.50	0.90	0.64
Kimi k2.5	L-Spec	0.63	0.48	0.72	0.57
	L-Theo	0.74	0.60	0.75	0.67
	L-Typo	0.57	0.42	0.80	0.55
Kimi k2	L-Spec	0.61	0.46	**0.79**	0.58
	L-Theo	0.69	0.53	**0.80**	0.64
	L-Typo	0.60	0.45	**1.00**	0.62
DS V3.1	L-Spec	0.62	0.47	0.73	0.57
	L-Theo	0.68	0.52	0.78	0.63
	L-Typo	0.60	0.45	0.90	0.60
Gem 2.5	L-Spec	0.58	0.43	0.67	0.52
	L-Theo	0.73	0.58	**0.80**	0.67
	L-Typo	0.60	0.44	0.80	0.57
GPT-5.2	L-Spec	**0.73**	**0.64**	0.47	0.54
	L-Theo	0.77	**0.73**	0.55	0.63
	L-Typo	**0.77**	**0.64**	0.70	0.67
GPT-5m	L-Spec	0.65	0.50	0.53	0.51
	L-Theo	0.75	0.64	0.65	0.64
	L-Typo	**0.77**	0.62	0.80	**0.70**
GPT-5n	L-Spec	0.63	0.45	0.35	0.39
	L-Theo	0.76	0.70	0.55	0.62
	L-Typo	0.73	0.60	0.60	0.60

Table 6: Model performance on Linguistics Knowledge Evaluation by class.

Gemini 3 Flash is the strongest model overall, with the highest scores on the three tasks(Extraction 0.894, Classification 0.840, Linguistic Knowledge Evaluation 0.657).

The remaining models exhibit a more asymmetric behavior. GPT-5.2 is consistently high on Extraction and Classification, but weaker on Linguistic Knowledge Evaluation. DeepSeek V3.1 performs

substantially better on classification than on extraction, indicating that its conservative prediction strategy may be better suited to labeling predefined statements than identifying them in free text. Kimi variants are comparatively stronger on Linguistic knowledge than most models.

Overall, the results suggest that no single model family dominates uniformly across all tasks except Gemini 3 Flash. This reinforces the importance of evaluating models across multiple subtasks when assessing their suitability for complex information extraction pipelines.

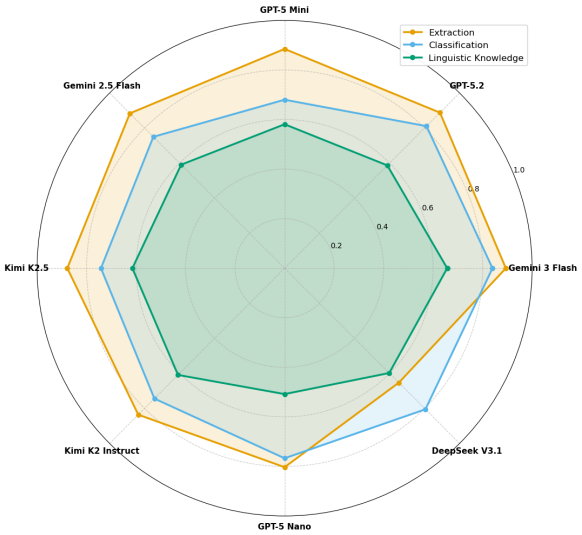


Figure 3: Overal F1-scores across the three tasks.

7. Conclusion

We have presented a comprehensive pipeline for extracting, classifying, and verifying linguistic claims from scientific articles. Our main findings are:

1. **Statement Extraction**: Gemini 3 Flash achieves the best F1 (0.894) for extracting atomic linguistic statements, with most models scoring above 0.80.

2. **Statement Classification**: Gemini 3 Flash again leads (F1 = 0.840), and inter-annotator agreement for classification ($\alpha = 0.750$) is substantially higher than for extraction ($\alpha = 0.299$).

3. **Linguistic knowledge Evaluation**: Current LLMs have limited but non-trivial linguistic knowledge. On a challenge set of 705 statements evaluated with eight models, the best accuracy is 75% (GPT-5.2), with theoretical statements being easier than language-specific ones. However, the top four models (accuracy 69–75%) are statistically indistinguishable (McNemar's test, all $p > 0.13$). Models range from conservative (GPT-5 Nano, high precision, low recall) to permissive (Kimi K2 Instruct, high recall, low precision), with GPT-5.2 achieving the best accuracy–recall trade-off. Within model families, larger models consistently outperform smaller ones.

We expanded this work to the rest of the paragraphs in the 11 Glossa articles. From the 1,082 unique paragraphs, we ended up with 3,655 unique sentences. A total of 7,713 atomic statements were extracted from the sentences, averaging 2.11 statements per sentence. All of the 7,713 atomic statements have been classified into L-Spec, L-Typo, L-Theo, C, or S.

Future work will extend the Linguistic Knowledge Evaluation to additional models and LLM-as-a-judge approaches. Another promising direction is systematic prompt optimization: all prompts in the current pipeline were manually designed, and frameworks such as DSPy (Khattab et al., 2023) could be used to automatically compile and tune prompt chains, potentially improving both extraction and veracity performance.

While the current collection already enables a first systematic evaluation, scaling it to several hundred thousand atomic statements annotated for L-Spec, L-Typo, and L-Theo would enable a more detailed and statistically reliable assessment of LLMs' linguistic knowledge. In particular, expanding the dataset to cover a broader and more diverse set of languages would allow us to test an important hypothesis: that LLMs capture the structural properties of well-resourced languages far better than those of typologically rare or low-resource languages. Such a resource would also provide the foundation for a standardized benchmark for linguistic knowledge evaluation, enabling systematic and reproducible comparison across models, prompting strategies, and future generations of LLMs.

Our long-term research goal is to incorporate corpus and treebank-based verification for L-Spec statements in UD-covered languages and investigate whether providing LLMs with retrieved context (e.g., relevant treebank data) improves their performance. Equally, we plan to extend our work on more typology-oriented journals, where we will verify with established typological databases such as WALS, Grambank, and APiCS.

8. Bibliographical References

Georgios Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. 2021. A review on fact extraction and verification. *ACM Computing Surveys*, 55(1):1–35.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: Why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93, Denver, CO. Association for Computational Linguistics.

Catherine Blake. 2010. Beyond genes, proteins, and abstracts: Identifying scientific claims from full-text biomedical articles. *Journal of Biomedical Informatics*, 43(2):173–189.

Clément Bleuze. 2024. *Analysing Claims in NLP Research*. Ph.D. thesis, INRIA.

Jifan Chen, Grace Kim, Aniruddh Sriram, Greg Durrett, and Eunsol Choi. 2024. Complex claim verification with evidence retrieved in the wild. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1).

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. 2022. CS-KG: A large-scale knowledge graph of research entities and claims in computer science. In *International Semantic Web Conference*. Springer.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Tobias Jansen and Tobias Kuhn. 2016. Extracting core claims from scientific articles. In *Benelux Conference on Artificial Intelligence*, pages 33–46. Springer.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, et al. 2023. DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2018. Information extraction from scientific articles: A survey. *Scientometrics*, 117:1931–1990.

Anna Pineda and Ane Berro. 2020. Hybrid intransitives in basque. *Glossa: a journal of general linguistics*, 5(1):22.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103. Association for Computational Linguistics.

Akshith Reddy Putta, Jacob Devasier, and Chengkai Li. 2025. ClaimCheck: Automatic fact-checking of textual claims using web evidence. In *Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing*, pages 303–316, Albuquerque, New Mexico, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mourad Sarrouti, Asma Ben Abacha, Yassine M'rabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational*

Linguistics, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Juraj Vladika and Florian Matthes. 2023. Scientific fact-checking: A survey of resources and approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Wu Youyou, Yang Yang, and Brian Uzzi. 2023. A discipline-wide investigation of the replicability of psychology papers over the past two decades. *Proceedings of the National Academy of Sciences*, 120(6).

A. Prompts

A.1. Prompts used for Sentence Extraction

A.1.1. Paragraph segmentation

You are an expert assistant helping a linguist segment a scientific linguistics article into sentences.

Task: Split each numbered paragraph below into grammatically complete sentences. Preserve the original text exactly—do not paraphrase, correct, or reorder. Process each paragraph independently.

Rules: 1. Complete sentences only: Each output unit must be a grammatically complete sentence. Merge fragments (dangling clauses, isolated citations, orphan parentheticals) with the sentence they belong to. 2. Linguistic examples: Numbered examples (e.g., "(1)", "(2a)") must be attached to the sentence that introduces or discusses them *without* their glosses/translations. Keep only the original text, removing gloss lines and translation. Input: "Example (1) shows the problem in Spanish. (1) El niño duerme. 'The boy sleeps.'" Output: ["The Example "El niño duerme." shows the problem in Spanish: "] Linguistic examples that are not cited or attached to any sentence must be completely removed. 3. In-text citations: Keep citations attached to their sentence. 4. Lists and enumerations: If a sentence introduces a list, keep short list items with the introducing sentence. Split only if items are full sentences themselves. 5. Section titles: Titles/headings may remain as standalone fragments (the only exception to rule 1). 6. Preserve numbering: Only include numbers if the original text has them.

A.1.2. Statement Extraction

You are an expert linguist extracting verifiable atomic statements from a linguistics article. Process each paragraph block below independently.

Core rules — each extracted statement must: 1. Be atomic: split coordinations ("A and B") into separate statements when each conjunct is independently meaningful. When atomicity is ambiguous, do not split. Exception: joint reference lists stay together (e.g., "Smith (2020) and Miller (2023) show that. . . "). 2. Be a factual assertion that can be independently understood and assessed. 3. Be directly and literally inferable from the source text — no chained inferences (from "A then B" and "B then C", extract only those two, never "A then C"). 4. Be self-contained: resolve all pronouns/references (e.g., "it" then "the subject NP") and expand abbreviations (e.g., "NP" then "noun phrase (NP)"). 5. Preserve technical precision. Always include the language name when the context makes it clear (e.g., "The verb agrees. . . " then "In Spanish, the verb agrees. . . "). 6. Fold linguistic examples into the statement text; remove example numbers, glosses, and translations, keeping only the original-language form. 7. Pure conditional ("if A then B"): extract only the whole implication as one statement; do not extract A or B separately. 8. Causal/inferential ("A. Hence B"): extract (a) A as a standalone statement, and (b) the implication "A implies B". Do NOT extract B alone. - Source: "Prepositions act as cues to retrieve the PP correlate. Hence, P-omission in sluices comes with a processing cost." - (a) "Prepositions act as cues to retrieve the PP correlate." (b) "Prepositions acting as cues to retrieve the PP correlate implies that P-omission in sluices comes with a processing cost." 9. If the author attributes a claim to another work, keep the attribution. If the author also endorses the claim, additionally extract the general (non-attributed) version (e.g., "X (2024) uses Y to prove Z" → also "Y can be used to prove Z"). 10. Ambiguous citation scope: assume the smallest scope (only the immediately preceding clause). Text after the citation is the current author's claim unless marked otherwise. 11. Quoted cited text: preserve verbatim; do not split further. 11. "X followed Y in their analysis of Z" then (a) "Y analyzes Z" and (b) "X uses the same methodology as Y for Z". 12. Non-restrictive "which" clauses yield an additional statement (e.g., "French, which is Indo-European, . . . " becomes "French is an Indo-European language."). If restrictiveness is uncertain, do not split.

A.2. Prompt used for Statement Classification

You are an expert linguist reviewer classifying statements from a linguistics article according to specific guidelines.

Statements to classify: {paragraph blocks}

Categories: * C (Citation/Attribution): State-

ments attributing an idea or data to another author or paper. Example: "As noted by Chomsky (1995), the minimal link condition applies." Ambiguity: If the statement is "according to" a paper or person, it is Type C. But if it is "according to" a rule or principle (even if attributed earlier), it is likely Type L (author asserting the rule's validity). Scope: "Ginzburg & Sag (2000) proposed [X]" is C. But the continuation "an interrogative sluice is not derived..." is L (if asserted by the current authors).

* L (Linguistic Statement): The authors themselves assert the claim. Sub-types:

- L-Spec (Linguistic - Specific): Empirical claims about a *specific* language. MUST NOT make comparisons with other languages. Verifiable on language data (corpora, treebanks). Language Field: You MUST specify the language name for L-Spec statements. Example: "Warlpiri is mostly SOV." (Language: Warlpiri) - L-Typo (Linguistic - Typological): Cross-linguistic phenomena or comparisons. Describes phenomena across languages ("All VSO languages have prepositions"). Compares a set of languages ("Sino-Tibetan languages tend to be more..."). Compares specific languages ("Polish is similar to Russian..."). - L-Theo (Linguistic - Theoretical): Theory-internal generalizations or claims. Not directly verifiable with raw language data (treebanks, corpora). Abstract concepts (e.g., "Move-α is blocked by locality constraints"). Evaluating theories ("This mismatch casts doubt on syntactic theories...").

* S (Structural/Methodological): Statements that do not make a scientific claim. Section titles ("3.2 Theoretical Background"). Definitions ("We define <term> as..."). Meta-discourse/Signposting ("Section X will review...", "The authors studied...", "This article"). Data information ("Our dataset contains 1200 sentences"). Methodology ("We use the Fisher test...", "We asked 12 native speakers..."). Technical/Algorithmic results ("It took 12 seconds", "20% were rejected").

Important Guidelines for Ambiguity: 1. If there is a citation in the statement -> Type C. (e.g., "Many non-P-stranding languages allow P-omission (Fortin 2007)..." -> Type C). 2. "According to [Rule/Principle]" -> Type L. 3. "According to [Person/Paper]" -> Type C.

Instructions: For each statement, provide: - Primary 'class' (L-Spec, L-Typo, L-Theo, C, S) and 'confidence' (0-1). - If L-Spec, provide the 'language' (e.g., "French", "Warlpiri"). - Dictionary/standard name. If not L-Spec, use null. - 'secondary_class' and 'secondary_confidence'. - Keep the 'statement_idx'.

A.3. Prompts used for Linguistic Knowledge Evaluation

A.3.1. Statement Negation

You are a researcher writing a paper. For each statement, write a statement that is a STRICT logical contradiction of the claim below: both statements cannot be true at the same time.

Rules: For each statement, - Keep the wording and structure as close as possible. - The change must reverse the truth conditions of the claim (not just express a different opinion). - Do not use "no", "not", or explicit negation. - Keep about the same length. Return a JSON object with a single key "contradiction".

A.3.2. Linguistic Knowledge Evaluation

Role: Linguistic Fact-Checker (Syntax/Morphosyntax). Task: Evaluate the accuracy of STATEMENT based on descriptive academic linguistics.

RULES: - No prescriptive grammar; use descriptive evidence. - If the statement is a single word or lacks propositional content, veredict is false. - Lower confidence for theoretical disagreements. - Return exactly one JSON object per statement, in order.

The Software Mention Detection and Coreference Resolution Shared Task 2026

Sharmila Upadhyaya[1], Wolfgang Otto[1,2], Julia Matela[3]
Frank Krüger[3], Stefan Dietze[1,2]
[1]GESIS – Leibniz Institute for the Social Sciences, Germany
[2]Heinrich Heine University Düsseldorf, Germany
[3]Wismar University of Applied Sciences, Germany
{sharmila.upadhyaya, wolfgang.otto, stefan.dietze}@gesis.org
{julia.matela, frank.krueger}@hs-wismar.de

Abstract

Software is referenced in research papers in many different ways: full names, abbreviations, misspellings, versioned names, or indirect references via websites and citations. This makes it hard to link mentions to a single software entity, which in turn limits large-scale analyses and knowledge graph construction. The Software Mention Detection and Coreference Resolution (SOMD) shared task 2026, organized at the Natural Scientific Language Processing (NSLP) workshop at LREC 2026, focuses on clustering software mentions that refer to the same software entity. We provide three subtasks covering gold mentions, automatically extracted mentions, and mentions sampled at scale from large-scale publications. Systems are evaluated with established coreference metrics (MUC, B^3, CEAFe) and their CoNLL average. This paper describes the task setup, datasets, evaluation, baseline, and the observed patterns in participant submissions, and outlines future directions for scalable software mention coreference resolution. The shared task was concluded with total five registered participants, with total 43 submissions for all subtasks. Finally, two system papers were submitted with competitive performance against baselines.

Keywords: software mentions, cross-document coreference, scholarly NLP, knowledge graphs, shared task

1. Introduction

Software has emerged as a tool in scientific research, yet it is often cited and described informally in scholarly articles. A single software can appear under multiple surface forms (e.g. different spellings or abbreviations), and a single surface form can refer to other software packages depending on context. As a result, building reliable links from text mentions to software entities remains difficult, especially at the scale of web or biomedical corpora. The Software Mention Detection and Coreference Resolution (SOMD) shared task 2026 targets this problem by posing a challenge that requires grouping (clustering) software mentions referring to the same underlying software entity across multiple documents. The task is widely studied under the framework of cross-document coreference resolution in the Information Extraction community. SOMD 2026 is hosted at the Natural Scientific Language Processing (NSLP) workshop at the Language Resources and Evaluation Conference (LREC) 2026 (NFDI4DataScience, 2026a,b). The shared task follows the earlier SOMD editions described in Krüger et al. 2024 and Upadhyaya et al. 2025, which focused on software-related information extraction. This iteration adds a dedicated evaluation setting for scalable cross-document coreference resolution (Keshtkaran et al., 2017).

Cross-document coreference resolution of software mentions supports downstream tasks, such as Knowledge Graph (KG) construction, in which software entities are linked to publications, authors, datasets, and research outcomes. SoftwareKG is a prominent example of a large-scale knowledge graph that represents software mentions extracted from a large corpus and supports analyses of software use and citation practices (Schindler et al., 2022; GESIS, 2026). Moreover, the quality of coreference resolution directly affects KG quality; collapsing either distinct software into a single node or splitting a single software into multiple nodes, both of which distort statistics and graph queries. Similarly, reliable cross-document coreference resolution supports reproducibility by consolidating mentions of the same underlying software entity. This enables subsequent linking to stable software identifiers (e.g., repository URLs, DOIs, or registry identifiers). This helps researchers identify the exact software used in a study and supports indexing in discovery services. Recent work on software citation and discoverability highlights persistent gaps between recommended citation practices and real-world referencing in scholarly articles (Katz and Chue Hong, 2024). Finally, resolving software mentions across documents enables analytics at scale, such as tracking software adoption across domains, identifying tool dependencies, and studying the impact of software releases. These use cases motivate shared, open benchmarks that combine high-

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 247–254

quality gold annotations with realistic noisy inputs.

The first and second SOMD shared tasks addressed software-related information extraction from scholarly publications. SOMD 2024 (Krüger et al., 2024) introduced a benchmark for detecting software mentions and related information in context and reported substantial variation across systems and subtasks. SOMD 2025 (Upadhyaya et al., 2025) extended the scope and introduced a challenging setting for software-related information extraction, while continuing to build community baselines and evaluation practices. Building on last year's challenge, SOMD 2026 focuses on the consecutive steps of the information extraction pipeline: cross-document coreference resolution. Earlier tasks treat mentions as local objects (within a document or sentence). However, for KG construction and large-scale analytics, mentions must be grouped across documents. SOMD 2026, therefore, frames coreference resolution as a cross-document coreference clustering problem over software mentions, using established coreference evaluation metrics.

Cross-document coreference resolution faces challenges such as surface-form variation (e.g., spelling differences and abbreviations), ambiguity in short names (e.g., "R", "SAS"), sparse metadata, and domain shifts across disciplines (Schindler et al., 2021). One of the central constraints for large-scale coreference resolution is scalability, as naive mention-pair approaches have $O(n^2)$ complexity and become infeasible at the corpus scale. Therefore, high-accuracy models based on cross-encoders or large language models are computationally expensive and difficult to deploy for large knowledge graphs. SOMD 2026 thus emphasizes search space reduction, efficient similarity search (e.g., approximate nearest neighbors), and the use of available metadata, while evaluating systems under both clean and noisy input conditions.

SOMD 2026 makes the following contributions:

- **A shared evaluation setting** for cross-document software mention coreference resolution.
- **Three subtasks** covering gold mentions from SoMeSci (Schindler et al., 2021), automatically extracted mentions, and large-scale samples from SoftwareKG (NFDI4DataScience, 2026b).
- **New and extended coreference annotations** derived from SoMeSci and SoftwareKG (Schindler et al., 2021, 2022).
- **A TF–IDF + DBSCAN baseline** (Salton and Buckley, 1988; Ester et al., 1996).
- **Baseline II (Semantic Centroids with Hierarchi- cal Density-Based Clustering.)** proposing blocking as scalable cross document coreference resolution approach.
- **Standard evaluation** using MUC, B^3, CEAFe, and the CoNLL average (Vilain et al., 1995; Bagga and Baldwin, 1998; Luo, 2005; Pradhan et al., 2012).

This paper summarises the challenge (January 20 to February 20, 2026), describing the subtasks, datasets, evaluation setup, baseline, and participating systems, and discusses scalability as the central challenge. The code to reproduce both baselines is publicly available online.[1][2]

2. Task Description

The shared task focuses on cross-document coreference resolution of software mentions in scholarly articles, formulated as a clustering problem.

Subtask 1: Coreference Resolution over Gold Mentions Given a set of gold-standard software mentions across multiple documents, systems must cluster mentions that refer to the same underlying software entity. Input is a list of software mentions with (i) mention string, (ii) sentence-level context, and (iii) optional metadata (e.g., URL, developer/author, references). Expected output is mapping from `mention_id` to `cluster_id`.

Subtask 2: Coreference Resolution over Predicted Mentions Given automatically extracted software mentions across multiple documents, systems must cluster mentions referring to the same software entity under realistic extraction noise. Input includes automatically detected mentions in the same format as Subtask 1. Output is the same as Subtask 1, i.e., a mapping from `mention_id` to `cluster_id`. Evaluation is performed on mentions that can be aligned to gold cluster annotations.

Subtask 3: Coreference Resolution at Scale Given an extensive collection of automatically extracted software mentions sampled at scale, systems must efficiently perform cross-document clustering. Input consists of automatically extracted mentions at a large scale, and output is a mapping from `mention_id` to `cluster_id`, evaluated on an annotated subset.

3. Dataset

SOMD 2026 builds directly on the datasets and annotations developed in previous SOMD shared

[1] Baseline I (TF–IDF + DBSCAN): https://github.com/sarmilaupadhyaya/SOMD2026

[2] Baseline II (Semantic Centroids with Hierarchi- cal Density-Based Clustering.): https://github.com/matjulia/somd2026

tasks, which focused on software mention detection and related attribute extraction. Consequently, we reuse already annotated or automatically predicted software entities and their associated metadata as input for the present task. Depending on the subtask, these mentions are either gold-standard annotations from SoMeSci (Schindler et al., 2021) (Subtask 1) or predicted outputs from prior extraction models (Subtasks 2 and 3) from SoftwareKG (Schindler et al., 2022). The current shared task addresses the subsequent stage of the pipeline, namely, cross-document coreference resolution over these pre-identified mentions.

3.1. Background: SoMeSci and SoftwareKG Datasets

SoMeSci is a gold standard dataset of software mentions in scientific articles, including disambiguation of spelling variants and additional information such as version, developer, URL, and citations (Schindler et al., 2021). It provides a high-quality starting point for defining mention-level clusters. SoftwareKG is a large-scale knowledge graph of software mentions extracted from PubMed Central articles, enabling analyses of software usage and citation practices (Schindler et al., 2022; GESIS, 2026). It serves as the basis for the scale-oriented sampling in Subtask 3.

3.2. Dataset Creation for SOMD 2026

Building on SoMeSci and SoftwareKG, we construct three subtask-specific datasets that differ in input quality and scale.

Subtask 1 For Subtask 1, we use the manually annotated SoMeSci (Schindler et al., 2023) gold-standard mentions and convert them into a unified mention-list format. Each mention is represented with its surface form, sentence-level context, and available metadata (e.g., developer, URL, citation markers). Cross-document coreference clusters are derived from the canonical software identifiers in SoMeSci. The dataset is split into 80% for training and 20% for testing.

Subtask 2 For Subask 2, we use a subset of automatically extracted mentions from SoftwareKG dataset (Schindler et al., 2022) consisting of mentions extracted from the same document collection as in SoMeSci (Schindler et al., 2023). These predicted mentions are converted into the unified list format using the same logic as in Subtask 1. To enable evaluation, we align predicted mentions with existing gold clusters whenever possible and identify *linkable* mentions. The linkable dataset is split into 80% training and 20% test data. The test set is extended by remaining mentions from the same document collection and 2,000 manually annotated mentions from SoftwareKG and, and evaluation is performed on the subset of the test split consisting of linkable or manually annotated mentions.

Subtask 3 For Subtask 3, we apply the SoMeSci extraction model to a larger subset of PubMed Central articles using the SoftwareKG pipeline. From the resulting large pool of predicted mentions, we sample mentions at scale. A subgroup is manually annotated with cross-document coreference clusters for evaluation, while the remaining mentions are used to assess scalability and runtime behavior.

Table 1 provides an overview of the dataset composition across all subtasks and splits, including the number of documents, software mentions, and cross-document coreference clusters. The test splits clusters are withheld and used for the ranking of the leaderboard on CodaBench. This overview illustrates the differences in scale and annotation type across Subtasks 1–3.

The datasets, baseline code, and evaluation scripts are released via the shared task website (NFDI4DataScience, 2026b).

4. Evaluation Metrics

We evaluate systems using standard coreference metrics:

- **MUC** measures how many links between mentions are needed to transform predicted clusters into gold clusters and tends to reward systems that produce fewer, larger clusters (Vilain et al., 1995).
- **B^3** computes precision and recall per mention based on overlap between predicted and gold clusters and is sensitive to both over-merging and over-splitting (Bagga and Baldwin, 1998).
- **CEAFe** aligns predicted clusters to gold clusters in a one-to-one fashion and scores based on the best alignment, penalizing both fragmentation and merging in an entity-centric way (Luo, 2005).

Following common practice in shared tasks, we also report the **CoNLL F1** score as the unweighted average of the F1 values of MUC, B^3, and CEAFe (Pradhan et al., 2012). Using multiple metrics is important because each metric emphasizes different error modes: MUC can mask over-merging, while CEAFe and B^3 react more strongly to cluster purity and fragmentation.

5. Participation and Approaches

A total of 5 participants registered for the SOMD 2026 shared task. A team fully participated by sub-

	Train			Test		Eval	
Subtask	Docs	Mentions	Clusters	Docs	Mentions	Mentions	Clusters
Subtask 1 (Gold Standard)	973	2,974	733	244	743	743	250
Subtask 2 (Predicted)	967	2,860	699	1,977	12,516	2,697	719
Subtask 3 (Predicted at Scale)	967	2,860	699	173,450	219,950	10,454	1,893

Table 1: Dataset statistics across Train and Test splits for three subtasks. Subtasks 2 and Subtask 3 share identical training sets. Subtask 3 incorporates an extended test set focusing on large scale analysis. The Eval set is the subset of the Test set used for system scoring. Consequently, Subtask 1 has identical Test and Eval splits.

mitting results for all three subtasks, while one team participated in only two subtasks. All three teams provided a system description. In this section, we introduce the three final approaches, along with a baseline model.

5.1. Baseline Systems

Baseline I: TF–IDF + DBSCAN.
We provide a baseline that clusters mentions using vector-space similarity and density-based clustering. Each mention is represented by a TF–IDF vector over character and word n-grams derived from (i) the mention surface form and (ii) a short context window (Salton and Buckley, 1988). Optional metadata fields (URL host, developer string) are appended as tokens when present. We apply DBSCAN with cosine distance on TF–IDF vectors (Ester et al., 1996). DBSCAN does not require the number of clusters as input, which fits open-world disambiguation. Hyperparameters (`eps`, `min_samples`) are tuned on the development split.

Baseline II: Semantic Centroids with Hierarchical Density-Based Clustering.
The second baseline (Matela and Krüger, 2026) is a hybrid framework based on Sentence-BERT embeddings, combining FAISS-based retrieval over training-set cluster centroids with HDBSCAN clustering for unassigned mentions, complemented by surface normalization and blocking strategies for scalability. The blocking strategy partitions mentions first by entity type and then by the first letter of the canonical name. A detailed system description is provided in Matela and Krüger 2026.

5.2. Participants' System

Table 2 provides an overview of the submitted systems. Team **aalkan** from the Harvard-Smithsonian Center for Astrophysics submitted two unsupervised, fine-tuning free systems for all three subtasks. The first system, Fuzzy Matching (FM), computes pairwise lexical similarity between mention strings using the Ratcliff/Obershelp algorithm and applies transitive closure over linked pairs to form clusters. The second system, Context-Aware Representations (CAR), encodes each mention using the lightweight `all-MiniLM-L6-v2` sentence transformer, combining a mention-level embedding with a document-level embedding aggregated from up to ten mention-bearing sentences, and clusters the resulting representations using agglomerative clustering with cosine distance. CAR consistently outperforms FM by approximately one CoNLL F1 point across all subtasks, with the advantage most visible on CEAFe, reflecting improved cluster purity. A controlled noise-injection study reveals complementary failure modes: CAR is substantially more robust to mention boundary errors, while FM degrades more gracefully under mention substitution. In terms of scalability, FM scales superlinearly with corpus size due to its pairwise comparison step, whereas CAR encodes each mention independently and scales approximately linearly, making it the more practical choice for large-scale pipelines.
mhassan from ZBW Leibniz Information Centre for Economics and the University of Greifswald submitted a supervised neural system for Subtasks 1 and 2. The system follows a three-stage pipeline. First, each mention is represented as a structured input string of the form `mention_text [SEP] sentence_context` and encoded using a SciBERT model (Beltagy et al., 2019) trained with Supervised Contrastive (SupCon) loss (Khosla et al., 2021), which optimizes all mention pairs within a batch simultaneously, pulling coreferent mentions together and pushing non-coreferent mentions apart in a shared 256-dimensional embedding space. Second, a set of software-aware heuristics adjusts the pairwise cosine similarity matrix before clustering: a name canonicalization boost ($\delta = +0.5$) is applied when two mention strings normalize to the same canonical form, and a developer conflict penalty ($\delta = -0.8$) is applied when metadata indicates different software developers. Third, Hierarchical Agglomerative Clustering (HAC) with average linkage and a fixed distance threshold of $\theta = 0.5$ produces the final clusters. To prevent

data leakage, the model is trained using cluster-level splitting, ensuring that all mentions of a given software entity are assigned exclusively to either the training or validation set. The system achieves a CoNLL F1 of 0.92 on both subtasks.
Neither participant system demonstrated practical scalability for Subtask 3. *mhassan* did not submit results for Subtask 3, likely due to the $\mathcal{O}(n^2)$ complexity of HAC over 219,950 mentions. *aalkan* completed Subtask 3 in approximately 2.2 hours using FM, compared to approximately 3 minutes for Baseline II, highlighting that blocking strategies are not merely an optimization but a practical necessity for cross-document coreference resolution at scale.

5.3. Shared Task Implementation

SOMD 2026 is hosted on CodaBench (Xu et al., 2021; CodaBench, 2026). The shared task schedule follows the official task page (NFDI4DataScience, 2026b), with training and development data released on January 20, 2026, and the competition phase closing on February 20, 2026. System description papers were due on March 10, 2026 (see the task website for details).

Participants submit a file that assigns a predicted `cluster_id` to each `mention_id` in the test set. The platform computes MUC, B^3, CEAFe, and CoNLL F1 and publishes leaderboards per subtask.

6. Results and Discussion

6.1. Participants and Overall Results

Two teams submitted valid systems to the shared task: *aalkan*, and *mhassan*. *aalkan* participated in all three subtasks, while *mhassan* submitted results for Subtasks 1 and 2. Across subtasks in which both teams competed the ranking was consistent, suggesting that the participants used approaches that generalized well across the different input conditions and that the relative system strengths were stable under the task variations.

Baseline II outperforms both participants' system across every subtask what demonstrates that the hybrid

Table 3 presents the final results. Compared to the TF-IDF + DBSCAN baseline (Baseline I), all submitted systems achieve consistent improvements in CoNLL and CEAFe, particularly in Subtasks 1 and 2, indicating that embedding-based and hybrid approaches provide more precise entity-level clustering. For the competition phase, *aalkan* (0.96, 0.96, 0.94) achieved the highest CoNLL score for three respective subtasks, following *mhassan* (0.92, 0.92). Additionally, our Baseline II achieved the overall performance (CoNLL F1: 0.98, 0.98, 0.96). From the result, we infer that the MUC scores are near saturation (0.97–0.99) across all systems, whereas larger differences are observed in CEAFe.

Overall, the high and consistent scores across gold mentions, predicted mentions, and the large-scale setting suggest that software mention coreference in scholarly text is primarily driven by surface similarity. Performance decreases only slightly on the large-scale task, indicating that the proposed approaches generalize well to noisy, larger inputs.

6.2. Interpreting Differences across Metrics

The metric patterns indicate that most systems reliably recover the broad coreference structure. High MUC scores indicate that systems correctly establish coarse links between mentions. In contrast, CEAFe separates systems more clearly, reflecting differences in cluster precision and entity-level alignment. Since CEAFe penalizes both over-merging and fragmentation, lower scores suggest less precise cluster boundaries.

B^3 scores fall between MUC and CEAFe and generally reflect overall system quality. The identical ranking across all three subtasks indicates that clustering strategy is the main factor influencing performance, while sensitivity to extraction noise or corpus scale appears limited.

In summary, the main remaining challenges lie in improving cluster purity and entity-level alignment rather than in establishing basic cross-document links or handling scale.

6.3. Scalability Analysis

Subtask 3 was explicitly designed to challenge systems on whether they can maintain accuracy while handling a significant increase in computational load. Runtime measurements conducted on the same machine illustrate how differently systems respond to this challenge. The system of a winning participant (*aalkan*) requires approximately 19 seconds on Subtask 2 and approximately 2.2 hours on Subtask 3. In contrast, Baseline II (Matela and Krüger, 2026) completes Subtask 2 in approximately 8 seconds and Subtask 3 in approximately 3 minutes, representing a significant runtime reduction on the large-scale task.

These results highlight that blocking is an essential component for scalable cross-document coreference resolution,

7. Conclusion and Future Directions

SOMD 2026 introduces a shared evaluation setting for cross-document software mention coreference resolution and provides three subtasks spanning

System	Rep.	Method	Clustering
aalkan	FM (1) / CAR (2)	Lexical (1) / Embedding sim. (2)	Transitive closure(1) / Agglomerative(2)
mhassan	SciBERT (SupCon)	Supervised contrastive	HAC (avg.)
Baseline I	TF–IDF	Cosine similarity	DBSCAN
Baseline II	SBERT	FAISS + hybrid retrieval	HDBSCAN (+ blocking)

Table 2: Overview of submitted systems and baselines for SOMD 2026. **Rep.** denotes the mention representation model used. **Method** summarizes the core modeling strategy for computing mention similarity or entity assignment (e.g., retrieval, contrastive learning, or lexical similarity). **Clustering** indicates the algorithm used to group mentions into cross-document coreference clusters. FM = Fuzzy Matching; CAR = Context-Aware Representations; SupCon = Supervised Contrastive Loss; HAC = Hierarchical Agglomerative Clustering.

System	F1-MUC	F1-B^3	F1-CEAFE	F1-CoNLL
Subtask 1: Coreference resolution over gold standard mentions across multiple documents				
aalkan	0.98	0.96	0.93	**0.96**
mhassan	0.97	0.94	0.86	0.92
Baseline I	0.92	0.86	0.68	0.82
Baseline II	**0.99**	**0.99**	**0.96**	**0.98**
Subtask 2: Coreference resolution over predicted mentions across multiple documents				
aalkan	0.99	0.96	0.93	**0.96**
mhassan	0.98	0.93	0.85	0.92
Baseline I	0.94	0.87	0.72	0.84
Baseline II	**0.99**	**0.99**	**0.95**	**0.98**
Subtask 3: Coreference resolution over predicted mentions across multiple documents at scale				
aalkan	0.99	0.94	0.90	**0.94**
Baseline I	0.98	0.94	0.90	0.94
Baseline II	**0.99**	**0.97**	**0.92**	**0.96**

Table 3: Results for SOMD 2026 across three subtasks, reported as F1 scores for MUC, B^3, CEAFe, and their unweighted average (CoNLL F1). MUC measures link-level overlap, B^3 evaluates mention-level cluster overlap, and CEAFe scores entity-level alignment via one-to-one cluster matching. Two teams submitted systems: *aalkan* participated in all three subtasks, and *mhassan* participated in Subtasks 1 and 2 only. Baseline I is a TF–IDF + DBSCAN system; Baseline II is a hybrid Sentence-BERT + FAISS + HDBSCAN system. Bold indicates the best score per column within each subtask.

gold mentions, automatically extracted mentions, and scale-oriented sampling from SoftwareKG. The new annotations and the benchmark design support research on robust and scalable clustering methods, which are needed for knowledge graph construction and large-scale studies of software use in science.

Future work can extend the benchmark in several ways: (i) adding explicit links to external identifiers (e.g., registry ids or repository ids) to bridge clustering and entity linking, (ii) providing standard blocking candidates to compare scalable approaches more directly, (iii) expanding to more domains and languages, and (iv) evaluating end-to-end pipelines that combine detection, metadata extraction, and disambiguation.
Furthermore, we would like to emphasize on investigating blocking strategies that reduce the clustering search space without sacrificing recall as a future direction. As the shared task results demonstrate, search space reduction is the primary bottleneck to deploying coreference resolution systems at knowledge graph scale. While Baseline II demonstrates that partitioning mentions by entity type and canonical name initial yields substantial runtime gains. More systematic approaches such as learned blocking, approximate nearest-neighbor indexing, or ontology-guided partitioning remain largely unexplored for software mention disambiguation and represent a promising direction for future research.

8. Ethics Statement

This shared task uses software mentions from scholarly publications and builds on the SoMeSci

and SoftwareKG resources. The data consists of scientific articles and derived mention-level annotations for software entities and their cross-document coreference relations. The task does not involve private communication, patient records, or other directly sensitive personal data.

9. Limitations

This benchmark is limited to the entity type *software* and does not cover cross-document coreference for other scientific entity types. In addition, the data is grounded in SoMeSci and SoftwareKG, with the large-scale setting based on PubMed Central articles. The extent to which the reported results generalize across disciplines, publication cultures, and languages remains to be tested.

A second limitation is that the task is not fully end-to-end. The benchmark starts from gold or automatically extracted software mentions and evaluates clustering over these mentions, rather than jointly evaluating mention detection, metadata extraction, and cross-document disambiguation in a single pipeline.

A third limitation concerns evaluation coverage. In Subtask 2, evaluation is restricted to mentions that can be aligned to existing gold clusters or were manually annotated, and in Subtask 3, system scoring is performed on an annotated subset of the large-scale test data. Reported scores should therefore be interpreted as benchmark scores for the evaluated subsets rather than exhaustive quality estimates over the full mention pool.

A fourth limitation is the limited number of competitive submissions. Although five participants registered, only two teams submitted valid systems. This restricts how strongly the shared task results can be used to characterize the broader state of the field.

Finally, the strong results across subtasks suggest that many benchmark cases can be handled using surface similarity and lightweight metadata signals. Harder cases involving short ambiguous names, sparse metadata, and stronger domain shift may still be underrepresented. This is consistent with the paper's own future directions, which include expansion to more domains and languages and evaluation of end-to-end pipelines.

10. Acknowledgements

This paper was prepared within the NFDI4DS and the BERD@NFDI consortium in the context of the work of the National Research Data Infrastructure (NFDI) Association. NFDI is funded by the Federal Republic of Germany and the 16 federal states. The NFDI4DS consortium is supported within NFDI by the German Research Foundation (DFG) – NFDI 27/1-2026, project number 460234259. The BERD@NFDI consortium is supported within NFDI by the German Research Foundation (DFG) – NFDI 27/1-2026, project number 460037581. We thank both NFDI4DS and BERD@NFDI for their funding and support. Special thanks go to all institutions and individuals contributing to the association and its goals. Finally, we would like to thank Luke Friedrich for his contribution to the annotation of the software coreferences.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC'98), Workshop on Linguistic Coreference*, pages 563–566, Granada, Spain.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text.

CodaBench. 2026. Codabench. https://www.codabench.org/. Accessed: 2026-02-20.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231.

GESIS. 2026. SoftwareKG dataset portal. https://data.gesis.org/softwarekg/. Accessed: 2026-02-20.

Daniel S. Katz and Neil P. Chue Hong. 2024. Special issue on software citation, indexing, and discoverability. *PeerJ Computer Science*, 10:e1951.

Aliakbar Keshtkaran, Siti Sophiayati Yuhaniz, and Suhaimi Ibrahim. 2017. An overview of cross-document coreference resolution. In *2017 International Conference on Computer and Drone Applications (IConDA)*, pages 43–48. IEEE.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning.

Frank Krüger, Saurav Karmakar, and Stefan Dietze. 2024. SOMD@NSLP2024: Overview and insights from the software mention detection shared task. In *International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs*, pages 247–256. Springer.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Julia Matela and Frank Krüger. 2026. Semantic centroids and hierarchical density-based clustering for cross-document software coreference resolution. arXiv:2603.24246.

NFDI4DataScience. 2026a. NSLP 2026 workshop website. https://nfdi4ds.github.io/nslp2026/. Accessed: 2026-02-20.

NFDI4DataScience. 2026b. SOMD shared task 2026 (NSLP 2026). https://nfdi4ds.github.io/nslp2026/docs/somd_shared_task.html. Accessed: 2026-02-20.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

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 and Knowledge Management (CIKM 2021)*, pages 4574–4583.

David Schindler, Felix Bensmann, Stefan Dietze, and Frank Krüger. 2022. The role of software in science: a knowledge graph-based analysis of software mentions in pubmed central. *PeerJ Computer Science*, 8:e835.

David Schindler, Tazin Hossain, Sascha Spors, and Frank Krüger. 2023. A multilevel analysis of data quality for formal software citation. *Quantitative Science Studies*, 5:637–667.

Sharmila Upadhyaya, Wolfgang Otto, Frank Krüger, and Stefan Dietze. 2025. SOMD2025: A challenging shared tasks for software related information extraction. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 137–145, Vienna, Austria. Association for Computational Linguistics.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Zhen Xu, Cédric Ré, et al. 2021. Codabench: Flexible, easy-to-use and reproducible meta-benchmark platform. arXiv:2110.05802.

Towards Efficient Self-Explainable Climate-Related Claim Verification with Generative Models

Siting Liang, Omar Adjali, Daniel Sonntag
German Research Center for Artificial Intelligence, Oldenburg University
Germany
{siting.liang, omar.adjali, daniel.sonntag}@dfki.de
{siting.liang, daniel.sonntag}@uni-oldenburg.de

Abstract

In this work, we present an empirical investigation into two self-explanatory inference paradigms using pre-trained language models with different sizes, based on our participation in the **ClimateCheck@NSLP 2026** shared task on climate-related claim verification. This task aims to address the increasing amount of climate misinformation and disinformation on social media, emphasizing the importance of basing claims on reliable scientific evidence. Our study investigates the impact of different explanation strategies on entailment-based verification performance in scientific claim verification, while analyzing the trade-off between reasoning complexity and computational efficiency.

Keywords: Scientific Claim Verification, Large Language Model, Natural Language Inference

1. Introduction

Scientific misinformation through digital media has been posing significant societal risks and motivating the development of automated claim verification systems (Sarrouti et al., 2021; Wadden et al., 2020; Ahmad et al., 2025). The task of scientific claim verification involves assessing whether a scientific claim is supported, refuted, or lacks sufficient evidence when compared against scientific literature. In this work, we examine two self-explanatory inference approaches, exploring how explanation strategies impact performance and the trade-off between reasoning complexity and computational efficiency in the context of verifying climate-related claims from the **ClimateCheck@NSLP 2026 shared task** (Abu Ahmad et al., 2026).

Traditional approaches to automated claim verification have primarily focused on achieving high predictive accuracy, often treating the problem as a black-box natural language inference (NLI) task (Bowman et al., 2015). Subsequent work, such as e-SNLI (Camburu et al., 2018) extended the standard natural language inference (NLI) framework by incorporating human-annotated explanations alongside entailment labels, enabling the joint learning of classification and explanation generation. This line of research advances self-explainable inference by encouraging models not only to predict entailment relations but also to provide explicit justifications for their decisions. Building upon this direction, more studies have increasingly focused on generative models to develop further self-explainable inference methods to meet the growing demand not only for accurate predictions but also for transparent and interpretable reasoning that domain experts can validate and trust (Jullien et al., 2023; Liang and Sonntag, 2025b).

Another challenge in scientific claim verification lies in long-context encoding, as the length of abstracts can easily span thousands of tokens. This exceeds the context windows of many pre-trained language models and imposes constraints on the choice of applicable models. Recent advances in large language models, many of which are capable of encoding thousands to tens of thousands of tokens, present new opportunities for developing systems that can both verify claims and generate human-readable explanations for their decisions (Wei et al., 2022). However, challenges such as hallucination and sensitivity to intermediate reasoning strategies remain significant obstacles, particularly in high-stakes scientific verification settings where faithful and evidence-grounded reasoning is essential (Xia et al., 2024; Saxena et al., 2024).

Previous studies have demonstrated that structured prompting frameworks can improve the reasoning abilities in large language LLMs (Yu et al., 2023). For instance, Liang and Sonntag (2025a) demonstrated that carefully designed multi-step reasoning prompts can improve the performance of LLMs on complex claim verification tasks in biomedical domains. However, such methods are often computationally expensive and inefficient. Despite the increased computation and processing time, the observed performance gains are minimal, suggesting that such methods may not be a practical solution for real-world verification systems. In this work, we attempt to simplify the reasoning pipeline to improve efficiency for the climate-related claim verification task. We conduct experiments to compare two explanatory paradigms, aiming to develop a more practical yet interpretable verification framework.

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 255–260
12 May 2026.

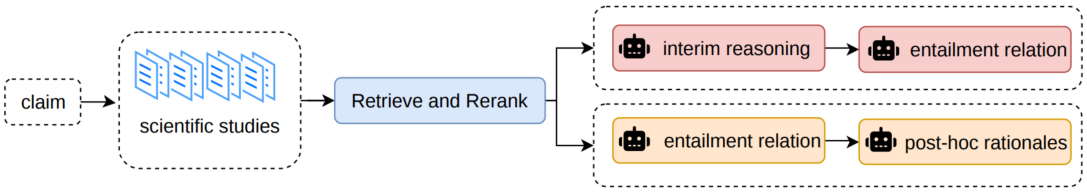


Figure 1: Overview of two-stage claim verification workflow with two explainability paradigms.

2. Approach

The climate-related claim verification workflow consists of two main stages: first, retrieving contextually relevant scientific abstracts from the corpus, and second, performing entailment-based verification between each claim-abstract pair. In the first stage, we follow the baseline procedure outlined in (Ahmad et al., 2025), using BM25[1] for initial retrieval, followed by a cross-encoder[2] to re-rank the results and identify contextually relevant scientific articles for each claim. For each claim–abstract pair, a pre-trained generative large language model is used to infer the entailment relationship ('Supports', 'Refutes', or 'Not Enough Information') using two alternative paradigms that are self-explanatory, as illustrated in Figure 1.

- **Intermediate Reasoning (IR),** where explanations are generated **prior to predicting** the entailment label, reflecting the model's step-by-step reasoning process.
- **Post-hoc Rationalization (PR),** where explanations are produced **after** the entailment label has been predicted, justifying the model's decision.

Our experiments are designed to address the following research questions:

1. How do different explainability paradigms impact the performance of large language models on complex reasoning tasks in scientific claim verification?
2. How does model size influence the effectiveness of each explainability paradigm in this context?

In our experiments, we evaluate a combination of lightweight open-source models and a low-cost commercial model to balance reasoning capability and efficient reproducibility, listed in Table 1.

GPT-4o-mini is a cost-efficient variant of the GPT-4 series that retains strong reasoning performance while being more practical for large-scale experimentation. In addition, we incorporate instruction-tuned open-source (Mistral-Nemo and Phi-3-Medium), which have demonstrated competitive performance on previous related work (Liang and Sonntag, 2025a). These mid-sized models provide a suitable balance between computational efficiency and reasoning capacity. We design separate prompting templates for the intermediate reasoning and post-hoc rationalization paradigms, as shown in Table 2.

In addition, we experiment with using GPT-4o-mini to generate augmented post-hoc rationalization samples based on the annotated entailment labels in the training set. These generated explanations are then used to fine-tune a much smaller generative model, such as T5-base[3] (Raffel et al., 2020), which is evaluated under both paradigms—prediction-first and prediction-after-explanation. This lightweight model requires less than 20 minutes to process the 1760 claim–abstract pairs at inference time, substantially reducing computational cost compared to large language models, which require over 200 minutes under the same setting.

Type	Parameters	Context Window
GPT-4o-mini (Achiam et al., 2023)	–	128K
Phi-3-Medium (Abdin et al., 2024)	∼14B	4K
Mistral-Nemo (Jiang et al., 2023)	12B	8K

Table 1: Comparison of the LLMs used in our experiments. We include both proprietary and open-source models to analyze the impact of model architecture and size on entailment-based claim verification under different explanatory paradigms.

Paradigm	Instruction Prompt
IR	*Given the Claim: '{claim}'* *and the abstract: '{abstract}'* *Summarize in short, is there enough information to determine if the claim aligns with the data points or if the data refutes the claim?* *→ Predict the 'logical relation' between the claim and the provided data. Select one label from ['not enough information', 'supports', 'refutes']. Return only valid JSON.*
PR	*Given the Claim: '{claim}'* *and the abstract: '{abstract}'* *Predict the 'logical relation' between the claim and the provided data. Select one label from ['not enough information', 'supports', 'refutes'].* *→ Explain your decision with evidence. Summarize in short.*

Table 2: Instruction prompts for the two explanatory paradigms with LLMs used in entailment-based claim verification.

[1] https://pypi.org/project/rank-bm25/

[2] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2

[3] https://huggingface.co/google-t5/t5-base

3. Results

In this section, we present and discuss the results obtained through submissions to the official leaderboard of the shared task. The evaluation scores are organized according to the two subtasks. Task 1.1 (Abstract Retrieval) is assessed using Recall@K (for K = 2 and 5) and B-Pref, measuring the effectiveness of retrieving relevant scientific abstracts for each claim. Task 1.2 (Claim Verification) is evaluated using weighted F1 scores, which reflect performance across all entailment classes (Supported, Refuted, Not Enough Information).

Metric	Recall@2	Recall@5	B-pref	Overall Score
Score	0.1931	0.3524	0.4010	0.3767

Table 3: Retrieval performance metrics using BM25 with sentence transformer-based reranking on Task1-1. Recall@K measures the proportion of relevant documents in the top-K results. B-pref (Binary Preference) evaluates ranking quality based on pairwise comparisons. Overall Score represents the average performance across all metrics.

These scores presented in Table 3 suggest that the current retrieval pipeline (BM25 + cross-encoder reranking) is not retrieving enough relevant scientific abstracts. Even in the top 5, only about 35% of relevant abstracts are retrieved. The B-Pref of about 0.4 also indicates the approach often ranks non-relevant abstracts above relevant ones. These low retrieval scores are likely to negatively impact downstream entailment-based claim verification, as they lack sufficient evidence to make accurate predictions.

In this work, however, we primarily focus on investigating the effectiveness of the explanatory paradigms in Task 1.2 (claim verification).

	Intermediate Reasoning				**Post-hoc Rationalization**			
Model	Prec.	Rec.	F1	Score	Prec.	Rec.	F1	Score
GPT-4o-mini	0.68	0.65	0.63	0.98	0.71	0.65	0.62	0.97
Phi-3-medium	0.67	0.64	0.62	0.97	0.63	0.50	0.41	0.76
Mistral-Nemo	0.67	0.65	0.64	0.98	0.63	0.50	0.41	0.76

Table 4: Performance scores of task 1.2 claim verification under intermediate reasoning and post-hoc rationalization paradigms in a zero-shot setting.

We compare the performance of three models under the intermediate reasoning and post-hoc rationalization paradigms on entailment-based claim verification (Task 1.2) in Table 4. Overall, the intermediate reasoning paradigm demonstrates more stable and consistent better performance across models in the zero-shot setting. GPT-4o-mini achieves the best results, indicating that larger, more capable models can better handle reasoning-generation tasks with long text input. Phi-3-medium and Mistral-Nemo have close model sizes and perform comparably to each other. However, under the post-hoc rationalization paradigm, both smaller models experience a sharp decline in recall and F1 scores. This contrast highlights the relative effectiveness of intermediate reasoning, suggesting that generating explanations prior to prediction can better support structured and evidence-grounded decision making. Although intermediate reasoning may introduce potential risks such as hallucination due to extended generation steps, the observed improvements in verification performance indicate that the trade-off between enhanced reasoning capability and generation risk is worthwhile in this context.

Paradigm	**Precision**	**Recall**	**F1**	**Score**
IR	0.49	0.49	0.49	0.84
PR	0.59	0.56	0.54	0.89

Table 5: Task 1.2 performance of T5-base trained with GPT-4o-mini augmented explanations under two explanatory paradigms.

T5-base requires significantly less computation than larger LLMs, making it an efficient and practical choice for large-scale claim verification. However, small pre-trained language models like T5-base typically cannot generate meaningful explanations independently, in contrast to LLMs. To address this, we use GPT-4o-mini to augment the training set with post-hoc rationalization examples conditioned on annotated entailment labels. Table 5 shows the performance of T5-base trained with the GPT-4o-mini–augmented post-hoc rationalizations under the two explanatory paradigms. While intermediate reasoning tends to be more stable for LLMs in a zero-shot setting, post-hoc rationalization outperforms intermediate reasoning for T5-base across all metrics. Fine-tuning on augmented post-hoc rationalizations appears particularly beneficial for smaller generative models, improving prediction quality when explanations are produced after the model makes its initial entailment decision.

Examples of generated explanations across models and explanatory paradigms are presented in Table 6. Overall, the scores indicate that model size remains the primary factor influencing reasoning capability in LLMs. Larger models consistently perform better across both paradigms, demonstrating greater robustness in entailment-based verification and the ability to generate coherent explanations. Nonetheless, the trade-off between performance, computational efficiency, and reproducibility remains an important consideration.

Model (Type)	Intermediate Reasoning	Post-hoc Rationalization
GPT-4o-mini (Zero-shot)	`The claim distinguishes short-term weather events from long-term climate trends. The abstract explains that the polar vortex drives winter variability and interacts with patterns like the North Atlantic Oscillation and El Niño-Southern Oscillation. Although the vortex can cause cold conditions, this does not contradict long-term climate change. | Relation: supports`	`supports. The abstract describes the polar vortex as a driver of winter variability, aligning with the claim that cold weather does not negate climate change. Short-term anomalies should not overshadow long-term climate trends.`
Phi-3 Medium (Zero-shot)	`The abstract presents the polar vortex as a natural weather pattern and climate change as a long-term trend. While vortex events influence surface temperatures, they are not directly linked to climate change. Short-term cold events should not be confused with long-term climate shifts. | Logical_relation: supports`	`supports. The abstract explains how the polar vortex affects winter circulation and temperature anomalies. Cold signatures may reflect natural variability rather than contradicting climate change, thus supporting the claim.`
Mistral-Nemo (Zero-shot)	`The claim aligns with the data. The abstract confirms that the polar vortex is a natural weather pattern and that climate change is a long-term trend, separate from short-term weather events. It also notes that the polar vortex can influence winter weather, but its effects are diverse and not always associated with large-scale cold conditions. Therefore, a cold winter (like those caused by the polar vortex) does not refute the reality of long-term climate change. | Relation: supports`	`supports. Reason: Weak vortex events cause cold-air outbreaks as part of natural variability. These do not contradict long-term warming trends. Summary: Cold events are natural variability and do not refute climate change.`
T5-base (Fine-tuned)	`The abstract identifies the polar vortex as a key driver of winter variability and associates weak and strong vortex winters with climate anomalies. This supports the claim that cold conditions do not negate long-term climate change. | Relation: Supports`	`Relation: Supports | Rationales: The polar vortex drives winter variability and climate anomalies. Cold temperatures therefore, reflect natural variability rather than contradicting long-term climate change.`

Table 6: Zero-shot prompting examples (GPT-4o-mini, Phi-3 Medium, Mistral-Nemo) are shown in the first three rows. Fine-tuned T5-base is shown in the last row. Left column presents intermediate reasoning or explanation→prediction; right column presents post-hoc rationalization or prediction→explanation.

4. Related Work

Large language models (LLMs) have demonstrated remarkable capabilities in reasoning and complex inference tasks due to their massive scale and extensive pre-training on diverse corpora. In particular, Chain-of-Thought (CoT) prompting (Wei et al., 2022) enables models to generate step-by-step intermediate reasoning, improving performance on tasks that require multi-step deduction. Zero-shot CoT prompting (Kojima et al., 2022), using simple instructions such as *Let's think step by step*, further shows that LLMs can perform structured reasoning even without explicit exemplars. However, performance can vary with task complexity and reasoning type (Huang and Chang, 2023). Prior work

in biomedical claim verification demonstrates the potential of self-explainable inference, where models jointly produce entailment predictions and supporting evidence (Jullien et al., 2023). Liang and Sonntag (2025a) showed that multi-step reasoning prompts can improve LLM performance on complex biomedical claim verification tasks, but these methods are often computationally expensive and yield only minimal gains. Drawing inspiration from biomedical and general scientific claim verification, we adapt these LLM reasoning and retrieval strategies to the climate domain.

5. Conclusion

Overall, we explore climate-related claim verification using LLMs of different sizes under two self-explainable paradigms: intermediate reasoning and post-hoc rationalization. We have conducted experiments comparing the performance of the models, and T5-base fine-tuned with augmented post-hoc rationalizations. Several key findings emerge from the experimental results: model size remains the primary factor influencing reasoning capability and verification performance. Meanwhile, smaller models like T5-base may benefit from fine-tuning with augmented explanations, and post-hoc rationalization provides the greatest gains for these models. There is a clear trade-off between performance, computational efficiency, and reproducibility. Future work will focus on fine-tuning lightweight LLMs, such as Phi-3 and Mistral, to further investigate the efficiency and effectiveness of self-explainable claim verification systems.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF) under grant numbers 01IW24006 (NoIDLEChatGPT), as well as by the Endowed Chair of Applied AI at the University of Oldenburg. We also appreciate the support of a grant from Accenture Labs. We also gratefully acknowledge the support provided by a grant from Accenture Labs[4].

Bibliographical References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Raia Abu Ahmad, Max Upravitelev, Aida Usmanova, Veronika Solopova, and Georg Rehm. 2026. ClimateCheck 2026: Scientific Fact-Checking and Disinformation Narrative Classification of Climate-related Claims. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025. The climatecheck dataset: Mapping social media claims about climate change to corresponding scholarly articles. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 42–56.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

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

Maël Jullien, Marco Valentino, Hannah Frost, Paul O'regan, Donal Landers, and André Freitas. 2023. SemEval-2023 task 7: Multi-evidence natural language inference for clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages

[4]https://iml.dfki.de/news/autoprompt-aims-to-improve-chatgpts-analysis-of-clinical-data/

2216–2226, Toronto, Canada. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 22199–22213.

Siting Liang and Daniel Sonntag. 2025a. Advancing biomedical claim verification by using large language models with better structured prompting strategies. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, pages 148–166, Viena, Austria. Association for Computational Linguistics.

Siting Liang and Daniel Sonntag. 2025b. Explainable Biomedical Claim Verification with Large Language Models. In *Joint Proceedings of the ACM IUI Workshops 2025*, volume 3957 of *CEUR Workshop Proceedings*, Cagliari, Italy. CEUR-WS.org. Co-located with the 30th Annual ACM Conference on Intelligent User Interfaces (IUI 2025).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yash Saxena, Sarthak Chopra, and Arunendra Mani Tripathi. 2024. Evaluating consistency and reasoning capabilities of large language models.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yu Xia, Rui Wang, Xu Liu, Mingyan Li, Tong Yu, Xiang Chen, Julian McAuley, and Shuai Li. 2024. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms.

Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. 2023. Towards better chain-of-thought prompting strategies: A survey.

Transferring Scientific English Pre-Trained Language Models to Multiple Languages Using Cross-Lingual Transfer

Nikolas Rauscher[1,2], Fabio Barth[2], Georg Rehm[2,3]
[1]Technische Universität Berlin, Germany
[2]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
[3]Humboldt-Universität zu Berlin, Germany
nikolas.rauscher@dfki.de, fabio.barth@dfki.de, georg.rehm@dfki.de

Abstract

In this paper, we present a pipeline for domain-adaptive pre-training and cross-lingual transfer of scientific language models from English to non-English languages. Starting from the multilingual scientific corpus SciLaD, we construct a cleaned English pre-training split and continually pre-train a T5-base encoder–decoder model, resulting in EN-T5-Sci. Our model achieves consistent zero-shot improvements on the Global-MMLU English benchmark, outperforming its base model, with particularly strong gains in STEM and Social Sciences. Despite its moderate size, it performs comparably to the much larger BLOOM model on scientific categories. Building on EN-T5-Sci, we transfer scientific knowledge to German, Japanese, Russian, Polish, Spanish, and Portuguese using the WECHSEL method. Our approach reinitializes language-specific embedding layers via aligned static embeddings while retaining the pre-trained Transformer weights, yielding six monolingual scientific T5 models. In zero-shot evaluation in each respective language, the transferred models generally outperform monolingual baselines. These results demonstrate that scientific domain knowledge acquired through English pre-training can be effectively transferred across languages, enabling competitive non-English scientific language models without training large multilingual systems from scratch.

Keywords: T5, scientific language models, pre-training, cross-lingual transfer, multilingual scientific NLP

1. Introduction

In recent years, large language models have shown remarkable improvements in scientific English language processing (Hu et al., 2025). Many of these language models are pre-trained on scientific text, which provides a complementary data source to generic web pages, as it is factually dense and stylistically consistent. They follow a standardized structure and undergo peer review, which ensures greater validity. Prior work has shown that adapting models to scientific literature improves performance on domain-specific tasks, even when using significantly less data than broad web scrapes (Phan et al., 2021; Taylor et al., 2022). However, the availability of high-quality scientific text is unevenly distributed, with English being the predominant language (Tardy, 2004). Even high-resource languages, such as German or Japanese, can be considered low-resource languages in the scientific domain compared to English. As a result, scientific language models have been English-centric, and multilingual language models are trained on noisier non-scientific data or machine-translated content (Phan et al., 2021; Taylor et al., 2022; Ali et al., 2025).

An approach for generating high-quality language models for low-resource languages is domain-adaptive pre-training (DAPT) with cross-lingual transfer (Minixhofer et al., 2022). Here, a language model is trained on a high-resource language using a large corpus and then efficiently and effectively transferred into a low-resource language using cross-lingual parameter transfer. Although this process has been a well-established method for generating low-resource language models in the general domain, this method has not been applied to the scientific domain.

We propose a pipeline for DAPT with cross-lingual transfer from a pre-trained English scientific language model to generate non-English scientific language models. We therefore continually pre-train an English T5-base model on a newly composed scientific corpus (Foppiano et al., 2025) and then transfer the acquired knowledge to six non-English languages by applying the well-established WECHSEL method to the scientific domain (Minixhofer et al., 2022). We present monolingual scientific language models in German, Japanese, Russian, Polish, Spanish, and Portuguese. Our transferred scientific language models outperform their baselines and achieve results comparable to those of a much larger multilingual LLM trained on general-domain and scientific data. Our contributions can be summarized as follows:

1. We create a pipeline for scientific knowledge transfer from English to non-English languages.
2. We train a scientific English T5-base model (EN-T5-Sci), which achieves consistent

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 261–269
12 May 2026.

zero-shot gains on Global-MMLU benchmarks, especially in science-related categories.

3. We present six monolingual scientific T5 models in German, Japanese, Russian, Polish, Spanish, and Portuguese by transferring EN-T5-Sci using WECHSEL. The transferred models outperform almost all general-domain monolingual models in their language and achieve strong performance on scientific question answering.

The rest of the paper is structured as follows: We first describe the pre-training dataset used for the continual pre-training of the English T5-base model and the language-specific data used for tokenizer training and cross-lingual transfer. We then describe the models and training setup, followed by the experimental evaluation of all models. Finally, we discuss related work and conclude the paper.

2. Dataset Construction & Pre-processing

The process of transferring a model from one language to another requires two data splits. A very large split in the initial language (English) for pre-training the language model, and a much smaller dataset in the target language to train the non-English tokenizer. As English has been the established lingua franca in the scientific domain, most scientific pre-training datasets already have a linguistic bias towards English. Many scientific datasets are published with English data only, or the text is not classified by language. For our approach, we use the recently published SciLaD dataset (Foppiano et al., 2025), which contains open-access articles from Unpaywall processed into plain text via GROBID. The corpus contains over 200 classified languages. The English split consists of approximately 11 million documents, accounting for 73.95% of the dataset. In contrast, the German split is only 1.99%, highlighting the language bias in scientific texts. The largest language split after the English split in the SciLaD dataset is Japanese, with 4.1%, followed by Portuguese, with 3.5%. Less than 1% of the data contains Polish-labeled text.

We build a cleaning pipeline using Hugging Face's DataTrove[1] library for the pre-training of the scientific English T5 model. Our cleaning pipeline removes non-semantic noise embedded during the PDF-to-text conversion. We remove approximately 184 million citation artifacts using regular expression (regex) filters. These include numeric brackets (e.g., "[1–3]"), parenthetical author-year patterns, and semicolon-separated author lists. We also delete artifact identifiers such as DOIs, ISBNs, and arXiv IDs. URLs and email addresses are replaced with placeholders.

Metric	Before	After	Change
Documents	10,030,761	10,027,111	-0.04%
Mean length (chars)	28,627	22,319	-22.0%
Mean perplexity	569.54	538.85	-5.4%
EN confidence	93.36%	93.75%	+0.4%
Paragraph duplicates	1.30%	0.75%	-42.3%
Digit ratio	1.56%	1.20%	-23.4%

Table 1: Impact of the DataTrove cleaning pipeline on the English SciLaD pre-training split, comparing corpus statistics before and after cleaning.

Through heuristic filtering, we also remove redundant text and tabular fragments. First, we delete structural noise, such as figure/table captions and repeated header metadata. We then remove tabular fragments if their numeric share exceeds 60%. Finally, we normalize the text by collapsing whitespace, standardizing ellipses to ASCII, and converting bullets to Markdown format. After processing, the average document length decreased by 22.0% while preserving 99.9% of the documents. The Wikipedia-LM perplexity improved by 5.4% and paragraph duplicates fell by 42.3% (see Table 1).

3. Model Training

In this section, we first describe the pre-training of the scientific English T5 model using the SciLaD dataset (Foppiano et al., 2025) and then outline our pipeline for scientific knowledge transfer from English to non-English languages.

3.1. Scientific English T5 Model

We perform continued pre-training on the clean English SciLaD data using the T5-base[2] Encoder–Decoder architecture (220M parameters), initialized from the original Google checkpoint. Following the approach of Raffel et al. (2023), we use span masked model training with a 15% masking rate and a mean noise span length of 3 tokens. We choose the T5 model over more recent generative decoder-only models to keep the computational cost for all monolingual language models reasonable. Note that, in this paper, we showcase the performance of knowledge transfer into non-English languages rather than building comparable language models for text generation, which is why

[1] https://github.com/huggingface/datatrove *accessed 2026-02-19*

[2] https://huggingface.co/google-t5/t5-base *accessed 2026-02-19*

we choose a smaller T5 model over an LLM with billions of parameters.

Sequence Preparation For the pre-training, we segment each document into fixed-length sequences of 512 tokens using a sliding-window strategy with 50% overlap to process long-form articles. This yields approximately 261M training datapoints, resulting in a span-corruption setup with an average decoder target length of about 114 tokens for 512-token inputs.

Optimization and Schedule We train on 4× NVIDIA H100 (80GB) for 487,500 steps (within the first epoch) with an effective batch size of 384 (per-GPU batch size 48 and gradient accumulation of 2). The learning-rate schedule uses a linear warmup for 20,000 steps to a peak of 10^{-3}, followed by an inverse-square-root decay. We apply gradient clipping of 0.5 and use the Adafactor (Shazeer and Stern, 2018) optimizer as the optimization method.

Monitoring and Checkpoint Selection We compute the validation loss and perplexity every 5,000 steps on a held-out validation split of 100,000 datapoints. We select the checkpoints with the minimal validation loss, referred to as EN-T5-Sci.

3.2. Cross-Lingual Transfer using WECHSEL

To transfer the EN-T5-Sci model to any non-English split, we build a pipeline that follows the WECHSEL approach (Minixhofer et al., 2022). The basic assumption in this approach is that Transformer layers primarily encode abstract, language-agnostic knowledge, while language-specific information is concentrated in the token embeddings. Accordingly, our pipeline transfers all pre-trained non-embedding weights from the EN-T5-Sci checkpoints to a non-English target model and reinitializes the embedding matrix for a non-English tokenizer. After the embedding matrix is re-initialized, the non-English tokenizer is trained on a language-specific scientific vocabulary. We run the pipeline for German, Japanese, Russian, Polish, Spanish, and Portuguese, resulting in six monolingual non-English scientific language models.

Tokenizer Training For each language, we first sample a 5 GB subset from the corresponding SciLaD language split and divide it into train, validation, and test splits. We then train a SentencePiece tokenizer on the train split to obtain a vocabulary that reflects the language- and domain-specific subword statistics. This is important because we replace the source embedding layer, and reusing an English tokenizer would likely yield suboptimal segmentation for non-English scientific text and a target embedding vocabulary that does not match the target distribution well.

During tokenizer training, we normalize line breaks to spaces and cap each document at 20k characters to limit outlier effects. We use SentencePiece BPE with a vocabulary size of 32k subwords, character coverage 1.0, and byte-fallback enabled. To ensure T5 compatibility, we fix special token IDs (`pad=0, eos=1, unk=2; bos=disabled`) and export the tokenizer with `extra_ids=100`, resulting in 32,100 tokens in total. The resulting tokenizer defines the target subword vocabulary U^t used in WECHSEL.

Embedding Initialization and Alignment To initialize the embedding matrix, we align English and non-English tokens in a shared static semantic space. Following the work of Minixhofer et al. (2022), we use multilingual fastText vectors and a bilingual dictionary to obtain aligned static subword embeddings for source and target vocabularies. Let U^s and U^t denote the source and target subword vocabularies, and let E^s and E^t denote the corresponding source and target model embedding matrices. For each target subword $x \in U^t$, we retrieve its k nearest source neighbors $y \in U^s$ by cosine similarity in the shared space,

$$s_{x,y} = \frac{u_x^t u_y^{s\top}}{\|u_x^t\| \, \|u_y^s\|},$$

and initialize the target embedding as a convex combination of the corresponding source embeddings, with mixture weights given by a softmax over similarities with temperature τ (k=10, τ=0.1). Concretely, letting $\mathcal{J}_x$ be the set of the k nearest source subwords for x,

$$e_x^t = \frac{\sum_{y \in \mathcal{J}_x} \exp(s_{x,y}/\tau) \cdot e_y^s}{\sum_{y' \in \mathcal{J}_x} \exp(s_{x,y'}/\tau)}.$$

4. Experimental Setup

As the number of publicly available benchmarks for the scientific domain is limited, we evaluate the models on a subset of Global-MMLU (Singh et al., 2025). Global-MMLU is a translation of MMLU (Hendrycks et al., 2021), a multiple-choice QA dataset spanning diverse subject areas, including elementary mathematics, computer science, and law. We evaluate each language model in its respective language. For comparability, we evaluate all models in a zero-shot setting using the EleutherAI `lm-evaluation-harness`[3], en-

[3] https://github.com/EleutherAI/lm-evaluation-harness *accessed 2026-02-19*

Language	Abbrev.	Base model checkpoint (Hugging Face)
German	DE-Base	GermanT5/t5-efficient-gc4-german-base-nl36
Japanese	JA-Base	megagonlabs/t5-base-japanese-web
Spanish	ES-Base	vgaraujov/t5-base-spanish
Polish	PL-Base	allegro/plt5-base
Portuguese	PT-Base	unicamp-dl/ptt5-base-portuguese-vocab
Russian	RU-Base	ai-forever/ruT5-base

Table 2: Non-English monolingual base checkpoints and abbreviations used in the experiments.

suring the prompt template and scoring configuration are fixed across all models. As a metric, we use accuracy using multiple-choice log-likelihood scoring. In multiple-choice log-likelihood scoring, the option with the highest log-probability is selected as the prediction. We use fixed seeds for all evaluations and report overall scores, as well as subject-level changes in Global-MMLU, to characterize domain gains and potential trade-offs.

As a baseline for the scientifically continued-pretrained English T5 model (EN-T5-Sci), we evaluate the original Google T5-base checkpoint (EN-T5-Base) and the SciFive model, a T5 model trained on biomedical literature, on the same setup. We additionally include BLOOM-176B, a multilingual decoder-only LLM trained on general and scientific data. Because BLOOM is architecturally different (decoder-only vs. encoder-decoder) and substantially larger, it is not a controlled baseline comparison. Therefore, it serves as a multilingual reference point. For the non-English experiments, we use one monolingual T5 checkpoint per language as the corresponding base model (Table 2). These checkpoints define the DE-Base, JA-Base, ES-Base, PL-Base, PT-Base, and RU-Base in the experiments.

For each target language, we also report BLOOM results on Global-MMLU as an additional multilingual reference. To establish domain-adapted monolingual baselines, we continuously pre-train the non-English base models for 15,000 steps on the respective 5GB SciLaD subsplit (DE-Base-CP, JA-Base-CP, RU-Base-CP, PL-Base-CP, ES-Base-CP, and PT-Base-CP). For this adaptation phase, we use the same sequence preparation as in English continued pre-training, i.e., 512-token sequences with a sliding-window strategy and 50% overlap. We also use the same span-corruption objective and peak learning rate of 10^{-3}. We reduce warmup to 1,500 steps and apply gradient clipping with a max norm of 1.0. We compare these controls against our WECHSEL-initialized models (DE-Trans-Init, JA-Trans-Init, RU-Trans-Init, PL-Trans-Init, ES-Trans-Init, and PT-Trans-Init). This setup enables a controlled comparison between standard monolingual adaptation and our initialization-based transfer approach.

5. Evaluation Results

5.1. English Language Model

The EN-T5-Sci model achieves a higher score on the English MMLU split compared to its base model, with an average accuracy score increase of 4 pp (see Table 3). The largest category-level gain is in Social Sciences (+9.4 pp), followed by STEM (+7.2 pp). Compared to the SciFive model, our model also performs on average 4 pp better. However, compared to the BLOOM model, the EN-T5-Sci only outperforms it in the Social Sciences category with a 0.9 pp higher score, while having an average 1.7 pp lower score. Note that the BLOOM model has 176 billion parameters, 800 times as many as the T5 model. When comparing only the scientific categories (STEM, Humanities, and Social Sciences), the EN-T5-Sci is only 0.6 pp worse than the BLOOM model.

Model	Avg	STEM	Hum	SocSci	Other
English Monolingual Models					
SciFive-Base	22.9	21.3	24.2	21.7	24.0
EN-T5-Base	22.9	21.3	24.1	21.7	23.9
EN-T5-Sci	26.9	28.5	24.2	**31.1**	25.1
Multilingual Model					
BLOOM-176B	**28.6**	**29.8**	**25.6**	30.2	**30.1**
Δ (EN-T5-Sci - EN-T5-Base)	+4.0	+7.2	+0.1	+9.4	+1.2

Table 3: Zero-shot accuracy (%) on the English **Global-MMLU-EN**, for SciFive-Base, EN-T5-Base, EN-T5-Sci, and BLOOM-176B, reported as overall and category-level results.

5.2. Non-English Language Models

The best-performing non-English transferred models are DE-Trans-Init and ES-Trans-Init, both with 26.89% average accuracy. Both are numerically very close to the English source model EN-T5-Sci (26.87%). For German, STEM and Social Sciences show the largest gains, and the control model (DE-Base-CP) shows no improvement over DE-Base. This indicates that the German gains mainly come from the cross-lingual transfer, not from short continued pre-training alone. DE-Trans-Init is only 0.34 pp worse than BLOOM on Global-MMLU-DE, while scoring 0.79 pp higher on STEM.

Model	Avg	STEM	Hum	SocSci	Other
English Source Model on Global-MMLU-EN					
EN-T5-Sci	26.87	28.51	24.19	**31.07**	25.10
BLOOM-176B	**28.60**	**29.80**	**25.60**	30.20	**30.10**
German models on Global-MMLU-DE					
DE-Base	22.95	21.25	24.21	21.71	23.98
DE-Base-CP	22.95	21.25	24.21	21.71	23.98
DE-Trans-Init	26.89	**28.64**	24.14	**31.07**	25.14
BLOOM-176B	**27.23**	27.85	**25.27**	30.19	**26.65**
Japanese models on Global-MMLU-JA					
JA-Base	25.23	24.58	23.89	**28.57**	24.62
JA-Base-CP	22.95	21.25	24.23	21.71	23.98
JA-Trans-Init	25.51	**26.26**	**27.12**	23.79	24.01
BLOOM-176B	**26.04**	25.44	26.67	25.64	**26.07**
Russian models on Global-MMLU-RU					
RU-Base	26.01	28.35	24.55	28.18	23.72
RU-Base-CP	24.17	23.25	**25.36**	23.24	24.24
RU-Trans-Init	26.36	27.12	24.89	28.86	25.33
BLOOM-176B	**28.04**	**30.26**	25.31	**30.61**	**27.36**
Polish models on Global-MMLU-PL					
PL-Base	25.51	26.26	**27.12**	23.79	24.01
PL-Base-CP	24.65	23.88	24.51	23.43	**26.87**
PL-Trans-Init	24.66	23.91	24.51	23.43	**26.87**
BLOOM-176B	**27.17**	**27.40**	26.55	**29.54**	25.52
Spanish models on Global-MMLU-ES					
ES-Base	25.51	26.26	**27.12**	23.79	24.01
ES-Base-CP	25.51	26.26	**27.12**	23.79	24.01
ES-Trans-Init	26.89	28.61	24.17	31.07	25.14
BLOOM-176B	**29.23**	**30.29**	26.57	**31.59**	**29.84**
Portuguese models on Global-MMLU-PT					
PT-Base	23.55	22.58	23.91	23.14	24.40
PT-Base-CP	23.02	21.54	24.08	21.84	24.07
PT-Trans-Init	24.98	24.64	24.44	24.34	26.75
BLOOM-176B	**29.17**	**29.50**	**26.40**	**32.27**	**29.96**

Table 4: Zero-shot accuracy (%) on Global-MMLU benchmarks, grouped by evaluation language, for baseline, continued-pretrained, transfer-initialized, and BLOOM-176B models across all evaluated languages.

Except for the transferred Polish model, all other transferred non-English models improve over their corresponding baselines. For JA-Trans-Init, RU-Trans-Init, and PT-Trans-Init, the mean average gain is 0.69 pp. In contrast, PL-Trans-Init is 0.85 pp below PL-Base on average (24.66% vs. 25.51%), although it attains the strongest Polish *Other* score (26.87%, tied with PL-Base-CP and above BLOOM's 25.52%). JA-Trans-Init outperforms BLOOM in STEM and Humanities. PT-Trans-Init is the only transferred model that outperforms its base model across all categories (+1.43 pp on average), but it remains 4.19 pp below BLOOM in overall average.

Overall, the results show that the transferred language-specific models improve over the respective baselines across most languages. We even observe category-level gains over the much larger BLOOM model, for example, in German STEM and Social Sciences, as well as Japanese STEM and Humanities, although BLOOM remains stronger on most overall averages.

5.3. Error Analysis

For the error analysis, we discuss in more detail the results of the English model and the German model as the best-performing non-English mod-

els. For both models, we analyze the best- and worst-performing categories in the MMLU evaluation in more detail. We highlight the largest accuracy gains and the categories in which the models fell short after pre-training or transfer, respectively.

English Model Table 5 lists the top- and bottom-performing subtasks for the English models. The largest performance gains of the English model are in subjects such as *high_school_statistics*, *professional_medicine*, and *college_chemistry*, underlining improved performance on science-related tasks. We see gains of over 30 pp in *high_school_statistics*, for instance, which shows the impact of specialization.

Subtask	EN-Base	EN-T5-Sci	SciFive	BLOOM
Top 4 (EN-T5-Sci)				
high_school_statistics	15.28	**47.22**	15.28	15.74
professional_medicine	18.38	**44.85**	18.38	18.75
college_chemistry	20.00	**41.00**	20.00	20.00
security_studies	18.78	**40.00**	18.78	18.78
Bottom 4 (EN-T5-Sci)				
human_aging	30.94	10.76	**31.39**	**31.39**
machine_learning	**31.25**	16.07	**31.25**	**31.25**
international_law	**23.97**	14.05	**23.97**	**23.97**
world_religions	**32.16**	17.54	**32.16**	31.58

Table 5: Zero-shot accuracy (%) on the top and bottom 4 **Global-MMLU-EN** subtasks, sorted by EN-T5-Sci performance. Scientific pre-training yields strong gains on STEM-oriented subtasks but regressions on unrelated subjects.

An interesting observation is the worst-performing subjects of the scientific T5 model. Here, the model performs worse on non-scientific tasks such as *human_aging*, *international_law*, and *world_religions*, with average scores 14.9 pp lower on those subjects. Moreover, the model performs worse on a science-related subject like *machine_learning*, dropping by 15.2 pp. One possible reason for this drop could be the structure of the *machine_learning* QA-pairs: most of them contain short answers with true/false options, which likely prompt the model to generate a broad probability distribution over the choices, since they seem more similar to the model.

Compared to BLOOM and SciFive, EN-T5-Sci also outperforms those models in the scientific categories (see Table 5). BLOOM and SciFive achieve similar scores to the base model in categories such as *human_aging*, *international_law*, and *world_religions*. These results highlight the transfer of scientific knowledge from the pre-training dataset.

German Model Similar to the scientific English model, the transferred German model has its strongest category-level results in STEM and Social Sciences. At the subtask level, the highest DE-Trans-Init accuracies are in *high_school_statistics*, *professional_medicine*, *college_chemistry*, and *security_studies*.

The lowest DE-Trans-Init accuracies are in *human_aging*, *international_law*, *machine_learning*, and *world_religions*. This pattern indicates that gains are uneven across subjects and include trade-offs in some tasks despite overall improvements in selected categories. One possible explanation is a specialization trade-off or catastrophic forgetting, in which the model emphasizes certain scientific patterns at the expense of broader knowledge (Haque, 2025). Since WECHSEL largely preserves the trained Transformer weights and mainly replaces and aligns the embedding layer, the transferred model keeps part of the source behavior while still exhibiting language-specific weaknesses.

Subtask	DE-Base	DE-Base-CP	DE-Trans-Init	BLOOM
Top 4 (DE-Trans-Init)				
high_school_statistics	15.28	15.28	**47.22**	37.50
professional_medicine	18.38	18.38	**44.85**	43.75
college_chemistry	20.00	20.00	**41.00**	31.00
security_studies	18.78	18.78	**40.00**	36.73
Bottom 4 (DE-Trans-Init)				
human_aging	**31.39**	**31.39**	10.76	15.25
international_law	**23.97**	**23.97**	14.05	19.01
machine_learning	**31.25**	**31.25**	16.07	25.00
world_religions	**32.16**	**32.16**	17.54	23.98

Table 6: Zero-shot accuracy (%) on the top and bottom 4 **Global-MMLU-DE** subtasks, sorted by DE-Trans-Init performance. Continued pre-training alone shows little effect, while transfer improves top subtasks similar to the source model.

Notably, the short continued pre-training phase (*DE-Base-CP*) does not show measurable improvements over *DE-Base* on the reported subtasks. This suggests that 15k continuation steps are likely insufficient to change zero-shot behavior on Global-MMLU-DE. Similar to the English model, DE-Trans-Init shows large positive shifts on selected subtasks like *high_school_statistics* and on *professional_medicine*, but also large drops on others such as *human_aging* and *machine_learning*. Compared with DE-Trans-Init, BLOOM is lower on all selected top subtasks but consistently higher on all selected bottom subtasks.

6. Related Work

Scientific Datasets We chose the SciLaD dataset because it is the most recently published scientific dataset and contains multiple language splits (Foppiano et al., 2025). However, there are six corpora that have been used for scientific model pre-training in recent research that are similar to the SciLaD dataset. PubMed Abstract[4] and PubMed

[4] https://pubmed.ncbi.nlm.nih.gov *accessed 2026-02-19*

Central[5] (PMC) are two corpora that contain abstracts and full text of biomedical and scientific journal literature from the U.S. National Institutes of Health's National Library of Medicine. S2ORC (Lo et al., 2020) is, with over 81 million open-access papers, the largest corpus of scientific text. This dataset contains open-access papers from the Semantic Scholar literature and is, like SciLaD, processed using GROBID (Grobid). The UnarXive (Saier et al., 2023) corpus draws its texts from arXiv, covering multiple scientific fields. The corpus, with 1.9 million documents, is much smaller than the SciLaD dataset. The ACL Anthology Network is a much smaller scientific corpus containing 24.6 thousand papers on computational linguistics from the ACL Anthology. The only downstream pre-training corpora that cover multiple scientific tasks are the BigBio (Fries et al., 2022) datasets. This dataset contains normalized datasets covering various NLP tasks in the scientific domain. Note that none of these datasets focus on multilinguality.

Scientific LLMs Scientific language models cover domain-specific knowledge from scientific domains such as chemistry, physics, astronomy, material science, life science, and earth science. In recent years, there have been many efforts to publish language models in these fields with a major focus on the medical and biomedical domain (Workshop et al., 2023; Hu et al., 2025). The closest model to our pre-trained T5 models is the SciFive model, which is a T5 model trained on biomedical literature (Phan et al., 2021). This model is trained on two corpora: PubMed Abstract and PMC. However, the SciFive model is English-centric and performs best on classic NLP tasks such as named-entity recognition and relation extraction. Taylor et al. (2022) published the first generative LLM for the scientific domain named Galactica. As this model shows strong performance on mathematical MMLU and downstream tasks, such as PubMedQA (Jin et al., 2019), Galactica is also a monolingual English language model. The first multilingual language model, which is also trained on scientific data, is the BLOOM model (Workshop et al., 2023). This model has been trained on the vast BigBio dataset (Fries et al., 2022). Although it has strong capabilities in multilingual NLP tasks, BLOOM has not been applied to the scientific domain. In recent years, generative LLMs have shown strong capabilities for performing scientific tasks without being fine-tuned solely on scientific domain-specific data (Abdullah et al., 2025; Jiang et al., 2024; Hu et al., 2025).

Language Transfer Methods Language transfer from English-centric language models is an ongoing research topic that has been used mostly for two reasons in the past (Lee et al., 2025). First, to reduce computational costs for non-English language models (Bhukya et al., 2023) and second, to create language models for low-resource languages (Remy et al., 2024). Ostendorff and Rehm (2023) introduced CLP-Transfer, a cross-lingual, progressive transfer learning approach. In this method, cross-lingual transfer is combined with progressive transfer, meaning increasing the model size, which is beyond the scope of this project. Lee et al. (2025) present Cross-Lingual Optimization (CLO) for language transfer using translated data. This approach requires a translation model for performing cross-lingual transfer.

7. Conclusion

In this paper, we present six non-English monolingual scientific language models that have been transferred from a new, continued pre-trained scientific English T5-base model. The English scientific model has been transferred to German, Japanese, Russian, Polish, Spanish, and Portuguese using a pipeline that applies the WECHSEL (Minixhofer et al., 2022) method. The pipeline and the models have been published and can be used to create more non-English scientific language models[6]. We show that scientific knowledge can be preserved through transfer, demonstrating that even for high-quality scientific text, language transfer is beneficial. The non-English models were primarily evaluated in a zero-shot setting on the Global-MMLU benchmark, and the results show strong performance on scientific QA benchmarks, with most surpassing baselines and achieving results comparable to those of multilingual LLMs such as BLOOM. The models have also been published on Hugging Face and are linked in the GitHub repository.

Limitations

While our work provides valuable insights into scientific language transfer and the shortcomings of language bias in the scientific domain, it also has several limitations. First, there is a severe lack of multilingual or non-English scientific benchmarks. As English is the lingua franca in the scientific domain, there have been few efforts in the community to build benchmarks for classic NLP tasks. Most available benchmarks are automatically translated, limiting the interpretability of the results. A set of

[5] https://www.ncbi.nlm.nih.gov/pmc accessed 2026-02-19

[6] https://github.com/nikolas-rauscher/scientific-english-crosslingual-transfer

manually curated multilingual benchmarks would benefit the analysis of our models.

Second, we face computational limitations while performing our experiments. We are aware that larger models have been trained and released in the past, and that using them for evaluation would, based on the scaling law, increase performance. However, we want to emphasize the rising computational costs of training them and the economic and environmental impacts that come with them. The goal of this work is not to improve state-of-the-art scientific language models but to showcase the usability of language transfer in the scientific domain. However, based on the results of this work, we currently aim to release a multilingual generative language model for the scientific domain using language transfer.

Acknowledgements

This work was supported by the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[7] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.), funded by the Federal Republic of Germany and its states through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). Further support was provided from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101189745 (HIVEMIND).[8]

8. References

Abdulhady Abas Abdullah, Arkaitz Zubiaga, Seyedali Mirjalili, Amir H. Gandomi, Fatemeh Daneshfar, Mohammadsadra Amini, Alan Salam Mohammed, and Hadi Veisi. 2025. Evolution of meta's llama models and parameter-efficient fine-tuning of large language models: a survey.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, et al. 2025. Teuken-7b-base & teuken-7b-instruct: Towards european llms.

Ramesh K. Bhukya, Anjali Chaturvedi, Hardik Bajaj, Udgam Shah, Sumit Singh, and Uma Shanker Tiwary. 2023. Efficiently transferring pre-trained language model roberta base english to hindi using wechsel. In *2023 26th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6.

Luca Foppiano, Sotaro Takeshita, Pedro Ortiz Suarez, Ekaterina Borisova, Raia Abu Ahmad, Malte Ostendorff, Fabio Barth, Julian Moreno-Schneider, and Georg Rehm. 2025. Scilad: A large-scale, transparent, reproducible dataset for natural scientific language processing.

Jason Alan Fries, Leon Weber, Natasha Seelam, et al. 2022. Bigbio: A framework for data-centric biomedical natural language processing.

Grobid. 2008–2026. Grobid. https://github.com/kermitt2/grobid.

Naimul Haque. 2025. Catastrophic forgetting in llms: A comparative analysis across language tasks.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Ming Hu, Chenglong Ma, Wei Li, et al. 2025. A survey of scientific large language models: From data foundations to agent frontiers.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, et al. 2024. Mixtral of experts.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering.

Jungseob Lee, Seongtae Hong, Hyeonseok Moon, and Heuiseok Lim. 2025. Cross-lingual optimization for language transfer in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15100–15119, Vienna, Austria. Association for Computational Linguistics.

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.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

[7] https://www.nfdi4datascience.de
[8] https://hivemind-project.eu

Malte Ostendorff and Georg Rehm. 2023. Efficient language model training through cross-lingual and progressive transfer learning.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. 2024. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp.

Tarek Saier, Johan Krause, and Michael Färber. 2023. unarxive 2022: All arxiv publications preprocessed for nlp, including structured full-text and citation network. In *2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, page 66–70. IEEE.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, et al. 2025. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation.

C Tardy. 2004. The role of english in scientific communication: lingua franca or tyrannosaurus rex? *Journal of English for Academic Purposes*, 3(3):247–269.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Transformer Encoders with Heuristic-Guided Contrastive Learning for Software Coreference Resolution

Mahmoud Hassan[1] **Dipendra Yadav**[2]
[1]ZBW - Leibniz Information Centre for Economics
[2]University of Greifswald
[1]m.hassan@zbw.eu, [2]dipendra.yadav@uni-greifswald.de

Abstract

This paper describes our system submitted to the Software Mention Detection and Coreference Resolution (SOMD) 2026 shared task, specifically for Subtask 1 (cross-document coreference resolution over gold-standard mentions) and Subtask 2 (cross-document coreference resolution over predicted mentions). The proposed approach employs a SciBERT architecture trained with Supervised Contrastive (SupCon) loss to generate dense mention representations, which are then clustered using Hierarchical Agglomerative Clustering (HAC) with average linkage. Software-aware heuristics are integrated to exploit domain-specific signals such as software name canonicalization and developer disambiguation to adjust pairwise similarity scores before clustering. The system achieved strong performance, with a CoNLL F1 score of 92.18% on coreference resolution over gold-standard mentions and 91.87% on coreference resolution over predicted mentions, showing significant performance of our approach in this area for human annotated and automated systems respectively.

Keywords: Coreference resolution, Software mention, BERT, Clustering, Contrastive Learning

1. Introduction

Software plays a critical role in scientific research, yet it remains one of the most inconsistently cited entities in scholarly publications (Schindler et al., 2024). The same software tool may be referenced under different names, versions, or abbreviations in multiple documents, making cross-document coreference resolution (CDCR) an essential task for building comprehensive research knowledge graphs. the shared task Software Mention Detection and Coreference Resolution (SOMD) 2026 addresses this challenge by focusing on identifying and clustering software mentions in scientific documents.

This paper contributes to two subtasks: Subtask 1 concerns coreference resolution over gold-standard mentions, where mention boundaries and types are provided, and the system must cluster coreferent mentions. Subtask 2 addresses coreference resolution over predicted mentions, where the system must handle noisier input from an upstream mention detection module. Both subtasks require grouping all mentions that refer to the same software entity into coreference clusters. Our approach focuses on a robust, scalable architecture that balances neural semantic similarity with domain-specific logic.

The system is built around three core components: (1) a SciBERT encoder (Beltagy et al., 2019) trained with supervised contrastive loss (Khosla et al., 2020) to produce cluster-friendly embeddings; (2) a set of software-aware heuristics that leverage domain-specific signals such as software name canonicalization, version numbers, and developer metadata to adjust pairwise similarity scores; and (3) Hierarchical Agglomerative Clustering (HAC) (Manning et al., 2008) for final cluster formation. This paper details the proposed approach, discusses the rationale behind each design decision with illustrative examples, and presents the experimental results.

All relevant code and implementations have been made publicly available on GitHub[1] for reproducibility and further research.

2. Related Work

Coreference resolution involves grouping textual expressions (mentions) referring to the same real-world entity into clusters (Pradhan et al., 2012). Although intra-document coreference has seen significant advances with Transformer-based models (Joshi et al., 2020), Cross-Document Coreference Resolution (CDCR) presents a wider search space and increased ambiguity (Cattan et al., 2021).

CDCR systems typically follow either a *Cross-Encoder* architecture, which models token-level interactions between mention pairs at $O(N^2)$ cost, or a *Bi-Encoder* (or Siamese) architecture, which encodes mentions independently into a shared semantic space (Humeau et al., 2020). The latter is preferred for large-scale tasks such as SOMD Subtasks 2 and 3 due to its efficiency in high-volume clustering.

The SoMeSci corpus (Schindler et al., 2021) provides gold-standard annotations for software mentions in scientific articles, including mention types (e.g., usage, creation), software types (e.g.,

[1]https://github.com/MahmoudMHassan/SOMD-2026

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 270–276
12 May 2026.

application, plugin), and relational metadata such as version numbers and developers. The SOMD shared task series (2024 (Krüger et al., 2024), 2025 (Upadhyaya et al., 2025), 2026) builds on this foundation, progressively introducing more challenging subtasks, including mention detection, relation extraction, and cross-document coreference resolution.

2.1. Contrastive Learning for Clustering

Metric learning aims to learn an embedding space where similar items are proximity-wise closer than dissimilar ones. A fundamental objective for this is triplet loss, popularized by FaceNet for face recognition (Schroff et al., 2015). Triplet loss operates on anchor-positive-negative triplets, ensuring that the anchor is closer to the positive by a specific margin.

However, Contrastive learning has emerged as a powerful paradigm for learning representations suitable for clustering tasks. The InfoNCE loss (van den Oord et al., 2018) and its supervised extension, Supervised Contrastive (SupCon) Loss (Khosla et al., 2020), provides a superior representation density for clustering and training models to produce embeddings where the same-class instances are pulled together and different-class instances are pushed apart. Unlike triplet loss, which focuses on a single positive/negative pair per anchor in a "local" update, SupCon contrasts an anchor against all other positives and negatives in a batch simultaneously. This "all-pairs" optimization results in more compact clusters and better separation, making it ideal for the CDCR objective, where the downstream task is direct clustering of the learned representations and the final evaluation is based on cluster metrics like CoNLL F1.

2.2. Data and task description

Given a mention, its type, and relations, the task is to form clusters of mentions (mention ids) referring to the same software. There are three subtasks in the SOMD 2026 challenge, where the difference between the three subtasks lies in the data source and size. For subtask 1, software mentions, types, and relations were human-annotated. For subtask 2, the mentions, types, and relations were automatically predicted by the entity and relation extraction models. The goal is to assess how well the system performs with noisy predictions. For Subtask 3, the data set is also predicted using a baseline model; however, the number of entities is large, and the evaluation would also include the runtime test. This paper describes the approach used in subtask1 and subtask2. Subtask 1 provides gold-standard mentions: 2,974 training mentions in 733 clusters and 743 test mentions. Subtask 2 provides predicted mentions: 2,860 training mentions in 699 clusters and 12,516 test mentions. Each mention includes the surface form, type, document identifier, sentence context, and relational metadata (version, developer, abbreviation).

2.3. Evaluation Metrics

All submissions are evaluated using standard coreference resolution metrics. The metrics are Message Understanding Conference (MUC (Vilain et al., 1995)) which measures how well predicted clusters recover gold coreference links, The B-Cubed metric (B^3 (Bagga and Baldwin, 1998)) which computes mention-level precision and recall based on cluster overlap and the Constrained Entity-Aligned F-measure (CEAFe (Luo, 2005)) which measures the similarity between predicted and gold clusters via optimal alignment. Lastly, the CoNLL Metric ((Pradhan et al., 2012)) is defined as the unweighted macro-average of the F1-scores of MUC, B^3, and CEAFe which is their arithmetic mean. The challenge results were based exclusively on the CoNLL F1-score, and all metrics are computed on the full test set and reported at the cluster level.

3. System description

The system implements a three-stage pipeline: (1) mention encoding using a SciBERT model, (2) heuristic-guided pairwise similarity adjustment, and (3) Hierarchical Agglomerative Clustering for final cluster formation. The same pipeline and trained model are used for both Subtask 1 and Subtask 2.

3.1. Mention Representation

Each mention is represented as a structured text string that combines the mention surface form with its surrounding sentence context. For each mention m, we construct the input as:

`"mention_text [SEP] sentence_context"`

where `mention_text` is the software name (e.g., "MATLAB") and `sentence_context` is the sentence containing the mention. This dual-input format, separated by the `[SEP]` token, is a standard practice in BERT-based entity representation tasks (Soares et al., 2019). It allows the self-attention mechanism to model the interaction between the entity name and its specific usage context, which is critical for disambiguating mentions like "Python" (the language) versus "Python" (the snake).

For example, given the mention "NQuery Advisor" appearing in the sentence "NQuery Advisor by StatsSols version 3.0 was used to calculate sample size", the input to the encoder becomes:

"NQuery Advisor [SEP] NQuery Advisor by StatsSols version 3.0 was used to calculate sample size".

This format exploits BERT's [SEP] token to delineate the mention identity from its context, allowing the model to attend to both components. The surface form of mention appearing twice—once isolated and once in context—provides the encoder with complementary views of the entity.

3.2. Model Architecture

A transformer based approach with shared weights is built on SciBERT[2] (Beltagy et al., 2019), a BERT-based model pre-trained on 1.14 million scientific papers. Its domain-specific vocabulary (scivocab) provides better subword tokenization for software names than general-purpose BERT, reducing information loss from excessive subword fragmentation. The architecture consists of three components: (1) a SciBERT encoder producing 768-dimensional token representations; (2) a Pooling Layer that extracts the [CLS] token as the aggregate sequence embedding; and (3) a Projection Head—a two-layer MLP (Linear$(768, 512) \rightarrow$ ReLU $\rightarrow$ Linear$(512, 256)$)—that maps the embedding to a 256-dimensional contrastive space. As noted by (Chen et al., 2020), this non-linear projection preserves the pre-trained features during contrastive training. Each mention is encoded independently in $O(n)$ passes, with pairwise similarities computed over cached embeddings, enabling efficient scaling to the 12,516 mentions in Subtask 2.

3.3. Supervised Contrastive Loss

The model is trained with supervised contrastive loss (SupCon) (Khosla et al., 2020). Unlike Triplet Loss (Schroff et al., 2015), which requires expensive semi-hard negative mining and optimizes one anchor–positive–negative triplet at a time, SupCon leverages the full label structure in each batch: for an anchor mention A in cluster C, all other mentions from C are treated as positives and all others as negatives simultaneously.

Consider a batch with three mentions of "MATLAB" (cluster A) and two of "SPSS" (cluster B). In a single step, SupCon (1) pulls all MATLAB mentions together, (2) pulls both SPSS mentions together, and (3) pushes all MATLAB mentions away from all SPSS mentions. This "all-pairs" global optimization produces denser intra-cluster and clearer inter-cluster separation than triplet loss, which is precisely the structure required for downstream clustering evaluated by CoNLL F1.

[2] https://huggingface.co/allenai/scibert_scivocab_uncased

The loss uses a temperature $\tau = 0.07$ (Chen et al., 2020), which sharpens decision boundaries and increases sensitivity to subtle differences between non-coreferent mentions (e.g., "MATLAB" vs. "MATLAB R2019b").

3.4. Software-Aware Heuristics

Software coreference resolution carries strong domain-specific signals that pure neural models may not fully exploit due to the lexical and semantic sparsity of software mentions. We introduce a heuristic layer that modifies the pairwise cosine distance matrix before clustering, framing this as a specialized form of domain-specific Entity Resolution (ER) (Čuljak et al., 2022).

Concretely, we apply two heuristics to adjust the similarity score $s(m_i, m_j)$ between any two mentions m_i and m_j of distinct documents:

Name Canonicalization We compute a canonical form $C(m)$ (lowercase, punctuation removal, abbreviation expansion). If $C(m_i) = C(m_j)$, we apply a boost $\delta_{\text{name}} = 0.5$ to the similarity score.

Developer Disambiguation Conflicting developer metadata (e.g. "Adobe" vs "Microsoft") incurs a penalty $\delta_{\text{dev}} = -0.8$, acting as a strong disambiguator for semantically similar entities (e.g., different PDF readers). This disambiguates semantically similar entities, such as "PyTorch" and "TensorFlow", which share nearly identical contextual embeddings (as deep learning frameworks) but are distinct software tools.

Each adjustment is applied additively to the cosine similarity matrix. Delta values ($\delta_{\text{name}} = 0.5$, $\delta_{\text{dev}} = -0.8$) were determined through a grid search in the validation split, sweeping both parameters independently over the range $[-1.0, +1.0]$ in increments of 0.1, and selecting the combination that maximized CoNLL F1 on held-out clusters. The large magnitude of δ_{dev} reflects the empirical finding that conflicting developer metadata is a near-definitive signal of non-coreference, while the moderate positive value of δ_{name} reflects that canonical name equality is strong but not infallible evidence of coreference (e.g., two distinct tools may share a generic name).

3.5. Hierarchical Agglomerative Clustering

Given the heuristic-adjusted similarity matrix S', we apply Hierarchical Agglomerative Clustering (HAC) to produce the final clusters of coreference. We define the distance between two mentions as $D'_{ij} = 1 - S'_{ij}$. HAC iteratively merges clusters C_a and C_b based on the average linkage criterion, which

minimizes the average distance between all pairs of mentions across the two candidate clusters:

$$D(C_a, C_b) = \frac{1}{|C_a||C_b|} \sum_{m_i \in C_a} \sum_{m_j \in C_b} D'_{ij} \quad (1)$$

The choice of average linkage over alternatives like Connected Components (which is functionally equivalent to single-linkage clustering (SLC)) or absolute complete linkage is motivated by robustness. Single-linkage methods are highly susceptible to the "chaining effect" (Čuljak et al., 2022), where a single noisy pair-wise prediction can erroneously merge two semantically distinct software clusters (semantic drift). Average linkage requires a consensus between all mention pairs, ensuring that the merged cluster remains coherent.

A fixed distance threshold $\theta = 0.5$ was used as the stopping criterion. This approach is naturally suited to CDCR as it does not require a-priori knowledge of the number of clusters K. The threshold was tuned in the validation set to maximize CoNLL F1.

3.6. Training Configuration

In CDCR, the data splitting strategy is critical to prevent data leakage and ensure the model learns generalized entity representations. We employ a Cluster-Level (Equivalence-Class) Splitting strategy: rather than splitting mentions independently, we assign entire coreference clusters to either the training or validation set. This ensures that if a software name (e.g., "SPSS") appears in the training set, it is completely absent from the validation set.

This strategy prevents the model from achieving high performance simply by memorizing specific software names; instead, it must learn to identify coreference based on context and surface-form patterns that generalize to unseen software entities.

The model is trained using the AdamW optimizer with a learning rate of 2×10^{-5} and a batch size of 16 for 30 epochs. We implement early stopping with a patience of 4, monitored using CoNLL F1 validation. For Subtask 1, early stopping triggered at epoch 10. For Subtask 2, early stopping triggered at epoch 5. All experiments were conducted on a single Google Colab T4 GPU environment.

4. Results

For Subtask 1, our system produces 261 clusters from 743 test mentions. For Subtask 2, the same model and pipeline are applied to 12,516 test mentions, resulting in 2641 clusters. The evaluation results could be shown in Table 1 and Table 2 for the tasks, respectively.

Metric	Precision	Recall	F1-Score
B^3	0.96	0.92	0.94
CEAFe	0.80	0.93	0.86
MUC	0.99	0.95	0.97
CoNLL F1	**0.92**		

Table 1: Evaluation results on the test phase of subtask 1

Metric	Precision	Recall	F1-Score
B^3	0.97	0.89	0.93
CEAFe	0.78	0.94	0.85
MUC	0.99	0.96	0.98
CoNLL F1	**0.92**		

Table 2: Evaluation results on the test phase of subtask 2

4.1. Comparative Analysis of Model Variants

To evaluate the impact of our architectural choices, we compared our best system (Bi-Encoder with SupCon, Heuristics, and HAC) with three experimental variants explored in our preliminary runs as shown in Table 3.

Setup	CoNLL F1
Bi-Encoder + InfoNCE	0.79
Bi-Encoder + HNM	0.81
Cross-Encoder Re-ranking	0.84
Bi-Encoder + SupCon + Heuristics + HAC	**0.92**

Table 3: Comparison of SciBERT-based model variants on the Test phase of Subtask 1.

The first variant serves as a baseline, utilizing a SciBERT architecture trained with standard InfoNCE loss and clustered via graph-based connected components (Single-Linkage Clustering). The second variant incorporates periodic hard negative mining (HNM) during training to improve cluster separation in the embedding space. The third variant employs a Cross-Encoder re-ranking stage to refine the clusters produced by the Bi-Encoder; however, this approach introduces a $O(N^2)$ complexity at the re-ranking phase, which is problematic for large-scale tasks like Subtask 2. Our best system, which integrates Supervised Contrastive (SupCon) loss and metadata-aware heuristics with Hierarchical Agglomerative Clustering (HAC) achieves better performance while maintaining linear scaling for encoding, which (Humeau et al., 2020) identify as a critical requirement for production CDCR systems.

Mention level vs Cluster level splitting The impact of using mention level splitting was observed in the beginning of the experimentation which caused data leakage where the model "sees" the same entity in both training and validation sets. During validation, it merely recognizes an entity it has already optimized for, rather than learning the generalized property of coreferent software mentions. It also caused cluster fragmentation which breaks the "equivalence class" nature of CDCR. Validation metrics such as CoNLL F1 were reduced because the gold standard for validation only contains a subset of each cluster, making it impossible to achieve perfect recall. In contrast to cluster level splitting which ensured zero leakage, where validation tests the model on software entities it has never seen. This simulates the real-world scenario (Subtask 2, 3) where the model must resolve the mentions for arbitrary software. It also ensured cluster integrity where Gold clusters in the validation set remain complete. The scoring metrics then reflect the true ability of the system to group the mentions.

SupCon vs. InfoNCEloss Following (Khosla et al., 2020), we observe that the supervised contrastive loss with HAC is more robust than InfoNCEloss with single-linkage clustering (SLC) as discussed in subsection 3.3 for CDCR. SupCon's global batch optimization prevents the "local minima" problems, where a model may satisfy the margin for a specific cluster while ignoring broader cluster dynamics.

4.2. The Role of Heuristics

Software Heuristics serve as a critical domain-specific refinement layer that complements the raw semantic power of SciBERT embeddings. These heuristics—which include name canonicalization, developer conflict detection, and exact name matching—provide a "safety net" that raw embeddings often lack. Their primary value over the variants presented in section 4.1 is the ability to enforce hard constraints and logical penalties based on metadata. For instance, the "Developer Conflict" heuristic prevents merging mentions that are semantically similar but belong to different developers, effectively reducing false positives in the clustering process. By adjusting the distance matrix before Hierarchical Agglomerative Clustering (HAC), these heuristics ensure that the final clusters respect known software attributes that deep learning models might otherwise overlook. To isolate the contribution of software-aware heuristics, we evaluate our system with and without the heuristic adjustment layer on the validation split as shown in Table 4. It was observed that the name canonicalization step specifically made a significant improvement to the final output. One of the main differences in the setup of the first 3 variants shown in Table 3, that it embeds the metadata (type, relations) that come with the training data directly into the SciBERT input string ex. `[MENTION] name [TYPE] type [REL] relations [SEP] sentence` , which may contain version and developer information directly in the input string, the SciBERT model learns to bake the heuristics into the embeddings themselves during training, allowing the model to learn the heuristics. In contrast, to our best system which used a simpler input as discussed in section 3.1 and relies on the aforementioned external "boost/penalty" adjustments.

Configuration	CoNLL F1
Bi-Encoder + SupCon + HAC (no heuristics)	0.41
+ Developer Disambiguation (δ_{dev}=-0.8)	0.46
+ Name Canonicalization (δ_{name}=+0.5)	0.94

Table 4: Incremental impact of software-aware heuristics on CoNLL F1 (validation split). Each row adds one heuristic to the previous configuration.

4.3. Qualitative Analysis

To illustrate the behavior of the system, we present representative cluster predictions and common error patterns from Subtask 1.

Successful Clustering The system correctly groups mentions of "SPSS" appearing in diverse contexts—"SPSS v20," "SPSS Statistics," and "SPSS (IBM Corp)"—into a single cluster. The name canonicalization heuristic (δ_{name}) increases pairwise similarity for these surface variants, while the SupCon-trained embeddings capture shared contextual patterns (e.g., "statistical analysis was performed using . . . ").

Developer Disambiguation Success The system correctly separates "Acrobat Reader" (Adobe) from "Foxit Reader" (Foxit Software), despite both sharing the contextual phrase "PDF reader." The developer disambiguation penalty ($\delta_{\text{dev}} = -0.8$) pushes these otherwise similar embeddings apart, preventing an erroneous merge.

Typical Error: Rare Software The most common errors occur with rare or domain-specific software tools that appear only once or twice in the corpus (singletons). Without sufficient training signal, the encoder produces generic embeddings for

such mentions, occasionally merging them with semantically similar but distinct tools. For example, a niche bioinformatics pipeline may be incorrectly clustered with a more common tool mentioned in a similar analytical context.

5. Conclusion

We presented a pipeline for cross-document software coreference resolution, combining a SciBERT bi-encoder trained with Supervised Contrastive (SupCon) loss, software-aware heuristics, and Hierarchical Agglomerative Clustering, achieving a CoNLL F1 of 92.18% and 91.87% on Subtask 1 and Subtask 2 respectively using an identical model across both. Our analysis highlights three core findings: SupCon's all-pairs optimization is directly aligned with cluster-based evaluation, making it particularly effective for CDCR; software-aware heuristics and neural representations are complementary, with symbolic constraints enforcing boundaries that embeddings alone cannot reliably capture; and cluster-level splitting is essential to ensure genuine generalization to unseen software entities.

5.1. Limitations and Future Work

Although the current system achieves high performance, our analysis of the optimized implementation reveals some runtime and technical constraints that offer opportunities for future research.

5.1.1. Limitations

Heuristic Brittleness Our software-aware heuristics rely heavily on regular expressions and fixed string matching. While effective for common tools like "MATLAB" or "SPSS," this approach is brittle when encountering non-standard software naming conventions or complex, nested version strings (e.g., "v.2.1-beta.rev4"). Furthermore, the heuristics depend on the quality of relation extraction from the previous pipeline stage; errors in developer or version detection propagate directly into the coreference scoring.

Runtime and Scalability The proposed pipeline exhibits three distinct asymptotic stages: (i) SciBERT encoding is $O(n)$, producing one embedding per mention in a single forward pass; (ii) heuristic adjustment is $O(n^2)$, iterating over all mention pairs to apply name canonicalization and developer-metadata comparisons; and (iii) HAC has a worst-case complexity of $O(n^3)$, reduced to $O(n^2 \log n)$ in practice via SciPy's Lance–Williams update formula. Crucially, the quadratic and super-linear stages operate over lightweight precomputed scalar distances, not neural forward passes, which justifies their inclusion relative to a Cross-Encoder baseline that would require full neural forward passes $O(n^2)$. Nevertheless, both the heuristic and the HAC stages require materializing the entire $n \times n$ similarity matrix. This quadratic memory and time footprint make the current pipeline infeasible as-is for datasets substantially larger than Subtask 2, such as Subtask 3.

Global Clustering Threshold We employ a fixed global threshold ($\theta = 0.5$) for HAC. This assumes that the optimal merging distance is consistent across all software domains. In practice, mature software ecosystems with many sub-modules (e.g., "Apache" projects) may exhibit different semantic densities than monolithic tools, potentially requiring adaptive or cluster-specific thresholds.

5.1.2. Future Work

Knowledge Graph Grounding Beyond lexical heuristics, grounding software mentions in external Knowledge Graphs (KGs) such as WikiData or specialized software catalogs (e.g. SoftwareKG) would provide a robust disambiguation signal. Symbolic relations from a KG (e.g., *owned_by*, *successor_of*) could replace manual developer rules, allowing the system to resolve aliases that are semantically distinct but referentially identical.

Approximate Nearest Neighbors and Poly-encoders To address scalability, we plan to integrate FAISS-based approximate nearest neighbor search to narrow the clustering search space. Additionally, replacing the Bi-Encoder with a Poly-encoder (Humeau et al., 2020) could provide the fine-grained token interaction of a Cross-Encoder while maintaining the inference speed required for large-scale CDCR.

Adaptive Threshold Learning We intend to replace the fixed threshold with a learned similarity metric or a meta-learning approach that adjusts merging criteria based on the local density of the embedding space, improving performance in highly fragmented software clusters.

6. Acknowledgments

The authors thank the organizers of the Shared task on Software Mention Detection and Coreference Resolution (SOMD) and the Natural Scientific Language Processing (NSLP) workshop for running this competition. We also thank the reviewers for their insightful and constructive comments, which helped raise the standard of this manuscript considerably.

7. Bibliographical References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566.

I. Beltagy, K. Lo, and A. Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

A. Cattan, A. Eirew, G. Stanovsky, M. Joshi, and I. Dagan. 2021. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, pages 1597–1607.

I. Čuljak, A. Spitz, R. West, and A. Arora. 2022. Strong heuristics for named entity linking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1826–1845.

S. Humeau, K. Shuster, M. Lachaux, and J. Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 18661–18673.

Frank Krüger, Saurav Karmakar, and Stefan Dietze. 2024. SOMD@NSLP2024: Overview and insights from the software mention detection shared task. In *Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, pages 216–233, Cham. Springer Nature Switzerland.

X. Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pages 25–32.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea.

D. Schindler, F. Bensmann, S. Dietze, and F. 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 and Knowledge Management (CIKM 2021)*, pages 4574–4583.

David Schindler, Tazin Hossain, Sascha Spors, and Frank Krüger. 2024. A multi-level analysis of data quality for formal software citation.

F. Schroff, D. Kalenichenko, and J. Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2895–2905.

Sharmila Upadhyaya, David Schindler, Frank Krüger, and Stefan Dietze. 2025. SOMD2025: A challenging shared task for software related information extraction. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*. Association for Computational Linguistics.

A. van den Oord, Y. Li, and O. Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding (MUC-6)*, pages 45–52.

UniCite: A Dataset and Unified Hierarchical Taxonomy for Multi-Dimensional Citation Analysis

Amina Mourky, Elena Leitner, Julian Moreno Schneider,
Raia Abu Ahmad, Ekaterina Borisova, Georg Rehm
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
Salzufer 15/16, 10587 Berlin, Germany
{firstname.lastname}@dfki.de

Abstract

Research in Citation Context Analysis (CCA) has produced numerous taxonomic schemes that vary from three to 12+ categories, with different granularities and no mappings between frameworks, severely limiting systematic comparison and progress. Despite decades of study, CCA methods have largely relied on fragmented frameworks that treat citation tasks independently, ignoring systematic relationships between function classification, sentiment analysis, and importance assessment. To address these research gaps, we present three integrated contributions. First, we develop UniCite, a two-level taxonomy (six primary functions, 12 subcategories, two orthogonal dimensions) that systematically integrates three existing schemes. Second, we develop a comprehensive dataset of 4,017 citations combining established resources with 1,547 newly extracted citations from 2018-2024 publications, all manually annotated under our unified framework. Third, we demonstrate systematic task relationships through multi-task learning, achieving 21.1% relative improvement in subfunction classification over single-task approaches.

Keywords: Citation Context Analysis, Scholarly Document Processing, Citation Classification

1. Introduction

Citations encode complex rhetorical relationships that extend far beyond simple bibliographic acknowledgment. These relationships reflect how scholar communities build knowledge, evaluate, and position new work within existing discourse (Garfield et al., 1964; Swales, 1986). Understanding citation functions has implications for scientific evaluation, literature analysis, and computational approaches to scholarly communication.

Citation Context Analysis (CCA) studies how and why authors cite previous work. This can involve different aspects including the classification of citation functions, sentiment toward cited work, and evaluation of citation importance to the citing paper's contributions. As scholarly publications accelerate and evolve, accurate citation analysis becomes increasingly important for bibliometric evaluation and understanding knowledge flow patterns.

However, CCA research faces two fundamental challenges that limit systematic progress. First, different groups have developed taxonomic schemes that vary in granularity and focus (e. g., Jurgens et al., 2018; Cohan et al., 2019). This variation in category boundaries, terminology, and annotation guidelines prevents cumulative research and systematic comparison across studies. Second, many established datasets focus on pre-2019 publications, failing to capture contemporary scholarly practices (Jurgens et al., 2018; Cohan et al., 2019; Lauscher et al., 2022).

We address these challenges through UniCite, a unified approach that integrates existing taxonomies while extending temporal coverage. Our hierarchical framework preserves semantic distinctions from existing schemes by integrating the argumentative categories from Teufel et al. (2006), the citation frames from Jurgens et al. (2018), and the cross-domain applicability from Cohan et al. (2019), while enabling systematic mapping and comparison as shown in Figure 1. The taxonomy operates on two-levels: Level 1 captures six broad functions (Background, Methodology, Extension, Comparison, Motivation, Future Work), while Level 2 provides twelve targeted categories that distinguish important cases within major functions, such as separating General Background from Previous Work citations, or differentiating Direct Use from Tool Usage in methodology citations. We supplement this with the comprehensive manual annotation of contemporary citations and validate our design through multi-task learning experiments that reveal systematic relationships between citation analysis tasks. Our data and code are publicly available at `https://github.com/aminamourky/UniCite.git`.

This paper makes three primary contributions:

1. **Unified Hierarchical Taxonomy:** We develop UniCite, a framework that systematically integrates existing classification schemes while preserving their semantic distinctions through a two-level hierarchy with orthogonal sentiment and importance dimensions.

2. **Comprehensive Annotated Dataset:** We develop a dataset of 4,017 citations, annotated by experts, that addresses temporal gaps by

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 277–288
12 May 2026.

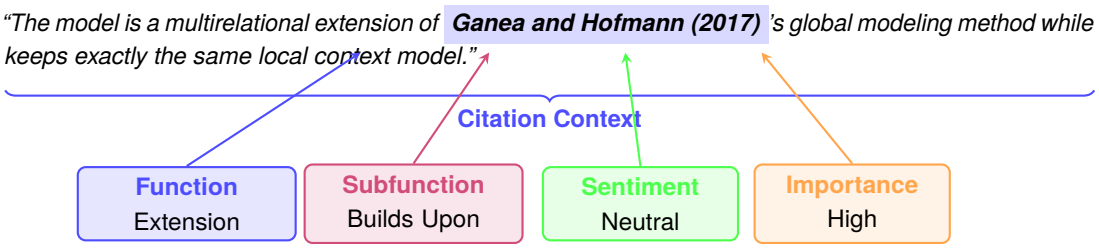


Figure 1: Example of an annotated citation from the UniCite dataset.

integrating established resources with 1,547 recent citations from 2018-2024 publications.

3. **Multi-Task Learning Validation:** We explore systematic relationships between citation analysis tasks through multi-task learning experiments, achieving substantial performance improvements that validate our taxonomic design and reveal beneficial task interactions.

2. Related Work

The computational analysis of citations has evolved along several research threads, each attempting to capture different aspects of how scholars use references in their writing. The foundations of CCA can be traced back to the early bibliometric work of Garfield et al. (1964), which established that citations fulfill distinct rhetorical roles, followed by early structured approaches to citation categorization (Moravcsik and Murugesan, 1979).

Early work established that citations could be automatically classified based on their rhetorical purpose. Teufel et al. (2006) developed a twelve-category system that provided detailed argumentative distinctions, proving that computational models could learn citation intent from textual features. Different groups subsequently developed frameworks reflecting their particular priorities. Jurgens et al. (2018) created a six-category framework targeting main citation functions, demonstrating how citation functions vary across paper sections. Cohan et al. (2019) introduced the SciCite dataset with a simplified three-category scheme arguing for a broader applicability across different scientific domains. Pride and Knoth (2020) developed the Argumentative Citation Types (ACT) framework by presenting a multi-dimensional approach that separates citation function from citation importance, establishing an orthogonal dimension that captures how central a citation is to the work's contributions.

The diversity of these approaches shows the genuine disagreements about how citation behavior should be modeled. Some researchers have argued for even more complex representations. Jochim and Schütze (2012) developed four orthogonal dimensions to capture complex citation relationships that one-dimensional frameworks fail to represent. Hernández-Álvarez et al. (2017) created the CONCIT scheme, which provided a rich annotation of citation contexts including detailed markup of citation aspects and evaluative indicators, emphasising the critical importance of fine-grained contextual information in citation analysis applications. Lauscher et al. (2022) challenged the single-label assumption, allowing citations to be assigned multiple functions simultaneously and thus revealing that compound citation purposes are common in scholarly writing.

These parallel developments have created a field where different communities work with diverse data and evaluation frameworks that restrict direct comparison. The lack of standardization makes it difficult to assess which approaches actually work better for real applications. The OpenCitations Data Model (Daquino et al., 2020) provides an ontological framework for representing bibliographic entities and citation relationships, enabling structured representation that complements taxonomic approaches. Additional challenges arise from metadata limitations in some existing datasets. For example, MultiCite (Lauscher et al., 2022) lacks paper titles. While it provides paper IDs, these are no longer traceable to the original publications, limiting researchers' ability to access broader publication context when needed. Such gaps can constrain analysis capabilities and cross-domain studies.

An additional limitation involved the artificial separation of related analytical tasks. Function, classification, sentiment analysis, and importance assessment are typically studied separately, even though they all require nuanced contextual interpretation of citation text. Research in related areas has shown that jointly modeling such tasks can improve performance through shared representations (Ruder, 2017). Recent work in citation analysis has begun exploring these approaches: Su et al. (2019) demonstrate performance improvements through neural multi-task learning for citation function and provenance, while Baig et al. (2021) achieve competitive results using multi-module architectures for citation context classification. Recent work also

emphasizes the importance of comprehensive context extraction, with Jantsch et al. (2025) showing that fine-grained citation contexts capturing semantic dimensions beyond single sentences improve classification performance by up to 25%. However, citation analysis has yet to systematically explore whether function, sentiment and importance annotations benefit from shared modeling or exhibit overlapping representational structure across comprehensive taxonomic frameworks.

Recent work has explored complementary approaches to citation classification. Jiang (2025) demonstrates that ensemble methods combining diverse base classifiers substantially improve performance, addressing the observation that no single model excels universally. Similarly, Duan and Tan (2026) propose a semantically orthogonal framework separating citation intent from content type, achieving superior cross-domain generalization. While LLM-based approaches have emerged (Koloveas et al., 2025), Kunnath et al. (2023) systematically evaluate prompting strategies for citation classification, finding that dynamic context-based prompting outperforms standard fine-tuning, while zero-shot GPT-3.5 performs well on multidisciplinary data but poorly on domain-specific datasets. Fogelson et al. (2025) further identify significant reproducibility challenges with LLM-based approaches, validating fine-tuned domain-specific models for methodological transparency. Beyond citation classification, Bolaños et al. (2025) propose a complementary annotation schema for classifying rhetorical roles in literature review sections, demonstrating that fine-tuned LLMs achieve strong performance on scholarly sentence classification tasks.

3. Multidimensional Taxonomy

To enable systematic comparisons in CCA, we develop a unified taxonomy (Figure 2) that integrates existing approaches[1] such as SciCite (Cohan et al., 2019), MultiCite (Lauscher et al., 2022), and Jurgens-CFC (Jurgens et al., 2018) into a hierarchical framework considering citation function, sentiment, and importance.

The citation function consists of two levels of granularity – six primary categories in Level 1 and twelve subcategories in Level 2.

Level 1 Functions cover the main ways researchers use citations. *Background* establishes context or foundations. *Methodology* involves applying cited methods, tools, or datasets. *Extension* shows how the current work builds upon or modifies the cited work. *Comparison* evaluates cited work against current research or other studies. *Motivation* justifies the need for the current research by pointing to gaps in existing work. *Future Work* discusses potential future research directions.

Level 2 Subcategories distinguish important cases within major functions: **Background** separates domain knowledge (*General Background*) from specific research findings (*Previous Work*). **Methodology** differentiates between applying methods (*Direct Use*), using software tools (*Tool Usage*), or utilising datasets (*Dataset Usage*). **Extension** separates conceptual foundations (*Builds Upon*) from adapted applications (*Adaptation*). **Comparison** distinguishes conceptual evaluation (*Conceptual Comparison*), performance assessment (*Performance Comparison*), and negative assessment (*Critique*). **Motivation** and **Future Work** remain without subcategories, as they represent relatively focused functions. We adopt a single-label design to maintain annotation feasibility and enable systematic comparison with existing benchmarks, with multi-label extension left for future work.

Orthogonal properties capture evaluative and structural aspects independent of function. **Sentiment** reflects the author's stance: *Positive* (favorable), *Neutral* (objective), or *Negative* (critical). **Importance** assesses centrality of the cited work to the current paper: *High* (essential to the paper's contribution), *Low* (peripheral), *Cannot Determine* (unclear from context).

4. Dataset

We develop a comprehensive dataset through a two-stage process: (1) systematic conversion of all existing citations to our unified Level 1 taxonomy, and (2) manual annotation of a representative subset for complete multi-dimensional labeling.

4.1. Source Data Collection and Mapping

Three established datasets form our foundation, selected for their complementary coverage and annotation quality. **SciCite** (Cohan et al., 2019) provides 11,020 citation contexts spanning computer science and biomedical domains with three-category function labels. **MultiCite** (Lauscher et al., 2022) covers 12,653 citation contexts from computational linguistics with seven function categories and detailed context modeling. For our dataset, we extracted only the single-label citations to maintain consistency with our single-label classification approach. **Jurgens-CFC** (Jurgens et al., 2018) offers 1,969 expertly annotated citations with six function categories that established a benchmark for citation function classification.

We supplement these resources with 1,547 new citations from publications spanning

[1] For more details on the compatibility with existing systems, see Appendix A, Section A.1.

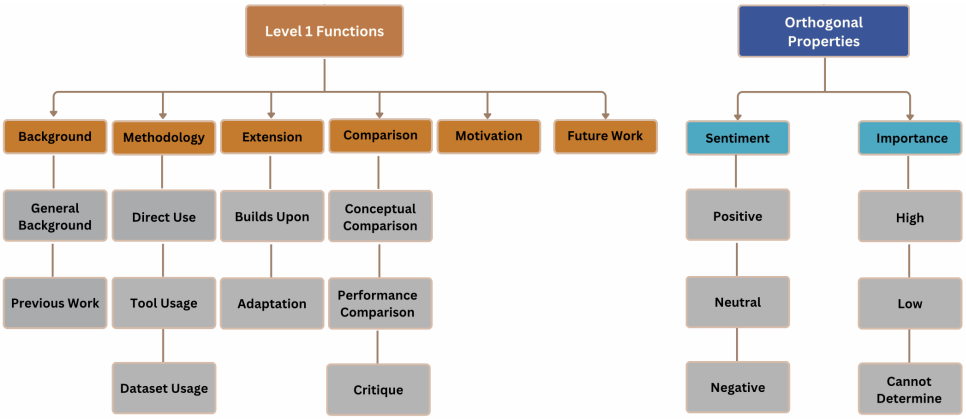


Figure 2: UniCite taxonomy with its hierarchical organization of citation functions and orthogonal properties.

2018-2024, a period marked by the rise of preprint culture, accelerated publication cycles, and growing interdisciplinary collaboration, which may have introduced new citation patterns not captured in earlier datasets. Computer science venues (ACL/EMNLP/NAACL, NeurIPS/ICML/ICLR, CVPR/ICCV/ECCV, ICSE/FSE/ASE) provided 1,200 citations. Interdisciplinary venues contributed 347 citations from robotics (RSS/ICRA/IROS) and human-computer interaction (FAccT/CHI). Paper selection was stratified by impact metrics and temporal distribution. We used PyMuPDF[2] with citation recognition to process PDF files. We extracted citation context by keeping 2–3 sentences surrounding citation markers with validation for author name, publication year, and format consistency, following established practices showing that multi-sentence contexts significantly improve classification performance over single sentences (Jantsch et al., 2025; Lauscher et al., 2022).

Our approach first converts all 27,189 citations from source datasets to our Level 1 taxonomy through systematic mapping rules with full automation. Direct mapping rules handle 89% of conversions, with the remaining 11% requiring merging operations where multiple source categories map to single target categories. However, complete multi-dimensional annotation (including Level 2 subcategories, sentiment, and importance) requires manual annotation that is expensive for the full dataset. We therefore selected a representative 4,017-citation subset for complete manual annotation, balanced across source datasets and supplemented with 1,547 newly extracted recent citations. This sampling strategy ensures coverage of existing taxonomic diversity while maintaining annotation feasibility and adding crucial temporal coverage.

[2] https://pymupdf.readthedocs.io/en/latest/

4.2. Annotation Process and Agreement

We developed comprehensive annotation guidelines[3] through an iterative process involving a pilot annotation phase and edge case analysis. Two graduate students with backgrounds in computational linguistics, both with prior experience in citation analysis, performed the annotation using INCEpTION (Klie et al., 2018). For the 2,470 citations converted from existing datasets, Level 1 labels were obtained through automated mapping, while Level 2 subcategories and orthogonal properties were annotated manually. For the 1,547 newly extracted citations, annotators provided complete annotations across all taxonomic dimensions. The workflow proceeded hierarchically: the annotators first assigned the Level 1 function, then selected the Level 2 subfunction, followed by sentiment and importance. For importance assessment, annotators consulted the full paper when available, which was the case for all but 31 citations (0.8% of the dataset).

We followed established protocols for quality assurance from CCA literature (Teufel et al., 2006; Cohan et al., 2019; Jurgens et al., 2018). The two annotators underwent 6-hour guideline training, 2-hour pilot annotation, and 3-hour calibration sessions addressing edge cases. We monitored the annotation quality weekly throughout the 8-week annotation period. Inter-annotator agreement assessment covered 1,110 citations (28% of total) with independent dual annotation. Level 1 function classification achieved Cohen's κ = 0.88 (Artstein and Poesio, 2008), demonstrating excellent agreement on primary categories. Level 2 subcategory classification achieved κ = 0.63, reflecting the inherent difficulty of fine-grained distinctions while remaining within acceptable ranges for complex annotation tasks. Sentiment classification achieved κ

[3] The annotation guidelines are available at https://github.com/aminamourky/UniCite/tree/main/annotation.

Dataset	No.	F L1	F L2	Sen	Imp
New citations	1,547	1,547	1,547	1,547	1,547
Jurgens-CFC	892	892	892	892	892
SciCite	828	828	828	828	828
MultiCite	750	750	750	750	750
Total	**4,017**	**4,017**	**4,017**	**4,017**	**4,017**
Manual	–	**1,547**	**4,017**	**4,017**	**4,017**

Table 1: Annotation coverage in UniCite across source datasets and taxonomic levels. Manually annotated labels are in blue.

= 0.92, and importance assessment achieved κ = 0.66. Our agreement scores are strong compared to prior work in citation analysis, considering Cohan et al. (2019) reported 86% agreement on 100 samples, while Lauscher et al. (2022) achieved 0.76 accuracy for function classification. We resolved all disagreements through careful examination of citation contexts and consistent application of our annotation guidelines.

4.3. Dataset Statistics and Composition

Table 1 presents the annotation coverage across source datasets and taxonomic levels. The UniCite dataset balances established annotation quality with contemporary coverage: new paper extractions are the largest component with 1,547 instances (38.5%), followed by Jurgens-CFC (892 instances, 22.2%), SciCite (828 instances, 20.6%), and MultiCite (750 instances, 18.7%).

Temporal distribution addresses existing dataset limitations: 1,750 citations (43.6%) are from 2018-2024 publications, with peak years including 2019 (367 citations), 2023 (313 citations), and 2021 (300 citations), while 2,267 citations (56.4%) maintain historical representation for comparative analysis.

Function distribution reflects realistic scholarly practices (see Figure 3): Background dominates with 1,600 instances (39.8%), followed by Methodology (1,028 instances, 25.6%) and Comparison (790 instances, 19.7%). Subcategory distribution shows Previous Work constituting 1,230 instances (30.6%) of background citations, while General Background comprises 370 instances (9.2%). Methodology subdivides into Direct Use (640 instances), Tool Usage (208 instances), and Dataset Usage (180 instances). Comparison categories include Conceptual Comparison (512 instances), Performance Comparison (230 instances), and Critique (48 instances). Orthogonal properties reflect academic discourse characteristics: Sentiment remains predominantly Neutral (3,753 instances, 93.4%), with Positive (164 instances, 4.1%) and Negative (100 instances, 2.5%) representing evaluative minorities. Importance assessment shows High (2,088 instances, 52.0%) slightly exceeding Low (1,712 instances, 42.6%), with Cannot Determine cases comprising 217 instances (5.4%).

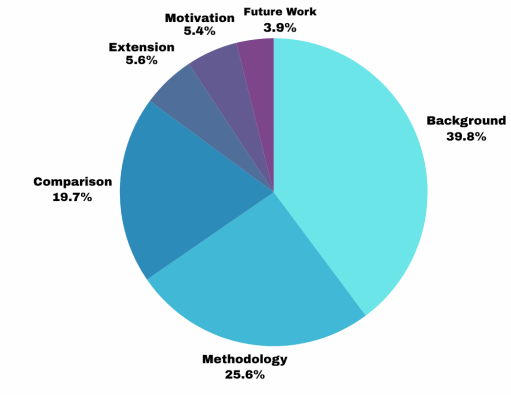


Figure 3: Distribution of function citations.

Domain representation spans Natural Language Processing (1,439 citations, 36.3%), Machine Learning (485 citations, 12.2%), Computer Science and Medicine (476 citations, 12.0%), Computer Vision (405 citations, 10.2%), Software Engineering (403 citations, 10.2%), Robotics (394 citations, 9.9%), and Human-Computer Interaction (360 citations, 9.1%), enabling both domain-specific and cross-domain evaluation.

5. Experiments

We evaluate the effectiveness of the framework through systematic experiments across multiple modeling paradigms. The experimental design addresses three primary research questions: (1) How do different citation analysis tasks interact in multi-task learning scenarios? (2) What benefits does hierarchical conditioning provide for subfunction classification? (3) How do multi-task approaches compare to specialized single-task models?

We use stratified sampling to maintain consistent class distributions across data partitions: Training Set (2,814 citations, 70%), Validation Set (602 citations, 15%), and Test Set (601 citations, 15%). Model performance is measured using accuracy and macro F_1, which addresses class imbalance by providing equal weight to all taxonomic categories. We supplement these with a hierarchical consistency metric that measures the agreement between function and subfunction predictions with respect to our taxonomic structure (Sun and Lim, 2001; Plaud et al., 2024). Further details can be found in Appendix A, Section A.2.

5.1. Reference Models

Large Language Models (LLMs): We evaluate GPT-3.5 Turbo[4] and GPT-4o mini[5] in few-shot configurations to assess zero-training performance on citation analysis tasks. These models were selected to represent two capability tiers of widely accessible commercial LLMs, enabling assessment of how model scale affects few-shot citation classification. GPT-3.5 evaluation uses 1,105 citations with taxonomy definitions plus 2 examples. GPT-4o mini evaluation uses 1,093 citations with expanded prompts containing 12 examples covering diverse label combinations. The difference in citation counts (12 less for GPT-4o mini) results from excluding the example citations used in the expanded prompts to prevent data leakage. Each prompt contains taxonomic guidelines, representative examples, and explicit formatting requirements. The full prompts used for both LLM evaluations are available in the project repository.

Single-Task Fine-Tuned Models: We train individual SciBERT models (Beltagy et al., 2019) for each task using task-specific classification heads. Function and subfunction models use weighted cross-entropy loss with a learning rate of 2e-5. Sentiment classification employs focal loss (Lin et al., 2017) (γ=2.0) for the 93% neutral class imbalance. Importance assessment uses weighted cross-entropy after filtering "Cannot Determine" labels.

5.2. Multi-Task Learning Approaches

Our multi-task learning (MTL) approach builds on established work in CCA. Su et al. (2019) demonstrate that neural architectures can effectively share representations between citation function and provenance tasks, while Baig et al. (2021) show that multi-module designs combining diverse feature types achieve competitive performance. We extend these foundations by investigating systematic relationships across function, sentiment, and importance dimensions.

Hierarchical MTL: Our hierarchical approach follows the hard parameter sharing paradigm established by Ruder (2017), where tasks share hidden layers while maintaining task-specific output layers. The design leverages principles from hierarchical classification (Silla and Freitas, 2011), where taxonomic structure provides natural learning progression from general to specific categories. Task selection follows insights from Standley et al. (2019), who showed that beneficial task relationships depend on semantic similarity rather than superficial domain overlap. The shared SciBERT encoder (Beltagy et al., 2019) generates common representations while task-specific heads process specialized features. A learned embedding of the predicted function category (192-dimensional, i.e., hidden_dim/4) is concatenated with the encoder features (768-dimensional), forming a 960-dimensional combined representation before subfunction classification. Training proceeds through three phases: individual task optimization, hierarchical conditioning introduction, and end-to-end joint learning across 8 epochs.

Progressive MTL: We train tasks sequentially to analyze interaction patterns. Phase 1 trains function and subfunction jointly (10 epochs). Phase 2 adds importance classification (6 epochs). Phase 3 incorporates sentiment analysis (6 epochs). Phase 4 performs final joint optimization (4 epochs). Each phase trains an independent model from scratch with its respective task combination, enabling direct comparison of how different task sets affect performance without confounding transfer effects.

All experiments run across three random seeds (42, 123, 456) with identical data splits. Epoch counts were determined empirically with early stopping (patience of 3–4 epochs) to prevent overfitting. We report means and standard deviations for all metrics, with 95% confidence intervals and t-tests for statistical significance assessment.

6. Results

The results of the few-shot approach, fine-tuning, and MTL on Function (F L1), Subfunction (F L2), Sentiment (Sen) and Importance (Imp) are shown in Table 2. The results of the fine-tuned models represent the mean $\pm$ standard deviation in three random seeds.

Approach	F L1	F L2	Sen	Imp
LLM Few-Shot				
GPT-3.5 Turbo	29.8	24.7	**47.4**	29.8
GPT-4o mini	50.8	39.3	46.3	37.6
Fine-Tuned Models				
SciBERT	68.7±1.7	53.9±0.7	40.9±1.4	71.3±1.2
Multi-Task Learning				
Hierarchical MTL	**72.5**±0.8	**65.3**±0.6	–	–
Progressive MTL	70.1±0.9	60.4±3.2	32.5±0.0	**74.2**±0.9

Table 2: Classification results across all approaches. Best values per task shown in bold.

LLM Performance. The LLM experiments establish task difficulty hierarchies while revealing limitations of general-purpose models for specialized citation analysis. GPT-3.5 Turbo with two examples achieves moderate performance across tasks. GPT-4o mini with 12 examples shows substantial

[4] https://platform.openai.com/docs/models/gpt-3.5-turbo

[5] https://platform.openai.com/docs/models/gpt-4o-mini

Ph	Tasks	F L1	F L2	Imp	Sen
1	Fun+Sub	71.2±0.3	56.7±3.4	N/A	N/A
2	Fun+Sub+Imp	70.1±1.0	58.3±1.6	74.2±0.8	N/A
3	Fun+Sub+Imp+Sen	70.1±0.9	60.4±3.2	74.2±0.9	32.5±0.0

Table 3: Progressive MTL results.

improvements on Function (+21%), Subfunction (+14.6%), and Importance (+7.8%), with a minimal decrease in Sentiment performance (-1.1%) compared to GPT-3.5 Turbo.

Single-Task Performance. Function classification achieves 68.7% ± 1.7% macro F_1. Subfunction classification shows increased difficulty at 53.9% ± 0.7% macro F_1. Sentiment classification exhibits high accuracy (93.0% ± 0.4%) but low macro F_1 (40.9% ± 1.4%) due to class imbalance. Importance assessment demonstrates stable performance (71.3% ± 1.2% macro F_1) with the lowest coefficient of variation (0.018).

Hierarchical MTL. Hierarchical conditioning between function and subfunction tasks produces significant improvements. Subfunction classification gains 11.4 percentage points in macro F_1 (53.9% → 65.3%, $p < 0.01$), representing a 21.1% relative improvement. This improvement substantially exceeds the 2% gains reported by Su et al. (2019) for function-provenance MTL, demonstrating the particular effectiveness of hierarchical conditioning for taxonomically structured tasks. Function performance also shows some improvement with an increase from 68.7% to 72.5% ($p = 0.013$). The conditioning architecture achieves 83.3% ± 0.5% hierarchical consistency between function and subfunction predictions (95% CI: [0.821, 0.845]).

Progressive MTL. Progressive training enables systematic analysis of task interaction patterns (see Table 3). Phase 1 (Function + Subfunction) establishes stable baseline joint performance with 0.712 $\pm$ 0.003 function F_1 and 0.567 $\pm$ 0.034 subfunction F_1. Phase 2 importance addition produces modest function degradation (-0.011 $\pm$ 0.010 F_1, $p > 0.05$) while providing modest subfunction improvement (+0.016 $\pm$ 0.016 F_1), with importance achieving robust performance (0.742 $\pm$ 0.008 F_1). Phase 3 sentiment integration shows negligible additional function impact (+0.001 $\pm$ 0.001 F_1) while providing modest subfunction improvements (+0.021 $\pm$ 0.022 F_1). Sentiment classification achieves consistent but poor performance (0.325 $\pm$ 0.000 F_1) across all seeds due to severe class imbalance.

To examine our architectural choices, we conducted ablation studies (see Appendix A, Section A.3, for more details).

7. Analysis and Discussion

The learning experiments provide evidence about the semantic relationships between citation analysis tasks. The hierarchical relationship between function and subfunction classification demonstrates significant synergistic effects, with joint learning yielding a substantial 11.4% improvement (21.1% relative improvement) for subfunction classification when conditioned on function predictions. This enhancement stems from the elimination of hierarchically inconsistent predictions, reducing the effective classification space and providing contextual guidance. The hierarchical consistency score of 83.3% indicates that the model successfully learns meaningful structural relationships between functional categories and their constituent subfunctions. This finding validates our design principles, demonstrating that the semantic hierarchy reflects genuine linguistic patterns in scholarly discourse rather than arbitrary categorization boundaries.

7.1. Task Interaction Patterns

The results show a clear distinction between tasks that benefit from joint learning and those that exhibit independence. Function predictions provide conditioning information that guides subfunction classification, while sentiment and importance exhibit orthogonal relationships to the hierarchical tasks, showing minimal beneficial interaction and, in some configurations, interference effects. This aligns with established work (Ruder, 2017), which emphasizes that successful joint learning depends on identifying tasks with genuinely beneficial semantic relationships. The conditioning mechanism reveals that function predictions serve as powerful contextual priors for subfunction classification, with implications for any hierarchically structured classification scheme in citation analysis.

The conditioning mechanism employed in hierarchical MTL reveals the significant impact of semantic structure on model performance. Function predictions serve as powerful contextual priors for subfunction classification, eliminating impossible category combinations and focusing the model's attention on semantically coherent options. This insight has implications beyond our specific taxonomy, suggesting that any hierarchically structured classification scheme in citation analysis could benefit from similar conditioning approaches.

7.2. Error Analysis and Boundary Ambiguities

Function Classification Confusion Patterns. The systematic analysis of function classification errors reveals consistent patterns of semantic ambiguity at taxonomic boundaries. Figure 4 presents

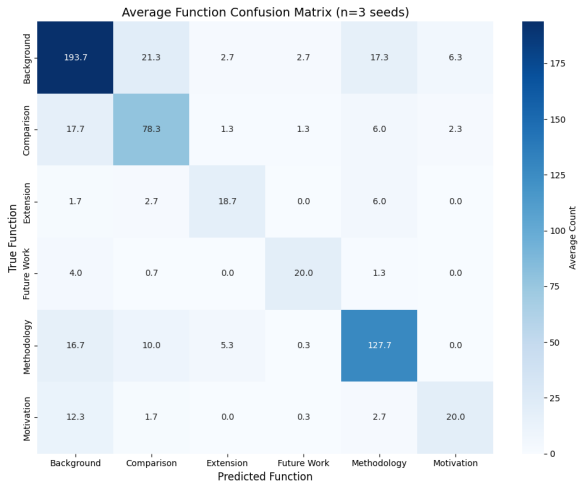


Figure 4: Average function confusion matrix across 3 random seeds.

the comprehensive confusion matrix for hierarchical MTL averaged across three random seeds, illustrating the distribution of classification errors across functional categories. The most prevalent confusion pairs demonstrate the inherent challenges in distinguishing between contextual establishment and comparative analysis in scholarly discourse. Background → Comparison confusion accounts for 21.3 ± 2.1 errors, while Comparison → Background confusion represents 17.7 ± 0.6 errors. Background → Methodology confusion shows 17.3 ± 1.2 errors. The confusion matrix visualization reveals the concentrated nature of these errors, with the majority of misclassifications occurring between semantically similar categories rather than exhibiting random distribution patterns. Citations serving dual purposes, such as "*Previous studies have shown...*" (labeled Background, predicted Comparison), illustrate the difficulty of imposing single-label constraints on multi-functional citations.

The error analysis reveals systematic bias toward majority classes: Background citations (39.8% of the dataset) show the strongest prediction bias, with 33.2% of true Motivation instances incorrectly predicted as Background.

Error Propagation in Hierarchical Systems. The most significant finding concerns error propagation in the hierarchical system. Function errors occur in 24.0% of instances, limiting subfunction performance: the conditioning mechanism leads subfunction predictions to categories within the predicted function, so function misclassification eliminates the possibility of selecting subfunctions from the correct function category. The subfunction confusion matrix shows 23.1% overall error rate across the hierarchical system. Given that function errors (24.0%) automatically propagate to subfunction errors, the conditioning architecture creates a bottleneck where upstream accuracy directly bounds downstream performance. When function predictions are correct, the model demonstrates strong subfunction classification capability, but this performance depends critically on accurate function-level decisions.

7.3. Class Imbalance Impact Assessment

Class imbalance effects demonstrate varying severity across different classification tasks, with the most dramatic impact observed in sentiment classification. The severe neutral class dominance (93.4% of instances) overwhelms the learning signal for positive and negative sentiment, resulting in systematic failure to learn minority class patterns. This imbalance creates a prediction distribution heavily skewed toward neutral classifications (95.5% of predictions), effectively rendering the sentiment classification task ineffective.

Sentiment Classification Failure: Sentiment classification demonstrates systematic failure across all configurations, achieving only 40.9% macro F_1 despite 93.0% accuracy. The model produces neutral predictions, with positive and negative sentiment achieving near-zero F_1 in multi-task configurations. Multi-task learning shows no improvement (F_1 = 0.325 across all phases), suggesting that academic writing's formal tone constrains evaluative expression to subtle contextual cues that current models cannot capture. Notably, LLMs outperform fine-tuned SciBERT on sentiment (GPT-3.5: 47.4% , GPT-4o mini: 46.3% vs. 40.9%), likely because their broader pretraining provides richer signal for nuanced evaluative language that domain-specific fine-tuning cannot recover under severe class imbalance.

Rare Category Performance: Rare subfunction categories exhibit variable performance depending on linguistic distinctiveness rather than sample frequency. *Future Work* achieves 76.9% accuracy with 26 test samples due to clear linguistic markers ("future work", "next steps"), while *Critique* shows 50.0% accuracy with only 4 samples despite lacking distinctive surface features. This indicates that category viability depends more on semantic coherence and linguistic distinctiveness than training data volume alone, though extremely small sample sizes (n<5) remain problematic regardless of feature clarity.

Mitigation Strategy Effectiveness: Loss function modifications show limited effectiveness for severe imbalance. Weighted cross-entropy provides modest improvements for moderate imbalance, while focal loss demonstrates meaningful but insufficient impact on subfunction classification. The hierarchical conditioning mechanism proves substantially

more effective, achieving 65.3% macro F_1 compared to 50.9% without conditioning (28.3% relative improvement). This architectural solution leverages semantic relationships rather than attempting optimization-based corrections alone.

8. Conclusions and Future Work

We established a unified taxonomic framework for multi-dimensional CCA and developed a comprehensive dataset of 4,017 citations, annotated by experts, integrating contemporary scholarly works with established schemes. Our MTL experiments demonstrate that citation analysis tasks exist within structured semantic hierarchies, with hierarchical conditioning providing substantial improvements for subfunction classification. The hierarchical multi-task learning approach achieved 65.3% $\pm$ 0.6% macro F_1 for subfunction classification, representing a 21.1% relative improvement over single-task approaches. The unified framework addresses critical temporal coverage limitations through contemporary annotation protocols while enabling systematic comparison across diverse taxonomic approaches. Future research should prioritize cross-dataset evaluation using external benchmarks and explore methodological extensions including soft parameter sharing and multi-label classification frameworks.

Limitations

Our evaluation focuses exclusively on the integrated dataset we developed, limiting claims about cross-dataset generalization and universal applicability across different academic fields. The single-label classification approach cannot capture citations that serve multiple functions simultaneously, with boundary ambiguities evident where citations establish context while making comparisons. Class imbalance presents persistent challenges despite mitigation efforts, with sentiment classification proving largely ineffective and rare categories showing poor performance with limited training examples. Additionally, the taxonomy does not include an 'Other' or catch-all category, which means every citation is assigned to one of the six defined functions. While this reflects our design goal of exhaustive functional coverage, it may force classification of ambiguous or multi-purpose citations into the closest available category rather than leaving them unclassified.

Acknowledgments

This work was supported by the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[6] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.). The consortium is funded by the Federal Republic of Germany and its states through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). This work was also co-funded by the EU Horizon Europe project SciLake (grant agreement 101058573).

9. Bibliographical References

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Yasa M. Baig, Alex X. Oesterling, Rui Xin, Haoyang Yu, Angikar Ghosal, Lesia Semenova, and Cynthia Rudin. 2021. Multitask learning for citation purpose classification. In *Proceedings of the Second Workshop on Scholarly Document Processing*, pages 134–139, Online. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Francisco Bolaños, Angelo Salatino, Francesco Osborne, and Enrico Motta. 2025. Modelling and classifying the components of a literature review. *arXiv preprint arXiv:2508.04337*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. 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 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Marilena Daquino, Silvio Peroni, and David Shotton. 2020. The opencitations data model. In *The Semantic Web – ISWC 2020*, volume 12507 of *Lecture Notes in Computer Science*, Cham. Springer.

[6] https://www.nfdi4datascience.de

Changxu Duan and Zhiyin Tan. 2026. Semantically orthogonal framework for citation classification: Disentangling intent and content. *arXiv preprint*.

Alex Fogelson, Ana Trišović, and Neil Thompson. 2025. Llms in citation intent classification: Progress, precision, and reproducibility challenges. In *Proceedings of the 3rd ACM Conference on Reproducibility and Replicability*, ACM REP '25, page 250–253, New York, NY, USA. Association for Computing Machinery.

Eugene Garfield, Mary Elizabeth Stevens, and Vincent E. Giuliano. 1964. Can citation indexing be automated? In *Statistical Association Methods for Mechanical Documentation, Symposium Proceedings*, volume 269, pages 189–196. National Bureau of Standards. Accessed August 4, 2025.

Myriam Hernández-Álvarez, José M. Gómez Soriano, and Patricio Martínez-Barco. 2017. Citation function, polarity and influence classification. *Natural Language Engineering*, 23(4):561–588.

Lasse M. Jantsch, Dong-Jae Koh, Seonghwan Yoon, Jisu Lee, Anne Lauscher, and Young-Kyoon Suh. 2025. FineCite: A novel approach for fine-grained citation context analysis. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24525–24542, Vienna, Austria. Association for Computational Linguistics.

X. Jiang. 2025. Ensembling approaches to citation function classification and important citation screening. *Scientometrics*, 130:1371–1419.

Charles Jochim and Hinrich Schütze. 2012. Towards a generic and flexible citation classifier based on a faceted classification scheme. In *Proceedings of COLING 2012*, pages 1343–1358, Mumbai, India. The COLING 2012 Organizing Committee.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Paris Koloveas, Serafeim Chatzopoulos, Thanasis Vergoulis, and Christos Tryfonopoulos. 2025. Can llms predict citation intent? an experimental analysis of in-context learning and fine-tuning on open llms.

Suchetha N. Kunnath, David Pride, and Petr Knoth. 2023. Prompting strategies for citation classification. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1127–1137.

Anne Lauscher, Brandon Ko, Bailey Kuehl, Sophie Johnson, Arman Cohan, David Jurgens, and Kyle Lo. 2022. MultiCite: Modeling realistic citations requires moving beyond the single-sentence single-label setting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1875–1889, Seattle, United States. Association for Computational Linguistics. Code and dataset available at `https://github.com/allenai/multicite`.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Michael J. Moravcsik and P. Murugesan. 1979. Citation patterns in scientific revolutions. *Scientometrics*, 1(2):161–169. Accessed August 5, 2025.

Roman Plaud, Matthieu Labeau, Antoine Saillenfest, and Thomas Bonald. 2024. Revisiting hierarchical text classification: Inference and metrics. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 231–242, Miami, FL, USA. Association for Computational Linguistics.

David Pride and Petr Knoth. 2020. An authoritative approach to citation classification. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, pages 337–340, New York, NY, USA. Association for Computing Machinery.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Carlos N. Silla and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2019. Which tasks should be learned together in multi-task learning? *arXiv preprint arXiv:1905.07553*. Presented at ICML 2020.

Xuan Su, Animesh Prasad, Min-Yen Kan, and Kazunari Sugiyama. 2019. Neural multi-task learning for citation function and provenance. In *Proceedings of the 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 394–395. IEEE.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM)*, pages 521–528. IEEE.

John Swales. 1986. Citation analysis and discourse analysis. *Applied Linguistics*, 7(1):39–56. Accessed August 4, 2025.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110, Sydney, Australia. Association for Computational Linguistics.

A. Appendix

A.1. Compatibility with Existing Schemes

Our framework integrates with existing classification schemes through structured mappings. The three established schemes we work with – SciCite (Cohan et al., 2019), MultiCite (Lauscher et al., 2022), and Jurgens-CFC (Jurgens et al., 2018) – align well with our Level 1 categories, allowing direct automated conversion. Citation functions appear across taxonomies with similar meanings, though terminology differs. Level 2 subcategories provide finer distinctions not always present in existing schemes, requiring manual annotation while maintaining broad Level 1 compatibility. This design allows researchers to use existing datasets while gaining access to more detailed analysis. Direct mappings handle 89% of conversions, with only MultiCite's "Similarities" and "Differences" requiring merging into our COMPARISON category.

This approach allows researchers to work with a unified framework while preserving insights from existing classification schemes. The high proportion of direct mappings shows that our taxonomy captures core distinctions that the field has developed, while the hierarchical structure provides additional analytical dimensions not available in individual existing approaches.

A.2. Experimental Configuration

All experiments were conducted on Google Colab with NVIDIA Tesla T4 GPU (16GB VRAM). All three random seeds (42, 123, 456) completed successfully. Single-task experiments required 5.2–8.8 minutes per task with consistent GPU utilization (75–77% mean) and peak memory usage of 3.2–3.6 GB. Hierarchical MTL trained in 24.8 minutes using only 2.0 GB peak memory (45% reduction), though with lower mean GPU utilization (15%) due to early stopping dynamics. Progressive MTL required the longest training time at 108.7 minutes total across three sequential phases, with 3.4 GB peak memory and 66% mean GPU utilization. Table 4 presents the complete hyperparameter configuration used across all experiments.

Category	Parameter	Value
Model Architecture		
Base Model	SciBERT	allenai/scibert_scivocab_uncased
Max Sequence Length	tokens	256
Hidden Size	dimensions	768
Dropout	Hierarchical MTL	0.1
	Progressive MTL	0.3
Training Configuration		
Batch Size	Function	16
	Subfunction	12
	Sentiment	16
	Importance	16
	Hierarchical MTL	8
	Progressive MTL	16 (GPU)
Learning Rate	Function	1.5e-5
	Subfunction	1e-5
	Sentiment	1e-5
	Importance	1e-5
	Hierarchical MTL	2e-5
	Progressive MTL	2e-5
Optimizer		AdamW
Weight Decay		0.01
Warmup Ratio	Function	0.05
	Other tasks	0.1
Max Epochs	Function	6
	Subfunction	10
	Sentiment	8
	Importance	8
	Hierarchical MTL	10
	Progressive MTL	Variable by phase
Early Stopping Patience	Function	4
	Subfunction	4
	Sentiment	3
	Importance	4
	Hierarchical/Progressive	3
Gradient Clipping	max norm	1.0
Loss Functions		
Function Classification		Weighted Cross-Entropy
Subfunction (Single-task)		Weighted Cross-Entropy
Subfunction (MTL)		Focal Loss ($\gamma = 2.0$)
Sentiment Classification		Focal Loss ($\gamma = 2.0 - 3.0$)
Importance Classification		Weighted Cross-Entropy
Multi-Task Learning Weights		
Progressive Phase 1		Function: 0.7, Subfunction: 0.3
Progressive Phase 2		Func: 0.5, Sub: 0.3, Imp: 0.2
Progressive Phase 3		Func: 0.4, Sub: 0.3, Imp: 0.2, Sen: 0.1
Hierarchical MTL		Function: 0.7, Subfunction: 0.3
Data Configuration		
Training Set		2,814 citations (70%)
Validation Set		602 citations (15%)
Test Set		601 citations (15%)
Random Seeds		[42, 123, 456]
Stratified Sampling		Yes (by citation_id)
Hierarchical MTL Architecture		
Conditioning Mechanism		Concatenation
Function Logits Dimension		6
Function Embedding Dim		192 (hidden_dim/4)
Combined Feature Dim		960 (768 + 192)
Use Function Conditioning		True

Table 4: Hyperparameters for all experimental configurations.

A.3. Ablation Study

Removing function conditioning reduces subfunction performance to 50.9% macro F_1, demonstrating the critical importance of hierarchical structure. Full hierarchical MTL with conditioning achieves 65.3%, representing a 14.4% absolute improvement. Replacing focal loss with standard cross-

entropy yields 56.6% macro F_1, showing an 8.7% benefit from focal loss adaptation for class imbalance. The conditioning mechanism proves far more effective than loss function modifications, emphasising the importance of leveraging known semantic relationships rather than purely optimization-based approaches.

XplaiNLP @ ClimateCheck 2026 Task 2: Comparing Hierarchical Approaches for Fine-Grained Climate Disinformation Narrative Classification

Arthur Hilbert[1,3], **Jing Yang**[1,2,4], **Vera Schmitt**[1,2,3,4]
[1]Technische Universität Berlin
[2]BIFOLD – Berlin Institute for the Foundations of Learning and Data
[3]German Research Center for Artificial Intelligence (DFKI)
[4]Centre for European Research in Trusted AI (CERTAIN)
{arthur.hilbert, jing.yang, vera.schmitt}@tu-berlin.de

Abstract

We present our submission to Task 2 of the ClimateCheck 2026 shared task on Disinformation Narrative Classification which requires assigning climate-contrarian claims to fine-grained disinformation narratives. Using `Qwen3-8B` as a fixed backbone, we systematically compare **data augmentation**, **prompt engineering** and **reinforcement learning**. Our experiments show that structured reasoning, particularly a chain-of-thought (CoT) prompting strategy aligned with the taxonomy's hierarchical structure, substantially improves Macro-F1 over both zero-shot baselines and augmentation-based fine-tuning. Our best configuration achieves ∼**0.625 Macro-F1**, ranking first in Task 2. Our findings demonstrate that carefully designed hierarchical prompting can rival more complex training interventions in low-resource, highly imbalanced narrative classification settings.

Keywords: disinformation, climate change, narrative classification

1. Introduction

As digital platforms play an increasingly central role in shaping perceptions of scientific issues, there is a pressing need for NLP systems that can identify and contextualize questionable claims and connect them reliably to evidence. We present our contribution to Task 2 of Abu Ahmad et al. (2026)'s ClimateCheck 2026 shared task[1] on **Disinformation Narrative Classification**. The shared task addressed the growing challenge of climate-related discourse on social media, where increased public engagement is accompanied by the spread of misleading narratives that emerge, adapt, and recombine over time within online information ecosystems (Abu Ahmad et al., 2025b,a).

The 2026 edition of ClimateCheck combines the previous iteration's tasks into Task 1: Abstract Retrieval and Claim Verification and adds a new Task 2: Disinformation Narrative Classification. However, we only participated only in Task 2 of the shared task, which was centered around assigning claims to fine-grained disinformation narratives derived from the CARDS taxonomy of contrarian climate claims (Coan et al., 2021; Rojas et al., 2024). This problem setting is technically challenging due to overlapping narrative themes, imbalanced label distributions, and a scarcity of data (Abu Ahmad et al., 2025b). In the official evaluation, systems were ranked by Macro-F1 rewarding the accurate classification of rare labels. Our best submission achieved the first place in Task 2 with a score of ∼**0.625 Macro-F1**.

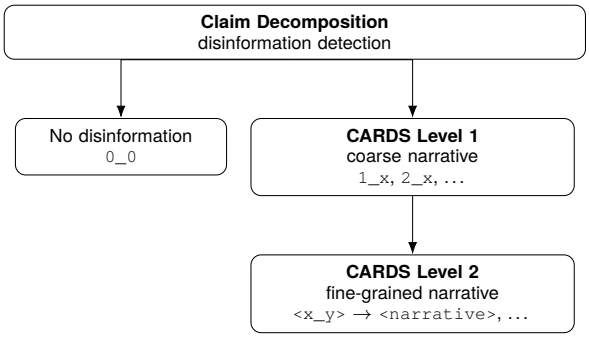


Figure 1: Our three-step hierarchical narrative classification approach.

Rather than varying model families, we focused on isolating the effects of different approaches by using a `Qwen3-8B` model (Yang et al., 2025) as the starting point across experiments. This choice allowed for direct comparison to the task baseline, which performed supervised finetuning on `Qwen3-8B` using Low-Rank Adaptation (LoRA)(Hu et al., 2021). The baseline was provided by team EFC, ClimateCheck 2025's 3rd place submission to the Claim Verification task (Upravitelev et al., 2025). Following the optional shared-task guidance, we tracked the inference emissions of our experiments using CodeCarbon (Courty et al., 2024).

We explore three main directions: **data augmentation**, **prompt engineering**, and **reinforcement learning** using Group-Relative Policy Optimization (GRPO) (Shao et al., 2024). For prompt de-

[1]https://github.com/Dagobert42/climatecheck2026-task2-hierarchical-approaches

Proceedings of the 3rd Int. Workshop on Natural Scientific Language Processing (NSLP 2026) @ LREC 2026, pages 289–296
12 May 2026.

sign, we investigate *hierarchical prompting* (Singh et al., 2025) and an off-shoot *chain-of-thought (CoT) prompting* (Wei et al., 2023), leveraging the layered structure of the CARDS taxonomy. Our best-performing configuration was achieved with a **CoT prompting** strategy on the vanilla `Qwen3-8B`.

Related Work. ClimateCheck 2026 Task 1 bears much resemblance to the AVeriTeC shared task, where participants must handle evidence retrieval and veracity prediction (Schlichtkrull et al., 2023, 2024). Thus, we searched for comparable shared tasks for Task 2. Our solution leans on the results from SemEval Task 10 Subtask 2, which featured a narrative classification setting within the PolyNarrative dataset (Nikolaidis et al., 2025), but focused on online news articles as opposed to claims (Piskorski et al., 2025). Teams explored approaches from simple vectorization techniques (Blombach et al., 2025) and cross-lingual knowledge transfer (Eleftheriou et al., 2025) to zero-shot agentic frameworks (Eljadiri and Nurbakova, 2025), with prompt- and fine-tuning-based approaches being omnipresent (Piskorski et al., 2025). The leading submission by team GateNLP used data augmentation and introduced a Hierarchical Three-Step Prompting (H3Prompt) approach (Singh et al., 2025). This aligns with earlier work on climate disinformation which also successfully explored data augmentation and hierarchical models (Piskorski et al., 2022; Rowlands et al., 2024; Rojas et al., 2024). Previously overlooked seems to have been the application of reinforcement learning, which has successfully been applied to mathematics (Yang et al., 2024; Wang et al., 2024) and other reasoning-intensive tasks (Suma and Dauncey, 2025).

2. Task Description.

Given a claim c and a predefined taxonomy of climate disinformation narratives $\mathcal{Y} = \{y_1, \ldots, y_M\}$, the task is to predict the subset of labels $\hat{\mathcal{Y}}_c \subseteq \mathcal{Y}$ that apply to c. This is a multi-class, multi-label classification problem, where each claim may be associated with multiple narratives. The goal is to approximate the gold label set $\mathcal{Y}_c^*$ for each claim.

This is commonly modeled as a multi-stage hierarchical classification (Singh et al., 2025; Eleftheriou et al., 2025), where each step is meant to narrow down the prediction to an increasingly reduced set of labels.

Dataset. The dataset at hand is highly imbalanced and comprised a label set of 33 narratives. The training split contained 763 distinct claims, while the test split consisting of 172 claims was used exclusively for submissions. Each datapoint consists of a claim with ID and its narrative label(s).

Narrative Label		Count
0_0	No disinformation narrative	556
5_1	Climate science is uncertain	44
2_1	It's natural cycles/variation	38
...		
1_8	Doesn't impact health	1
2_2	It's non-greenhouse gas forcings	1
1_5	Oceans are not warming	1

Table 1: Most and least frequent narratives in the train dataset.

3. Methods

We compare approaches pertaining to prompting and fine-tuning large language models (LLM). Models were sourced through the `Transformers` library (Wolf et al., 2020) and fine-tuned using `Unsloth` (Daniel Han and team, 2023).

Data Augmentation. Following Singh et al. (2025), we perform targeted data augmentation using (`qwen3-30b-a3b-instruct-2507`) to mitigate severe class imbalance and enrich rare narratives. We first apply a skewed inverse-frequency reweighting to oversample minority narratives from the training set. Given the inverse-frequency w_i for each label, we obtain its sampling probability p_i via normalization and apply a smoothing factor α to maintain some of the original distribution:

$$p_i = \frac{w_i^\alpha}{\sum_{j=1}^{N} w_j^\alpha} \qquad \text{with } \alpha = 0.5$$

With this we sample a list of 1,000 augmentation candidates with repetition and subsequently employ a two-step prompting strategy to improve augmentation quality. Initially, a model is asked to generate a concise explanation of why a candidate claim fits its assigned narrative label(s). Secondly, given narrative labels and the generated explanation, the model produces 10 new synthetic claims for the same narrative label(s). We sampled 10,000 synthetic claims under this procedure, which were subsequently used in a supervised pre-training objective.

We perform parameter-efficient supervised fine-tuning of using LoRA adapters with rank $r = 16$, $\alpha = 16$. The original training data is split further into an 80% train and a 20% validation set. The model is pre-trained on the augmentation data with the validation loss as an early stopping criterion at a patience of 3. Then the model is further fine-tuned using the same criterion on the 80% train split. Finally, we perform a short 2-epoch tuning on the validation split to get maximum mileage out of the limited train data.

Experiment	Macro-F1	Macro-P	Macro-R	Micro-F1	Weighted-F1
Baseline	51.36	52.99	57.37	79.78	78.44
Zero-shot	42.85	45.51	50.98	74.51	73.54
Zero-shot + CoT	38.24	39.59	46.50	68.38	69.33
Zero-shot + H	43.13	39.37	58.18	58.01	62.50
Zero-shot + Reasoning	56.19	61.46	58.23	84.18	80.75
Zero-shot + CoT + Reasoning	**62.48**	**70.71**	**63.11**	**84.42**	82.06
A + FT	54.18	55.18	57.96	84.33	**82.20**
H + A + FT	54.22	57.97	56.84	83.57	81.86
RL	57.20	64.91	59.60	84.33	81.68
CoT + RL	60.47	70.21	58.80	83.05	81.56

Table 2: Performance metrics across experiments using `Qwen3-8B` as base model. All values in %. Best results (highest) per metric are highlighted in **bold**. *Baseline* results were provided by organizers via fine-tuning `Qwen3-8B` on task training data. Zero-shot = no additional training was applied, A = additional pre-training using augmented data, FT = fine-tuning on original training data, H = hierarchical prompting, CoT = chain-of-thought prompting, RL = additional fine-tuning with reinforcement learning using GRPO.

Prompt Engineering. We compare three prompting strategies:

(1) Simple A direct instruction-based prompt asks the model to classify a claim according to the taxonomy without explicitly structuring the reasoning process.

(2) Hierarchical The classification is decomposed into a two-turn conversation, where first, the model identifies the high-level narrative group(s). Second, conditioned on this prior selection, it predict the fine-grained sub-label(s). This explicitly operationalizes hierarchical classification.

(3) CoT To arrive at its prediction the model is instructed to follow an implicit hierarchical process with three steps: (i) extracting the core assertion(s) and separating non-disinformation claims, (ii) selecting the appropriate high-level narrative group (1_x to 5_x), and (iii) choosing the most specific sub-label. This encourages hierarchical reasoning while maintaining a single forward pass.

We provide an overview of the exact prompt templates in the Appendix A.

GRPO Fine-Tuning. We experiment with further reasoning optimizations using GRPO as implemented in the `TRL` library (von Werra et al., 2020). The base model was prepared using LoRA (Hu et al., 2021) with rank $r = 32$, $\alpha = 32$. We train for 250 steps with a learning rate of 5×10^{-6} and 8 sampled generations per prompt.

Claims for GRPO fine-tuning are sampled using the same skewed inverse-frequency reweighting as for data augmentation (with $\alpha = 0.5$). GRPO uses a weighted and signed reward function, which simply assigns the inverse class-weight for correct predictions and an equal negative weight for incorrect predictions. In addition, a brevity reward penalizes overly long generations beyond 250 words to prevent excessive reasoning.

4. Results

We report classification metrics and environmental impact for all evaluated approaches on Task 2. Table 2 summarizes predictive performance.

Experiment	Time	Emissions	Energy
	s	$kgCO_2eq$	kWh
ZS	187	0.0053	0.0138
ZS + CoT	**119**	0.0055	0.0145
ZS + H	700	0.0352	0.0924
ZS + Reasoning	4838	0.1489	0.3910
ZS + CoT + Reasoning	2438	0.1296	0.3403
A + FT	206	0.0090	0.0237
H + A + FT	160	**0.0045**	**0.0118**
RL	2458	0.1261	0.3310
CoT + RL	2541	0.1229	0.3226

Table 3: Emission statistics for the evaluated experiments at inference time. Best (lowest) values are in **bold**. ZS = Zero-Shot setting.

Table 3 provides the corresponding inference-time environmental statistics, including duration (in seconds), estimated emissions ($kgCO_2eq$), and energy consumption (kWh), measured using CodeCarbon under identical hardware conditions.

Figure 2 visualizes the trade-off between predictive performance (Macro-F1) and computational cost (inference duration), mostly induced by use of reasoning capabilities.

5. Discussion

Performance Trends. Across all approaches, structured reasoning substantially improves zero-shot performance. In terms of Macro-F1, naive zero-shot approaches lag considerably behind fine-tuning and structured reasoning methods. Intro-

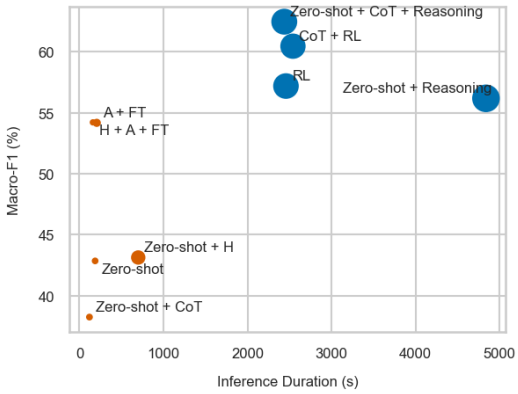


Figure 2: Macro-F1 (%) versus inference duration (s). Blue denotes reasoning-based approaches, orange non-reasoning approaches. Point size proportionally indicates emissions in $\mathrm{kgCO_2eq}$.

ducing explicit reasoning steps yields consistent gains, with the highest overall Macro-F1 achieved for CoT prompting with reasoning. This suggests that guided intermediate reasoning helps the model better differentiate fine-grained narrative categories, particularly in a highly imbalanced multi-label setting.

Hierarchical prompting as an intermediate step between simple zero-shot and reasoning methods does not yield significant improvements. Nor does its augmented and fine-tuned variant deviate in any significant way from non-hierarchical augmentation and fine-tuning. Augmented fine-tuning improves over the fine-tuning baseline and yields strong Weighted-F1, reflecting improved alignment with the dominant class. However, Macro-F1 gains remain moderate compared to reasoning-based zero-shot methods. This indicates that synthetic data improves robustness but fails to resolve minority-class discrimination.

Reinforcement learning via GRPO improves reasoning performance for the naive prompt setting but surprisingly falls short when using the CoT-prompt variant. These results remain non-conclusive on how reward shaping with class-sensitive weighting affects minority class predictions.

Efficiency vs Performance Trade-off. The environmental analysis reveals a clear and expected trade-off between predictive quality and computational cost. Pure zero-shot and lightweight prompting strategies are highly efficient but underperform in Macro-F1. In contrast, reasoning-heavy approaches dramatically increase inference time and emissions. Notably, augmentation with subsequent fine-tuning achieves competitive performance with comparatively low emissions. Figure 2 illustrates this Pareto-like frontier.

Error Patterns. As the test set is not made public we cannot provide any confusion matrices. However, minority labels with extremely low training frequency remain challenging across all methods, indicating that imbalance persists even after augmentation and reward weighting.

Limitations. First, Macro-F1 remains sensitive to extremely rare labels, some of which contain only one or two instances. Second, reasoning-based methods incur substantial computational overhead, which inhibits performance comparison across multiple runs to show how stable (or not) our experimental results are. Third, synthetic augmentation relies on model-generated claims which introduces distributional artifacts that do not fully reflect real-world disinformation patterns. Finally, reinforcement learning rewards are label-exact and do not explicitly account for partial or hierarchical correctness, potentially underestimating semantically close predictions.

Overall, our results highlight the importance of structured reasoning and class-aware optimization for fine-grained narrative classification, while underscoring the need to balance predictive gains with environmental and computational cost.

6. Conclusion

We presented our contribution to Task 2 of ClimateCheck 2026 on Disinformation Narrative Classification. Using `Qwen3-8B` as a fixed backbone, we systematically compared data augmentation, prompt engineering, and reinforcement learning. Our results demonstrate that structured reasoning, particularly using a CoT-based prompting strategy, improves Macro-F1, ultimately achieving first place in the shared task with $\sim$0.625 Macro-F1.

Beyond predictive performance, we evaluated the environmental cost of each method. Our findings contextualize a stark trade-off between reasoning-based gains and computational overhead, highlighting the importance of reporting emissions alongside accuracy in shared tasks. While augmentation and GRPO can provide additional improvements, they do not consistently outperform carefully constructed prompting strategies in this low-resource setting.

Future work should explore the stability of predictive performance across multiple inference runs, hybrid reward formulations to account for hierarchical correctness, and techniques for reducing the computational footprint of reasoning-heavy approaches.

Acknowledgements

The work on this paper is performed in the scope of VeraXtract (16IS24066) funded by the German Federal Ministry for Research, Technology and Aeronautics (BMFTR).

Bibliographical References

Raia Abu Ahmad, Max Upravitelev, Aida Usmanova, Veronika Solopova, and Georg Rehm. 2026. ClimateCheck 2026: Scientific Fact-Checking and Disinformation Narrative Classification of Climate-related Claims. In *Proceedings of the 3rd International Workshop on Natural Scientific Language Processing (NSLP 2026)*, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025a. The ClimateCheck dataset: Mapping social media claims about climate change to corresponding scholarly articles. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 42–56, Vienna, Austria. Association for Computational Linguistics.

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025b. The ClimateCheck shared task: Scientific fact-checking of social media claims about climate change. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 263–275, Vienna, Austria. Association for Computational Linguistics.

Andreas Blombach, Bao Minh Doan Dang, Stephanie Evert, Tamara Fuchs, Philipp Heinrich, Olena Kalashnikova, and Naveed Unjum. 2025. Narrlangen at SemEval-2025 task 10: Comparing (mostly) simple multilingual approaches to narrative classification. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2240–2248, Vienna, Austria. Association for Computational Linguistics.

Travis G. Coan, Constantine Boussalis, John Cook, and Mirjam O. Nanko. 2021. Computer-assisted classification of contrarian claims about climate change. *Scientific Reports*, 11(1):22320.

Benoit Courty, Victor Schmidt, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, SabAmine, inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Amine Saboni, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, alencon, Michał Stęchły, Christian Bauer, Lucas-Otavio, JPW, and MinervaBooks. 2024. mlco2/codecarbon: v2.4.1.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Konstantinos Eleftheriou, Panos Louridas, and John Pavlopoulos. 2025. KostasThesis2025 at SemEval-2025 task 10 subtask 2: A continual learning approach to propaganda analysis in online news. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 899–908, Vienna, Austria. Association for Computational Linguistics.

Mohamed Nour Eljadiri and Diana Nurbakova. 2025. Team INSALyon2 at SemEval-2025 task 10: A zero-shot agentic approach to text classification. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 965–980, Vienna, Austria. Association for Computational Linguistics.

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

Nikolaos Nikolaidis, Nicolas Stefanovitch, Purificação Silvano, Dimitar Iliyanov Dimitrov, Roman Yangarber, Nuno Guimarães, Elisa Sartori, Ion Androutsopoulos, Preslav Nakov, Giovanni Da San Martino, and Jakub Piskorski. 2025. PolyNarrative: A multilingual, multilabel, multidomain dataset for narrative extraction from news articles. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31323–31345, Vienna, Austria. Association for Computational Linguistics.

Jakub Piskorski, Tarek Mahmoud, Nikolaos Nikolaidis, Ricardo Campos, Alipio Mario Jorge, Dimitar Dimitrov, Purificação Silvano, Roman Yangarber, Shivam Sharma, Tanmoy Chakraborty, Nuno Guimaraes, Elisa Sartori, Nicolas Stefanovitch, Zhuohan Xie, Preslav Nakov, and Giovanni Da San Martino. 2025. SemEval 2025 task 10: Multilingual characterization and extraction of narratives from online news. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2610–2643, Vienna, Austria. Association for Computational Linguistics.

Jakub Piskorski, Nikolaos Nikolaidis, Nicolas Stefanovitch, Bonka Kotseva, Irene Vianini, Sopho Kharazi, and Jens P. Linge. 2022. Exploring data

augmentation for classification of climate change denial: Preliminary study. In *Proceedings of Text2Story - Fifth Workshop on Narrative Extraction From Texts held in conjunction with the 44th European Conference on Information Retrieval (ECIR 2022), Stavanger, Norway, April 10, 2022*, volume 3117 of *CEUR Workshop Proceedings*, pages 97–109. CEUR-WS.org.

Cristian Rojas, Frank Algra-Maschio, Mark Andrejevic, Travis Coan, John Cook, and Yuan-Fang Li. 2024. Hierarchical machine learning models can identify stimuli of climate change misinformation on social media. *Communications Earth & Environment*, 5(1):436.

Harri Rowlands, Gaku Morio, Dylan Tanner, and Christopher Manning. 2024. Predicting narratives of climate obstruction in social media advertising. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5547–5558, Bangkok, Thailand. Association for Computational Linguistics.

Michael Schlichtkrull, Yulong Chen, Chenxi Whitehouse, Zhenyun Deng, Mubashara Akhtar, Rami Aly, Zhijiang Guo, Christos Christodoulopoulos, Oana Cocarascu, Arpit Mittal, James Thorne, and Andreas Vlachos. 2024. The automated verification of textual claims (AVeriTeC) shared task. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 1–26, Miami, Florida, USA. Association for Computational Linguistics.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.

Iknoor Singh, Carolina Scarton, and Kalina Bontcheva. 2025. Gatenlp at semeval-2025 task 10: Hierarchical three-step prompting for multilingual narrative classification.

Adam Suma and Samuel Dauncey. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Max Upravitelev, Nicolau Duran-Silva, Christian Woerle, Giuseppe Guarino, Salar Mohtaj, Jing Yang, Veronika Solopova, and Vera Schmitt. 2025. Comparing LLMs and BERT-based classifiers for resource-sensitive claim verification in social media. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*, pages 281–287, Vienna, Austria. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement.

A. Appendix

We provide the exact prompt templates used in our experiments for data augmentation and prompt-based classification.

Explanation Prompt This prompt generates a concise explanation of why a claim fits specific disinformation narratives. It is used as an intermediate step in the data augmentation pipeline to improve the quality of synthetic samples.

```
Your task is to concisely explain why
    the following claim fits specific
disinformation narrative(s) on climate
    change.

Claim: {claim}
Narrative IDs: {narrative_codes}
Narrative descriptions:
- {narrative_descs}

Provide a short reasoning of why the
    claim caters to the specific
narrative(s). Do not repeat the claim
    and output nothing else.
Start directly with the explanation and
    limit your response to one
    paragraph.
```

Augmentation Prompt This prompt generates synthetic training examples conditioned on a claim, its narrative labels, and an explanation. It is used to create additional labeled data for fine-tuning.

```
Your task is to augment training data
    for a classifier to combat climate
change disinformation. Claims in the
    dataset are categorized by the
following taxonomy:

{taxonomy}

Example claim: {claim}
Narrative ID: {narrative_codes}
Narrative descriptions:
    {narrative_descs}
Explanation: {explanation}

Generate 10 new claims that cater ONLY
    to the described disinformation
narrative(s) on climate change.
Each new claim must be distinct and
    consistent with the narrative(s).
Do not include any extra keys,
    commentary, emojis or markdown.

Output ONLY a JSON list in the
    following format:
[
  {"claim": "...", "narrative": "..."},
  ...
]
```

Simple Classification Prompt This prompt performs direct zero-shot classification without explicitly guiding the reasoning process.

```
You are an expert in detecting climate
    change related disinformation.

Your task is to classify the claim
    using the provided taxonomy.

Taxonomy:
{taxonomy}

Rules:
- Output ONLY a valid JSON list of
    strings.
- Each item MUST be exactly "key ->
    value".
- If no disinformation is found, output:
  ["0_0 -> No disinformation narrative"]

Claim: "{claim}"
Output:
```

Basic Reasoning Prompt This prompt performs direct classification without explicitly guiding intermediate reasoning steps. It serves as a lightweight baseline for zero-shot classification.

```
You are an expert in detecting climate
    change related disinformation.
You get a claim and your task is to
    classify the claim using the
    taxonomy
codes provided.

Rules:
- Output ONLY a valid JSON list of
    strings.
- Each item MUST be exactly "key ->
    value" where key is a taxonomy code
  and value is its narrative name.
- If no disinformation is found, output:
  ["0_0 -> No disinformation narrative"]
- Do not output anything else.

Taxonomy:
{taxonomy_str}

Claim: "{claim}"
Output:
```

CoT-style Reasoning Prompt This prompt introduces structured reasoning by guiding the model through a three-step process: extracting claims, identifying high-level narratives, and selecting fine-grained labels.

```
You are an expert in detecting climate
    change related disinformation.

Your task is to classify the claim
    using the provided taxonomy.
```

```
Taxonomy:
{taxonomy}

Follow this 3-step reasoning process
    internally:

Step 1 - Extract the core assertion(s)
Step 2 - Identify the high-level
    narrative group
Step 3 - Choose the most specific
    sub-label

Output ONLY a valid JSON list of
    strings.

Claim: "{claim}"
Output:
```

Hierarchical Prompt (Level 1) This prompt represents the first stage of hierarchical classification, where the model predicts coarse narrative groups.

```
You are an expert in detecting climate
    change related disinformation.

Your task is to classify the claim
    using the provided taxonomy.

Taxonomy:
{taxonomy}

Step 1 - Extract the core assertion(s)
Step 2 - Identify the high-level
    narrative group(s)

Claim: "{claim}"
Output:
```

Hierarchical Prompt (Level 2) This prompt represents the second stage of hierarchical classification, where the model refines predictions into fine-grained narratives.

```
Now complete step 3 using your answers
    from step 2:

Step 3 - Choose the most specific
    sub-label

Output ONLY a valid JSON list of
    strings.

Claim: "{claim}"
Output:
```

Author Index