

# HPC-ED: Testing Automated Agents to Assess the Quality of Training Resource Metadata

Habiba Morsy  
Kean University  
University of Virginia  
morsyh@kean.edu

Essence Toone  
Kean University  
toonee@kean.edu

Charlie Dey  
Texas Advanced Computing Center  
charlie@tacc.utexas.edu

Zilu Wang  
Cornell University Center for  
Advanced Computing  
zw427@cornell.edu

Mary Thomas  
University of California, San Diego  
mpthomas@ucsd.edu

David Joiner  
Kean University  
djoiner@kean.edu

## ABSTRACT

We present a proof-of-concept system for automating quality assurance in the HPC-ED federated training catalog using large language models (LLMs). The HPC-ED catalog system integrates metadata crawling, video transcript extraction, and model-based evaluation to score and provide recommendations on metadata quality at scale.

## KEYWORDS

HPC, Metadata Quality, Large Language Models

## 1 INTRODUCTION

### 1.1 The HPC-ED Catalog

The HPC-ED project collects item-level metadata for training objects in partner catalogs and digital libraries to enhance content discovery for the HPC and broader cyberinfrastructure (CI) communities. Existing projects have produced a wide range of training materials, including tutorials, workshops, course modules, and recorded events. Surveys of HPC training providers and users have consistently identified difficulties in finding resources at the right depth, filtering results, and identifying trusted sources [1].

To address these issues, the **HPC-ED** project was launched to improve discovery, sharing, and reuse of HPC and CI training resources through a *federated training catalog* architecture. The catalog allows providers to retain ownership while publishing standardized metadata to a shared repository [8]. Using an established search backend (Globus Search) and a minimal metadata set aligned with FAIR principles [5, 7], HPC-ED enables both centralized and embedded discovery interfaces [10]. Since its inception, HPC-ED has completed a **pilot phase** demonstrating the feasibility of the federated approach, including:

- **Metadata framework:** Adopting and extending community metadata standards for training materials, including

the Research Data Alliance’s recommended minimal set and Dublin Core Learning Resource Type [3, 7].

- **Architecture prototype:** Share and discovery workflow using Globus Search with JSON-formatted records [6].
- **Community** – Conducting workshops, tutorials, and hackathons to gather feedback and grow early adopters [8].
- **Planned scaling improvements** – Designing a decentralized “pull-model” architecture for production deployment, integrating quality checks and diverse client tools [10].

### 1.2 Quality Assurance Automation

While metadata publication processes in the HPC-ED project have matured, **quality assurance (QA)** remains largely manual, relying on human review and link checking. Given the scale of the catalog and the dynamic nature of online resources, automating QA is critical for sustaining trust, maintaining relevance, and enabling scalable federation.

In this paper, we present a proof of concept project that investigates the use of commercial LLMs to automate QA recommendations on metadata quality. Our approach has been to start with a curated subset of the catalog and perform a 2 stage AI assisted review of submitted metadata. The first crawls the URL provided in the metadata to summarize the content, while the second uses an agent to review the metadata in comparison to the summary.

## 2 METHOD

### 2.1 Overview

The HPC-ED Beta catalog, as of March 1, 2025, was used as the test set. It contained 128 entries, 5 of which included a YouTube video, either linked or embedded, within a crawl depth of two.

Each item was parsed using BeautifulSoup with Python’s built-in `html.parser` [9]. A custom crawler scraped text and URLs from each entry’s root metadata URL, descending to a maximum depth of two. For YouTube links, the `youtube_transcript_api` library extracted transcripts [2].

All crawled text was summarized using GPT-3.5 Turbo. A separate review agent, implemented via OpenAI API calls, assessed metadata alignment and provided scoring with rationale. This agent was evaluated across four models: GPT-3.5 Turbo, GPT-4o Mini, GPT-4.1 Nano, and GPT-4.1 Mini.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2026 Journal of Computational Science Education  
<https://doi.org/10.22369/issn.2153-4136/17/1/5>

For each item in the test set, all crawled text was passed to a summarization agent using OpenAI GPT 3.5 turbo. Summaries were stored for each item in the test set for further analysis.

A separate review agent was developed to compare the submitted metadata for each entry against both the catalog description and the crawled summary. The agent produced a score assessing the alignment between the submitted metadata and the content, along with a brief rationale. This review agent was implemented using OpenAI API calls and tested with four models: GPT-3.5 Turbo, GPT-4o Mini, GPT-4.1 Nano, and GPT-4.1 Mini.

## 2.2 Metadata Review Agent

To evaluate metadata quality, we developed an automated review pipeline implemented in Python. After crawling each catalog entry (depth  $\leq 2$ ) and summarizing its content using gpt-3.5-turbo, we passed the original metadata and summary to a validation agent. The agent was prompted to rate the accuracy of each metadata field, provide reasoning, and, if necessary, suggest improvements. We enforced structured responses in strict JSON format for downstream parsing. Below is the exact prompt used:

```
prompt = f"""
Evaluate the accuracy of the keywords metadata tag on a scale of 1-5
(1 = incorrect, 5 = fully accurate).
If the score is 2 or lower, suggest a better list of keywords based on
the abstract and summary.
```

```
Metadata:
Title: {metadata.get('Title', '')}
URL: {metadata.get('URL', '')}
Keywords: {metadata.get('Keywords', '')}
Abstract: {metadata.get('Abstract', '')}
Summary: {metadata.get('Summary', '')}
Duration: {metadata.get('Duration', '')}
Learning Resource Type: {metadata.get('Learning Resource Type', '')}
Expertise Level: {metadata.get('Expertise Level', '')}
```

Respond in this JSON format:

```
{
  "Keywords": {
    "score": X,
    "reason": "reason here",
    "suggested_keywords": ["keyword1", "keyword2", "keyword3"]
  }
}
```

A representative output (simplified for brevity) is shown below. (Note that the AI agent treated the additional GPT 3.5 summary as a catalog metadata item to review instead of as ancillary material to guide the review – this will be modified in future versions.)

```
{
  "Title": {
    "score": 4,
    "reason": "The Title field is filled and descriptive, but lacks
    specific details."
  },
  "URL": {
    "score": 3,
    "reason": "The URL is included but should ideally point to the
    specific resource."
  },
  "Keywords": {
    "score": 2,
    "reason": "Keywords are missing, which are essential for
    searchability and categorization."
  },
  "Abstract": {
    "score": 4,
    "reason": "The Abstract provides a concise summary but
```

```
could be more detailed."
  },
  "Summary": {
    "score": 3,
    "reason": "Summary is similar to an abstract; score
    could be higher if it's distinct and adds value."
  },
  "Duration": {
    "score": 1,
    "reason": "Duration is missing, which is crucial for
    learners to estimate time commitment."
  },
  "Learning Resource Type": {
    "score": 5,
    "reason": "Clearly specifies the type of learning resource,
    well-defined."
  },
  "Expertise Level": {
    "score": 4,
    "reason": "Provides guidance on the expected expertise
    level of the audience, but could be more detailed."
  }
}
```

This structured output was parsed and appended to the dataset, allowing quantitative analysis of metadata accuracy across all entries.

## 3 RESULTS

### 3.1 Summarization

An example summary of a crawled item, *Python for High Performance* (<https://cvw.cac.cornell.edu/python-performance>), is shown below. The original keywords were: Python, performance, compiling, software, extension modules, algorithms, parallelization, optimization.

**Catalog Description** Python is a very popular programming language for scientific computing, due to both the expressiveness of the language itself and the availability of a rich ecosystem of packages, tools, and libraries that have been developed by the community to support a wide array of different computational tasks. Python is an interpreted language, however, and therefore Python programs are intrinsically slower than equivalent programs written in a compiled language. This roadmap introduces packages, tools, and strategies that are useful for achieving high computational performance with Python, both on workstations and on multiprocessor clusters.

**AI Summary** The content discusses Python's popularity in scientific computing due to its expressiveness and ecosystem of packages. It emphasizes achieving high computational performance with Python on workstations and clusters. The workshop assumes prior experience in Python, UNIX/Linux, and general programming concepts. It targets scientists and engineers interested in improving computational performance. Readers can refer to additional resources for Python introduction. System requirements vary from laptops to High Performance Computing systems, and access to relevant packages is necessary for running code examples.

### 3.2 Keyword evaluation

As part of the automated QA process, keyword quality was evaluated for each crawled item by multiple models. Models were

prompted to score the accuracy of the original keywords, provide a short rationale, and suggest an improved set of keywords.

For item *Python for High Performance*, the keyword evaluation scores ranged from 3 to 4 across the four tested models. GPT-3.5 and GPT-4.1 nano each scored the original keywords a 3, noting that while the terms were generally relevant, they lacked specificity for high-performance or scientific computing. Suggested improvements from these models included adding terms such as “scientific computing,” “optimization,” “compilation,” and “clusters.” GPT-4.1 mini and GPT-4o both assigned a score of 4, finding the keywords largely appropriate but suggesting more precise language and better coverage of HPC concepts.

### 3.3 Comparison of sample items with similar model results, keyword evaluation

In the test set, 23 items had the same score value returned by the 4 different models tested. 5 here are sampled for review, and the submitted description and keywords are shown, as well as the AI generated summary of the crawled page. Each models score and reasoning is shown.

#### Item Title: Data Transfer

**Catalog Description:** Transferring data and code between your workstation and a remote computer is a common part of scientific workflows. Sometimes this data can be quite large, and sometimes you wish to transmit your data securely. And recently, data transfers between cloud storage and computing facilities are becoming increasingly common. There are a number of utilities available to help you accomplish these essential tasks. Your choice of data transfer utility will depend on how much data you are transferring, how you prefer to perform the transfer, and your priorities (including transfer speed, ease of use, security and validation). This topic presents several data transfer options and their pros and cons, as well as ways to make these transfers faster. While the file transfer techniques presented here are useful in many situations, the included examples will use TACC’s Stampede2 and Frontera as the remote computers.

**Submitted Keywords:** data transfer, security, SCP, performance, strategies, data syncing, data compression

**AI Summary:** The content discusses the importance of transferring data and code between your workstation and a remote computer, especially in scientific workflows. It mentions the significance of securely transmitting large data and the increasing trend of data transfers between cloud storage and computing facilities. Different utilities are available for data transfers, and the choice depends on factors like the amount of data, preferred transfer method, and priorities such as speed, security, and validation. The topic covers various data transfer options, their pros and cons, and methods to make transfers faster. Specific examples include using TACC’s Stampede2 and Frontera as remote computers. The workshop aims to improve participants’ ability to transfer files efficiently and choose appropriate transfer utilities based on the situation.

**AI Model:** 3.5

**AI Score:** 4

**AI Reason:** The keywords data transfer, security, SCP, performance,

strategies, data syncing, data compression are fairly accurate based on the abstract and summary.

**AI Model:** 4.1 nano

**AI Score:** 4

**AI Reason:** The keywords accurately cover key aspects such as data transfer, security, SCP, performance, data syncing, and data compression, which are relevant to the abstract and summary. However, they could include more specific terms like ‘cloud storage’ and ‘utilities’ to better reflect the content.

**AI Model:** 4.1 mini

**AI Score:** 4

**AI Reason:** The current keywords mostly reflect the subject matter of data transfer, security, and performance aspects discussed in the abstract and summary. However, ‘SCP’ and ‘data compression’ are not explicitly mentioned or emphasized in the summary or abstract, while other important terms like ‘cloud storage’, ‘file transfer utilities’, ‘validation’, and ‘scientific workflows’ are missing.

**AI Model:** 4o mini

**AI Score:** 4

**AI Reason:** The keywords largely represent the main topics discussed in the abstract and summary, including key considerations for data transfer. However, they could be more specific to the context of scientific workflows and cloud computing.

#### Item Title: AI with Deep Learning

**Catalog Description:** Deep learning comprises a set of methods for Machine Learning and Artificial Intelligence, based on the use of multilayer neural networks to carry out learning. Deep learning techniques can identify patterns in data even within large data sets, and often require substantial computational resources for training model parameters and making predictions. The Frontera supercomputer at the Texas Advanced Computing Center (TACC) is built to support large computational workloads such as those involved with deep learning. Software packages such as TensorFlow, Keras, and PyTorch are widely used to build deep learning pipelines.

**Submitted Keywords:** deep learning, machine learning, artificial intelligence, TensorFlow, TACC, Frontera, PyTorch, parallelism

**AI Summary:** Deep learning is a method within Machine Learning and Artificial Intelligence that uses multilayer neural networks to enable learning and pattern identification in data, often requiring substantial computational resources. Frontera supercomputer at TACC supports deep learning workloads, with software packages like TensorFlow, Keras, and PyTorch commonly used. Prospective users of Frontera are typically familiar with HPC and running jobs on clusters. The workshop does not have formal prerequisites, but some familiarity with Linux is recommended. Running example codes requires access to systems with TensorFlow and PyTorch or the ability to install them in a Python virtual environment. Access to supercomputers at TACC, such as Frontera, requires an allocation to the system.

**AI Model:** 3.5

**AI Score:** 4

**AI Reason:** The keywords metadata tag is mostly accurate and relevant to the content of the abstract and summary.

**AI Model:** 4.1 nano

**AI Score:** 4

**AI Reason:** The keywords accurately cover major topics such as deep learning, machine learning, AI, and software tools like TensorFlow and PyTorch. 'Frontera' and 'TACC' are relevant as they relate to the infrastructure discussed. However, 'parallelism' is somewhat broad and less emphasized specifically, and 'Frontera' and 'TACC' are more about the hardware platform than the core concepts.

**AI Model:** 4.1 mini

**AI Score:** 4

**AI Reason:** The keywords mostly align with the abstract and summary, covering deep learning, machine learning, artificial intelligence, TensorFlow, TACC, Frontera, PyTorch, and parallelism. However, 'Keras' is mentioned in both abstract and summary as a key software package but is missing from the keywords. Also, 'HPC' or 'High Performance Computing' could be relevant given the emphasis on supercomputing resources like Frontera.

**AI Model:** 4o mini

**AI Score:** 4

**AI Reason:** The keywords accurately reflect the core topics of deep learning, machine learning, and the relevant software packages, but could be more specific to the context of the workshop.

**Item Title: MPI Point-to-Point**

**Catalog Description:** Point-to-point communication encompasses all the methods MPI offers to transmit a message between a pair of processes. MPI features a broad range of point-to-point communication calls; they differ in subtle ways which can affect the performance of your MPI program. This roadmap details and differentiates the various types of point-to-point communication available in MPI-3.0 and discusses when and how to use each method. We will examine blocking as well as nonblocking communication calls and go through some examples using these methods. All exercises and examples are verified to work on Stampede2 and Frontera. MPI also provides for transmission of messages among groups of processes, which is called collective communication. Collective communication is the subject of a different roadmap. This is the second of five related roadmaps in the Cornell Virtual Workshop that cover MPI. To see the other roadmaps available, please visit the complete roadmaps list.

**Submitted Keywords:** MPI, message passing, buffer, non-blocking, parallelism, probe, deadlock, persistent communication

**AI Summary:** The content discusses point-to-point communication in MPI, which involves transmitting messages between pairs of processes. It explains the various methods available in MPI-3.0 for point-to-point communication, including differences that can impact performance. The roadmap provides details on both blocking and nonblocking communication calls, with examples. It also mentions that collective communication, which involves message transmission among groups of processes, is covered in a separate roadmap. Various exercises and examples discussed in the content are verified to work on Stampede2 and Frontera. Completing this roadmap will enable users to use MPI on these platforms effectively.

**AI Model:** 3.5

**AI Score:** 4

**AI Reason:** The keywords metadata tag is mostly accurate and

relevant to the content of the abstract and summary.

**AI Model:** 4.1 nano

**AI Score:** 4

**AI Reason:** The keywords accurately reflect key concepts discussed in the abstract, such as MPI, message passing, non-blocking communication, and deadlock. However, they could be slightly refined for completeness and clarity.

**AI Model:** 4.1 mini

**AI Score:** 4

**AI Reason:** The keywords are generally relevant and cover many core concepts related to point-to-point MPI communication such as MPI, message passing, non-blocking, probe, and persistent communication. However, the keyword 'buffer' is somewhat generic and not emphasized in the abstract or summary. Also, 'deadlock' is not explicitly mentioned or discussed as a focus in the description.

**AI Model:** 4o mini

**AI Score:** 4

**AI Reason:** The keywords accurately reflect the main topics discussed in the abstract and summary, but could be more specific in relation to the types of communication methods and performance considerations.

**Item Title: MATLAB Programming**

**Catalog Description:** MATLAB is a programming and numeric computing platform used to analyze data, develop algorithms and create models. This roadmap includes topics that provide a basic introduction to MATLAB, discuss how to write MATLAB code that can be compiled for speed, and offer tips on getting the best performance out of MATLAB. The example code in these topics was run and verified using MATLAB 2017a through 2021a. Most of it (outside of Mex and other FFI examples) should also work with slight modifications in GNU Octave.

**Submitted Keywords:** MATLAB, programming, scripting, compiling, functions, GPU, data types, arrays

**AI Summary:** MATLAB is a platform for programming and numeric computing used for data analysis, algorithm development, and modeling. This content roadmap covers basic MATLAB introduction, optimizing code for speed, and maximizing performance. The example code is verified on MATLAB versions 2017a to 2021a and can work with minor adjustments in GNU Octave. A MATLAB license is necessary to run the software, and trial licenses are accessible. MATLAB is frequently licensed on high-performance computing systems like those at the Texas Advanced Computing Center (TACC).

**AI Model:** 3.5

**AI Score:** 4

**AI Reason:** The provided keywords are relevant to the content in the abstract and summary. They accurately reflect the topics covered in the MATLAB programming resource.

**AI Model:** 4.1 nano

**AI Score:** 4

**AI Reason:** The keywords accurately reflect core topics discussed in the abstract and summary, such as MATLAB, programming, algorithms, code optimization, performance, and compatibility with GNU Octave. However, some keywords like 'compiling' are less

emphasized, and 'GPU' and 'data types' are only briefly mentioned and not central to the content.

**AI Model:** 4.1 mini

**AI Score:** 4

**AI Reason:** Most keywords accurately reflect the content; however, 'scripting' and 'data types' are less emphasized in the abstract and summary. 'Compiling' is relevant, and 'GPU' is somewhat tangential as GPU usage is not explicitly covered in the abstract. Keywords could be more aligned with data analysis, algorithm development, modeling, and performance optimization aspects highlighted in the description.

**AI Model:** 4o mini

**AI Score:** 4

**AI Reason:** The keywords are generally relevant to the topic of MATLAB programming and cover important aspects such as coding, data handling, and performance optimization. However, they could be more focused on specific use cases and the content roadmap.

### Item Title: COMPLECS: Parallel computing concepts

**Catalog Description:** All users of advanced cyberinfrastructure, whether they develop their own software or use 3rd party applications, should understand fundamental parallel computing concepts. In this webinar we cover supercomputer architectures, the differences between threads and processes, implementations of parallelism (e.g., OpenMP and MPI), strong and weak scaling, limitations on scalability (Amdahl's and Gustafson's Laws) and benchmarking. We also discuss how to choose the appropriate number of compute cores or nodes when running your applications and, when appropriate, the best balance between threads and processes. This webinar does not assume any programming experience and is suited for a wide audience, including current and prospective users of parallel computers, anyone who expects to write a proposal for computer time or those who are simply curious about parallel computing.

**Submitted Keywords:** Expanse, HPC Training, Industry, TSCC, Voyager

**AI Summary:** SDSC enables discovery through innovative supercomputers, research computing services, and infrastructure support. They provide AI-enabled data science, cyberinfrastructure solutions, consulting, and training for academic researchers and industry partners. The organization translates research into impacts at national, state, and UC system levels, meeting stakeholder needs and advancing scientific discovery. SDSC bridges gaps through education and training, facilitating the impactful use of data and computational resources.

**AI Model:** 3.5

**AI Score:** 2

**AI Reason:** Some of the keywords provided are relevant such as 'HPC Training', but there are better keywords that can be derived from the abstract and summary.

**AI Model:** 4.1 nano

**AI Score:** 2

**AI Reason:** The current keywords (Expanse, HPC Training, Industry, TSCC, Voyager) do not accurately reflect the core content of the webinar, which focuses on fundamental parallel computing

concepts, architectures, and scalability. They are either too broad or unrelated to the specific subject matter.

**AI Model:** 4.1 mini

**AI Score:** 2

**AI Reason:** The current keywords focus on specific HPC systems (Expanse, TSCC, Voyager) and a general term 'HPC Training', which are only tangentially related to the webinar content. They do not accurately capture the core themes of the webinar such as parallel computing concepts, parallelism, supercomputing architectures, and related technologies.

**AI Model:** 4o mini

**AI Score:** 2

**AI Reason:** The current keywords do not accurately reflect the focus on fundamental parallel computing concepts, supercomputer architectures, and training for users of advanced cyberinfrastructure.

### 3.4 Comparison of sample items with different model results, keyword evaluation

The 5 catalog items with the greatest standard deviation between the 4 model scores are listed below.

#### Item Title: Parallel I/O

**Catalog Description:** This roadmap presents basic concepts and techniques that will allow your application to take advantage of parallel I/O to increase throughput and improve scalability. The parallel I/O software stack is introduced from the hardware level on up. Emphasis is placed on the Lustre parallel file system, and on MPI-IO as a fundamental API for enabling parallel I/O. These are the building blocks of typical HPC software stacks, including those available on the HPC systems at TACC.

**Submitted Keywords:** parallel I/O, message passing, file systems, storage, Lustre, MPI-IO, collective operations, parallelization

**AI Summary:** The content discusses a roadmap for understanding parallel I/O techniques to enhance application performance by leveraging parallel file systems and MPI-IO. It covers fundamental concepts related to Lustre parallel file system and MPI-IO, commonly used in HPC environments like TACC systems. Prior knowledge of Linux shell commands, parallel programming, and MPI is assumed, with suggested roadmaps for prerequisites. Programming experience in C or Fortran is recommended, with foundational roadmaps available. Stampede3 or Frontera systems are used for examples and exercises, requiring specific system access. The roadmap is a continuation of the MPI Advanced Topics content but is not a mandatory prerequisite.

**AI Model:** 3.5

**AI Score:** 3

**AI Reason:** The keywords provided are relevant to the content mentioned in the abstract and summary, but they could be more specific and targeted.

**AI Model:** 4.1 nano

**AI Score:** 4

**AI Reason:** The keywords largely align with the abstract's focus on parallel I/O, file systems, MPI-IO, and Lustre in HPC contexts. Minor elements like message passing and collective operations are

relevant but less emphasized, which explains the slightly lower score.

**AI Model:** 4.1 mini

**AI Score:** 5

**AI Reason:** The keywords accurately reflect the main topics and concepts discussed in the abstract and summary, including parallel I/O, message passing, file systems, storage, Lustre, MPI-IO, collective operations, and parallelization, which are relevant to the content of the resource.

**AI Model:** 4o mini

**AI Score:** 5

**AI Reason:** The keywords relevantly capture the main topics discussed in the abstract and summary, including important terms like Lustre and MPI-IO, which are central to the content.

### Item Title: HPC Workshop: MPI

**Catalog Description:** This two day workshop is intended to give C and Fortran programmers a hands-on introduction to MPI programming. Both days are compact, to accommodate multiple time zones, but packed with useful information and lab exercises. Attendees will leave with a working knowledge of how to write scalable codes using MPI – the standard programming tool of scalable parallel computing.

**Submitted Keywords:** MPI

**AI Summary:** A two-day workshop aims to provide C and Fortran programmers with a practical introduction to MPI programming, focusing on writing scalable codes for parallel computing. The workshop features hands-on activities and attendees will gain knowledge on using MPI effectively. The slides provided are from the latest MPI workshop, and inquiries about the workshop can be directed to *Name deleted*.

**AI Model:** 3.5

**AI Score:** 4

**AI Reason:** The keyword 'MPI' is accurate and relevant based on the abstract and summary.

**AI Model:** 4.1 nano

**AI Score:** 3

**AI Reason:** The keyword 'MPI' is relevant but too broad; the meta-data lacks specific keywords related to parallel programming, HPC, C, Fortran, or workshop training.

**AI Model:** 4.1 mini

**AI Score:** 2

**AI Reason:** The current keyword 'MPI' is too broad and generic given the detailed context of the workshop. It does not fully capture the scope, audience, or programming languages involved in the workshop.

**AI Model:** 4o mini

**AI Score:** 2

**AI Reason:** The keyword 'MPI' is too narrow and does not capture the comprehensive content of the workshop or the programming languages involved.

### Item Title: Data Analysis with R

**Catalog Description:** This lecture presents several examples of data analysis and visualization in R. It will demonstrate a variety of analyses intended to help researchers determine if learning R is a good investment for their research, including new data analysis examples drawn from the WCM community.

**Submitted Keywords:** data-analysis, visualization, R, RStudio, data-science, statistics, training, programming

**AI Summary:** *Name deleted*, a computational scientist with the Cornell University Center for Advanced Computing, discussed data analysis in R, focusing on the Tidyverse and ggplot. He shared examples of using R for data analysis, introduced the importance of data types, variables, functions, and data frames in R. He also talked about how to import and manipulate data, generate summary tables, and create plots using ggplot, including examples of cumulative total calculations and choropleth maps. Additionally, he mentioned the benefits of using literate analysis tools like Quarto Notebooks for reproducibility and interactive plots like Plotly for exploration. *Name deleted* emphasized the significance of factors like the pipe operator for efficient code writing and recommended resources for further learning in R analysis.

**AI Model:** 3.5

**AI Score:** 3

**AI Reason:** The keywords data-analysis, visualization, R, RStudio, data-science, statistics are relevant based on the abstract and summary provided.

**AI Model:** 4.1 nano

**AI Score:** 3

**AI Reason:** The keywords include general terms related to R and data visualization but lack specific references to the key tools and concepts discussed, such as Tidyverse, ggplot, data frames, and Reproducibility tools like Quarto and Plotly, which are central to the abstract and summary.

**AI Model:** 4.1 mini

**AI Score:** 5

**AI Reason:** The keywords accurately reflect the content of the lecture, covering data analysis, visualization, R programming, RStudio, data science, statistics, training, and programming, which align well with the abstract and summary.

**AI Model:** 4o mini

**AI Score:** 4

**AI Reason:** The keywords are largely accurate and relevant to the content of the video, though there could be more focus on specific tools and concepts mentioned in the summary.

### Item Title: HPC Workshop: Shared Memory Programming Using OpenMP

**Catalog Description:** This workshop is intended to give C and Fortran programmers a hands-on introduction to OpenMP programming. Attendees will leave with a working knowledge of how to write scalable codes using OpenMP. Hands-on exercises are included to give attendees practice with the concepts presented.

**Submitted Keywords:** OpenMP

**AI Summary:** The workshop provides C and Fortran programmers with a practical introduction to OpenMP programming, focusing on writing scalable codes using OpenMP. Attendees participate in hands-on exercises to reinforce the concepts presented. Contact *Name deleted* at *Name deleted* for any inquiries.

**AI Model:** 3.5

**AI Score:** 5

**AI Reason:** The keyword 'OpenMP' accurately reflects the main topic of the workshop.

**AI Model:** 4.1 nano

**AI Score:** 3

**AI Reason:** The keyword 'OpenMP' is relevant but too broad; the metadata would benefit from more specific keywords such as 'Shared Memory Programming,' 'Parallel Computing,' 'C Programming,' and 'Fortran.'

**AI Model:** 4.1 mini

**AI Score:** 3

**AI Reason:** The keyword 'OpenMP' is relevant but too general and singular given the content focuses on shared memory programming, C and Fortran languages, and hands-on training. More specific keywords could provide better metadata accuracy.

**AI Model:** 4o mini

**AI Score:** 2

**AI Reason:** The current keyword 'OpenMP' is too narrow and does not capture the full scope of the workshop's content which includes programming languages and concepts related to scalable coding.

## Item Title: MPI One-Sided Communication

**Catalog Description:** One-sided communication methods were added to MPI as a part of the MPI-2 improvements and were greatly expanded in MPI-3 by including support for shared memory windows, windows with dynamically attached memory, request-based communication calls, and more window locking mechanisms. On Stampede2 and Frontera, the one-sided communication methods implemented in the Intel MPI and MVAPICH2 libraries use the Remote Direct Memory Access (RMA or RDMA) functionality provided by low-latency interconnect fabrics such as Omni-Path and InfiniBand. In this roadmap, we will introduce the various components of MPI RMA and how to use them. All exercises and examples are verified to work on Stampede2 and Frontera. This is the fifth of five related roadmaps in the Cornell Virtual Workshop that cover MPI. To see the other roadmaps available, please visit the complete roadmaps list.

**Submitted Keywords:** MPI, communication, window, synchronization, remote memory access, collective operations, fence, buffer

**AI Summary:** MPI has added one-sided communication methods as part of its improvements, with expanded support for shared memory windows, dynamically attached memory, request-based communication calls, and various window locking mechanisms in MPI-3. These methods are implemented in Intel MPI and MVAPICH2 libraries on systems like Stampede2 and Frontera using Remote Direct Memory Access (RMA) functionality. This roadmap introduces various components of MPI RMA and provides exercises and examples verified to work on Stampede2 and Frontera. It is

the fifth out of five related roadmaps in the Cornell Virtual Workshop covering MPI. The completion of this roadmap should enable users to understand and utilize MPI RMA features effectively, with specified system requirements.

**AI Model:** 3.5

**AI Score:** 5

**AI Reason:** The keywords provided align well with the content of the abstract and summary.

**AI Model:** 4.1 nano

**AI Score:** 3

**AI Reason:** The keywords broadly cover MPI, communication, and synchronization but lack specificity about the focus on Remote Memory Access (RMA) and shared memory windows, which are central topics in the abstract and summary.

**AI Model:** 4.1 mini

**AI Score:** 4

**AI Reason:** The keywords generally cover the main topics such as MPI, communication, window, synchronization, remote memory access, fence, and buffer. However, 'collective operations' is less relevant here since the abstract and summary focus primarily on one-sided communication and Remote Memory Access (RMA) mechanisms rather than collective operations. Also, adding 'RMA', 'one-sided communication', 'Intel MPI', and 'MVAPICH2' would improve relevance.

**AI Model:** 4o mini

**AI Score:** 2

**AI Reason:** The current keywords do not fully encompass the specific aspects of one-sided communication methods introduced in MPI-2 and MPI-3, such as request-based communication, shared memory windows, and RMA functionality, which are significant in the abstract and summary.

## 4 DISCUSSION

As a summarization agent, GPT 3.5 was sufficient to create reasonable summaries of crawled content and submitted metadata, overlooking HTML formatting and general page boilerplate, to access meaningful content on the page. Both catalog metadata and crawled content were fed into the summarization agent, it might be interesting in future to look at whether catalog metadata as an input improves summarization ability relative to crawling alone.

Additionally, a substantial number (23 of 128 items) had the same scores between the models. This combined with the ability of smaller models to distinguish between content and non-content when summarizing crawled media does indicate that there is strong potential for the use of commercial unmodified LLMs for automation of metadata review.

While overall, the quality of agent responses scales as expected with the size and capability of the LLM, in some cases we saw higher level model "overthinking" analysis. Consider HPC Workshop: Shared Memory Programming Using OpenMP under the differing results. The catalog item is for a workshop on OpenMP. The submitted keyword is "OpenMP." GPT-3.5 found this to be an adequate keyword, whereas GPT-4.1 and GPT-4o scored it lower with the reason that the keyword was too vague, wanting the languages used in the workshop and additional training also included.

We noticed that stronger models tended to favor more expansive keyword sets, which could harm discovery by failing to focus on the core principles of each catalog item. The larger models created suggested keyword lists with more depth and penalized submitted metadata for the lack of it, but in many cases simpler keywords were a feature rather than a flaw. This poses a challenge for further automation of metadata review, one which we hope can be mitigated through better prompt engineering. It may also be helped through other efforts to define an ontology for HPC training.

Additionally, we would like to include direct comparisons with LLM based metadata review and human metadata review in the future.

## 5 RELATED WORK

Members of the HPC-ED team have been using HPC-ED metadata to develop LLMs as part of their training curricula. These include the SDSC HPC/CI Training Catalog LLM that uses the Microsoft GraphRAG model [4]; the Texas Advanced Computing Center built a Retrieval-Augmented Generation (RAG) application that scrapes the HPC-ED federated catalog, using Sentence Transformers for semantic search, then generating a personalized, step-by-step curricula from natural language queries. The system ranks relevant modules, suggests prerequisites and follow-up topics, and supports HTML/PDF export, helping students and instructors quickly connect questions to authoritative HPC learning materials while lowering barriers for beginners. The Cornell University Center for Advanced Computing is also developing a RAG agent but which is based on Cornell Virtual Workshop (CVW) training materials; starting with an AI-enabled search function, which will accept natural language queries, including the level and depth of material sought in addition to topic or goal, and output a simple answer as well as link(s) to CVW learning resources.

The LLMs will be tested using the crawler described above.

## ACKNOWLEDGMENTS

This work builds on the efforts of all of the partners of the HPC-ED project and is supported by CyberTraining: Pilot: HPC ED: Building a Federated Repository and Increasing Access through CyberTraining – NSF grant OAC-2320977.

## REFERENCES

- [1] K Cahill, D Joiner, S Lathrop, S Mehringer, and A & Navarro, J-P & Weeden. 2022. Final Results: National Survey on Educational and Training Materials Repositories. <https://www.cac.cornell.edu/about/pubs/Survey2022.pdf>
- [2] Johannes Depoix. 2025. YouTube Transcript API. <https://github.com/jdepoix/youtube-transcript-api>. Accessed: 2025-08-11.
- [3] Dublin Core. 2043. LRMI Learning Resource Type vocabulary. Retrieved September 8, 2024 from [https://www.dublincore.org/specifications/lrmi/concept\\_schemes/learningResourceType](https://www.dublincore.org/specifications/lrmi/concept_schemes/learningResourceType)
- [4] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, Robert Osazuwa Ness, and Jonathan Larson. 2025. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv:cs.CL/2404.16130 <https://arxiv.org/abs/2404.16130>
- [5] Leyla Garcia, Bérénice Batut, Melissa L. Burke, Mateusz Kuzak, Fotis Psomopoulos, Ricardo Arcila, Teresa K. Attwood, Niall Beard, Denise Carvalho-Silva, Alexandros C. Dimopoulos, Victoria Dominguez Del Angel, Michel Dumontier, Kim T. Gurwitz, Roland Krause, Peter McQuilton, Loredana Le Pera, Sarah L. Morgan, Päivi Rauste, Allegra Via, Pascal Kahlem, Gabriella Rustici, Celia W.G. Van Gelder, and Patricia M. Palagi. 2020. Ten simple rules for making training materials FAIR. *PLoS Computational Biology* 16, 5 (2020), 1–9. <https://doi.org/10.1371/journal.pcbi.1007854>
- [6] Globus. 2024. Globus Search Service. <https://docs.globus.org/api/search/>.
- [7] N. J. Hoebelheinrich, K. Biernacka, M. Brazas, L. J. Castro, N. Fiore, M. Hellström, E. Lazzeri, E. Leenarts, P. M. Martinez Lavanchy, E. Newbold, A. Nurnberger, E. Plomp, L. Vaira, and A. van Gelder, C. W. G. & Whyte. 2022. Recommendations for a minimal metadata set to aid harmonised discovery of learning resources. <https://doi.org/10.15497/RDA00073>
- [8] Susan Mehringer, Mary P Thomas, Charlie Dey, Kate Cahill, David Joiner, Richard Knepper, and Jaime H Powell. 2023. Scaling HPC Education. In *Tenth SC Workshop on Best Practices for HPC Training and Education BPHTE23*. Denver, 41–46. <https://doi.org/10.22369/issn.2153-4136/15/1/8>
- [9] Leonard Richardson. 2024. Beautiful Soup Documentation. <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. Accessed: 2025-08-11.
- [10] Mary P Thomas, Susan Mehringer, Katharine Cahill, Charlie Dey, Brian Guilfoos, David Joiner, John-Paul Navarro, Jaime H Powell, and Richard Knepper. 2024. Building a Federated Catalog for CyberTraining Materials: The HPC-ED Pilot Project. In *Practice and Experience in Advanced Research Computing 2024: Human Powered Computing* (New York, NY, USA), Vol. 1. ACM, 1–5. Issue 1. <https://doi.org/10.1145/3626203.3670586>