MOP: Augmenting and Standardizing Heterogeneous Knowledge Graph Data Sources

Julia Evans[†], Mirjan Hoffmann[†], Sophie Matter[†] and Axel Klinger

TIB – Leibniz Information Centre for Science and Technology, Hanover, Germany

Abstract

We present MOP (Metadata Optimization Pipeline), an application to harmonize and enrich heterogeneous metadata from scientific knowledge graphs. Such metadata often varies widely in its quality, completeness, and consistency, particularly in freetext fields like titles and descriptions, which negatively impacts finadability. MOP addresses this limitation by leveraging large language models (LLMs) to enrich existing metadata or generate missing fields. In a multi-stage enrichment process, LLMs are used to generate summaries from the full text of open-access resources, which then serve as input to produce additional metadata fields. This enriched metadata is stored separately from the original records, preserving the integrity of human-curated data while still enhancing discoverability and usability of the resource metadata. In this paper we discuss implementation details, analyze LLM output quality, and reflect on challenges encountered and lessons learned, particularly with regard to managing compute resource limitations. MOP demonstrates a practical, modular approach to improving functional metadata quality through LLMs in large, distributed knowledge graphs.

Keywords

Linked Data, Data Integration, LLM Assistance, Data Extraction

1. Introduction

Knowledge graph-centric metadata aggregation systems operate on a wide array of distributed sources, resulting in metadata which is often heterogeneous, sparse, and inconsistently structured, particularly in freetext fields such as titles and descriptions [1, 2]. These inconsistencies stem from the varied requirements and guidance followed by individual repositories, which are themselves shaped by local institutional practices and technical constraints. Poor metadata quality can hinder searchability, reduce findability, and compromise semantic interoperability across systems, ultimately limiting the utility of aggregated resources for end users and downstream applications. This is a known problem for repositories indexing objects of scientific knowledge such as scholarly articles, academic works, or research artifacts.

To address these challenges, we present the use of a large language model (LLM) [3] within a targeted metadata enrichment pipeline for unreliable or incomplete metadata records. LLMs excel at producing fluent text and synthesizing unstructured content into coherent narratives [4], making them well-suited to generating fields such as title, description, and keywords. At the same time, LLMs cannot be trusted to reliably produce schema-compliant output or follow strict vocabulary constraints. We believe that effectively capitalizing on LLM strengths while mitigating their limitations results in a tradeoff that prioritizes stability and maintainability over achieving state-of-the-art results.

This work presents MOP (Metadata Optimization Pipeline), a concrete and deployed application for cleaning up messy metadata records. It integrates Semantic Web technologies such as linked data principles [5] and SKOS vocabularies [6] with LLMs to address real-world challenges in metadata aggregation. Structured fields such as keywords and subjects, which must conform to specific data models or controlled vocabularies, are post-processed using lightweight correction functions and mapping utilities to ensure consistency and compliance with our knowledge graph schema [7]. Our

5th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment, Nov 2024, Nara, Japan

[†] These authors contributed equally.

julia.evans@tib.eu (J. Evans); mirjan.hoffmann@tib.eu (M. Hoffmann); sophie.matter@tib.eu (S. Matter); axel.klinger@tib.eu (A. Klinger)

0000-0001-6442-3510 (A. Klinger)

system tackles practical issues of semantic interoperability, schema compliance, and data quality that arise when harmonizing heterogeneous metadata across distributed repositories. We provide evidence of implementation decisions, post-processing strategies, and the constraints faced during deployment, such as scalability concerns. By combining LLMs with a schema-aware enrichment pipeline, this work exemplifies how LLMs can enhance knowledge graphs without compromising their structural integrity. As an example use case, we apply MOP to an open educational resources (OER) repository, which contains varied scientific works ranging from texts to experiment notes to software. However, it is generally resource-agnostic and can be applied to other domains. Moreover, the lessons learned and design decisions we have faced are valuable and informative for any similar works.

This paper is structured as follows. First, section 2 provides an overview of the data sources and applicable use case. Details of our approach are described in section 3. Then section 4 shows an example of the system results in the frontend and describes common error types. Afterwards in section 5, we present discussion points around implementing such systems and explain how our approach has evolved. Related work is presented in section 6. Lastly, section 7 concludes this paper and presents possible future work.

2. Data Source and Use Case

Our work is grounded in a concrete in-use application: MOP has been developed on OERSI (Open Educational Resources Search Index), a federated search platform aggregating metadata from a wide range of educational repositories. OERSI exemplifies the metadata challenges outlined above: heterogeneous inputs, inconsistent freetext descriptions, and varied schema adherence across sources. It connects a wide range of OER sources, including state initiatives, university and library repositories, and subject-specific collections. OERSI has been developed collaboratively by the Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (hbz) and the German National Library of Science and Technology (TIB) as an open-source project and is used by ministries in all German states, especially the states of Lower Saxony, North Rhine-Westphalia, and Hesse.[1] All development takes place publicly on GitLab[2].

Rather than storing educational content directly, OERSI aggregates and standardizes metadata from diverse sources, allowing users to perform uniform searches across its network of connected repositories without duplicating the content. The structure is based off of the Allgemeines Metadatenprofil für Bildungsressourcen (AMB). This schema for describing educational resources across different contexts is primarily built on Schema.org and the Learning Resource Metadata Initiative (LRMI), with supplementary use of elements from the Simple Knowledge Organization System (SKOS).

The connected sources in OERSI contain metadata following different schemas, with varied requirements and guidelines for text input. Each source (or source type, i.e. EduSharing instance) has a custom mapping to the OERSI schema. While this always results in a technically correct representation, the quality and utility of the resulting metadata representation varies, most especially in freetext fields. Some of the weakness are outlined below.

- **Titles**: The original titles of resources may make sense in the context of their hosting repository, but may be inconsistent or uninformative outside of it, such as *"Lecture 09. RNA.: Part 2"*, which lacks meaningful context outside of its specific course sequence.
- **Descriptions**: Many resources either lack descriptions entirely or only offer extremely brief summaries. Others describe the institute or process which produced the work but say little of the work itself. And still others describe the intended audience of the resource but not the content. While these latter two styles are informative in their own right, more content-specific information could nonetheless improve their discoverability and usefulness.

[1] In the months of April, May, and June 2025, OERSI received almost 1 million (965,149) requests through the API and more than 63,000 visits to the website.

[2] https://gitlab.com/oersi

- **Keywords**: The majority of resources (approximately two-thirds) indexed by OERSI do not have keywords, which can be useful for getting a very quick overview of the resource or support with subject classification.
- **Subject**: This property is highly significant for filtering, but is not present for all resources, or is sometimes a top-level concept when a more specific classification would be more precise and relevant.

Additionally, some content which would be helpful for presenting search results to users is not part of the schema.

- **Card descriptions**: For user interfaces like search result cards, a concise, single-sentence summary is ideal. However, the first sentence of a description is not always suitable for this purpose, as it may be too vague, overly technical, or lack standalone clarity.

OERSI imposes very few required fields in order to maximize the resource coverage. Nonetheless, certain fields are highly valuable for enabling effective discovery and leaving them out is limiting. Some repositories, especially for open textbooks, lack subject classification. If subjects could be reliably and accurately assigned, it would significantly enrich the findability of these materials. One challenge to doing so, however, is the need for the generated subjects to follow a controlled vocabulary. In our system, subject terms must conform to the Hochschulfächersystematik, or Higher Education Subject Classification, a standardized higher education subject taxonomy based on the German statistics office's (Destatis) classification of subject groups, study areas, and study subjects, which is commonly used in Germany and Austria.

To address these limitations without interfering with the integrity of human-generated and -curated metadata, MOP introduces a parallel layer of automatically-generated enrichment. The goal is to generate supplementary metadata fields, such as improved descriptions, titles, or subject classifications, which users may optionally view alongside the original metadata. These enriched fields are stored as separate properties to preserve provenance and enable transparent differentiation between human and LLM-generated content. The following sections outline how MOP retrieves source material, extracts and processes its content as text, generates metadata using LLMs, and stores the LLM output.

3. MOP Architecture

The architecture of MOP is modular. There is as little coupling as possible between the modules so that the individual modules can be easily adapted or replaced. While some domain- or resource-specific configuration is necessary, particularly for taxonomy alignment and language handling, the pipeline is designed to be modular and schema-agnostic, allowing key components (e.g., prompt templates, postprocessing, subject mapping) to be adapted independently. This makes it easier to reuse MOP and adapt it to other usage scenarios. All code is fully accessible to the public in our GitLab repository[3].

3.1. Loading Data

MOP has been built as a supplementary module to OERSI, but with its own separate and self-contained architecture. The application begins by fetching metadata records from the OERSI index. Although most of the resources indexed in OERSI have an open license, MOP only processes records that explicitly permit derivative works. For each such record, the application extracts the direct download link to the resource, if present. These links are then used to retrieve the actual resource files. Upon successful retrieval, MOP processes the content by extracting textual data, either through text extraction for PDF files or transcription for AV files. For PDF files, the bytes in the HTML response are decoded and parsed as text.[4] For AV file types, only the audio is processed using the Faster Whisper model to transcribe

[3] https://gitlab.com/oersi/sidre/metadata-optimization
[4] The `pypdf` package was used for processing PDFs.

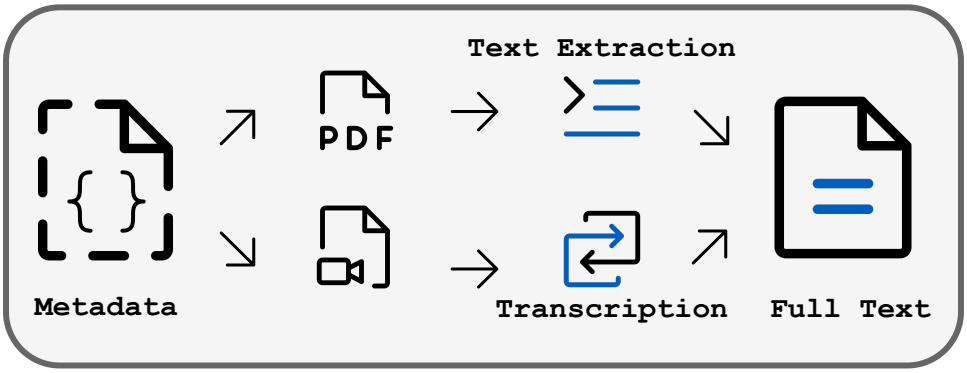

Figure 1: Loader Module: Loading data from the original OER metadata in various modalities to acquire the full text for further processing.

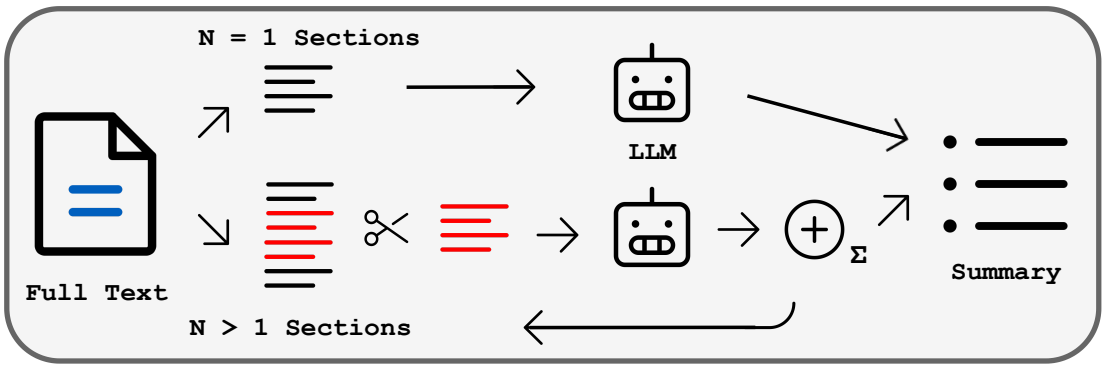

Figure 2: Generator Module - Summary: Generating a summary of the full text to be used for downstream requests. The number of sections is determined by the tokenized length of the full text plus prompt, divided by the LLM context size.

the content into text. The resulting full text of each file is then cached locally as a plain text file. This workflow is depicted in Figure 1.

3.2. Generating LLM Enrichments

Workflow. The generation module of the MOP pipeline comprises two phases that apply LLMs to enrich each resource with additional metadata. The first phase is responsible for generating a summary based on the full text of the resource. Because the full text may be quite lengthy, it is first necessary to determine if it exceeds the context size of the LLM. For this we use the model's own tokenizer to correctly tokenize (i.e., chunk words into smaller lexical units) the full text as well as the prompt. If their combined tokens exceed the context window of the LLM, the text is segmented into smaller sections. The segmentation is done by identifying the longest possible sequence of tokens that fits within the context window and then locating the last sentence-ending punctuation mark (period, exclamation mark, or question mark) within that window. Each section is then individually summarized, with all sectional summaries concatenated in order to produce a coherent overall summary for the resource.[5] Resources for which a summary cannot be generated are excluded from further processing. See Figure 2 for a representation of this process.

The second phase focuses on generating additional metadata fields, the selection of which is configurable. The currently supported fields are description, short description (optimized for search result

[5]Only if the concatenated summaries would later exceed the available amount of tokens, the summarization is performed again - this time with the summaries themselves.

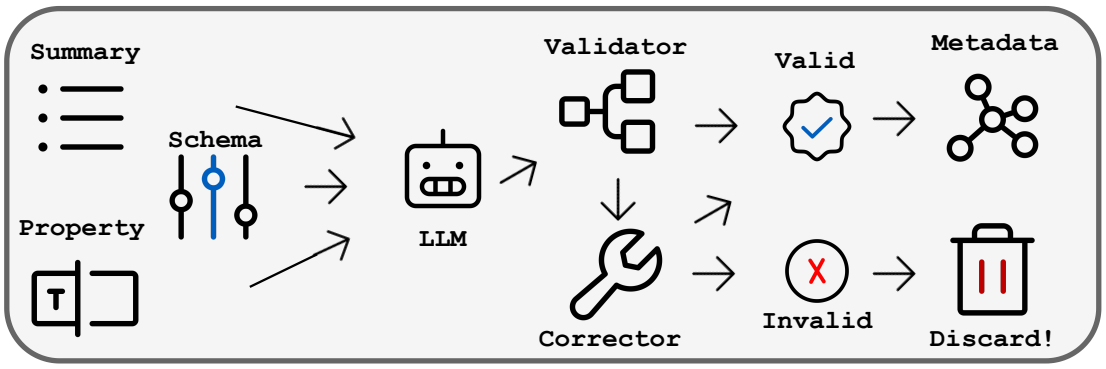

Figure 3: Generator Module - Metadata: Generating individual metadata fields using as input the LLM summary, property-specific prompt, and schema information of expected data type and, for subjects, the subject hierarchy vocabulary. The output is then checked against a validator to ensure schema compliance.

cards), title, keywords, and subjects. For each field, the LLM is queried separately, using the previously generated summary as input, and additional fields from the record metadata – such as title or resource type (Textbook, Exercise, Lesson Plan, etc) – may be included depending on the configuration specification. Queries for subjects also include the controlled vocabulary for reference, with instruction that the generated subject must conform exactly to one of these options. (Refer back to section 2 for details on the controlled vocabulary we use.) The output is validated against a predefined JSON schema and then all successfully generated metadata and summaries are cached in a local relational database for future retrieval and reuse. This workflow is shown in Figure 3.

LLM Model. The choice of language model affects both the performance and the infrastructure surrounding the service [8]. Considering that LLMs require a lot of compute to be able to function properly for a production-ready service, and taking into account the internal constraints on hardware requirements [9], we investigated LLMs with fewer than 40B parameters.[6] An additional requirement was that the model must be free and open-source. We selected Qwen2.5 7B after performing a small comparative study[7] between Qwen2.5, LLaMA 3 8B Instruct, and Phi-3.5, and finding it performed the most consistently. The model is strong at multilingual text generation, it was finetuned particularly on long-text summarization and information extraction, and most importantly it has a context window that is on the larger end of the scale which can accommodate 131K tokens. Considering that several of the OER repositories we operate on contain textbooks, a large context size is highly valuable. (That said, at the moment, the compute resources available to us for this project do not permit taking advantage of the maximum context size.) Another aspect to consider carefully is the choice of the LLM engine that will serve the model and its services. Various options are created and are still being worked on by the community to serve as viable options. We chose Ollama as it offers an easy onboarding process and provides access to plenty of open-source models that are compatible with it. Furthermore, it offers quantization support to enable low-resource systems to run LLMs.

Ensuring Schema Compliance. One challenge of incorporating LLM-generated data into linked data is maintaining schema integrity. For this, we implement a lightweight validation and correction pipeline. While most of the enriched fields (e.g., title, description) are freetext and thus inherently unstructured, certain fields - such as keywords and subject - require structured representations. Since LLM outputs are returned as raw strings, it is necessary to validate and, where needed, correct the generated content to meet these structural requirements. We employ JSON Schema for validation and use a set of type correction functions to address common format inconsistencies. These include simple heuristics to transform strings or dictionaries into arrays, enforce homogenous item types within lists, and serialize complex types into the required formats. This approach has been informed by our

[6]The choice of the threshold is experimental, and based on consultations with industry and research experts.
[7]We qualitatively evaluated four prompt types across six resources and found Qwen2.5 to be the most consistent in output quality. See the Appendix for more information.

observations of typical LLM output errors on our specific data.

Generating valid subject metadata presents particular challenges due to the need for output aligned with a controlled vocabulary and the complexity of subject classification itself. In addition to conforming to the taxonomy of Hochschulfächersystematik, or Higher Education Subject Classification, they must be represented as a structured JSON object containing a unique identifier and at least one language-tagged preferred label. To map LLM-generated subject strings to valid entries in this taxonomy, we perform a normalization step (removing punctuation, enforcing string types, and stripping extraneous characters) on the generated term before checking it against a predefined mapping dictionary. If a valid match is found, the term is transformed into the expected format with a persistent URI and English label. If no match is found, the subject is discarded.

3.3. Updating Records

The final module in the MOP architecture is the updater, which provides the LLM-generated metadata to OERSI. An API provided by OERSI is used for this purpose. Metadata imported via this API endpoint is made available immediately in the OERSI metadata and is also stored for future inclusion after subsequent update cycles. This process runs asynchronously to metadata harvesting in OERSI and therefore has no noticeable effect on that process. The MOP architecture has been designed in such a way that the output process can be changed relatively easily so that other output methods can be easily implemented in the future.

4. LLM Output

A screenshot showing how LLM-generated metadata is presented in the frontend of the OERSI test system is shown in Figure 4. Overall, the output quality of modern LLMs is high, and our pipeline produces fluently written and mostly relevant metadata for the majority of records. The LLM generally performs well enough for us to proceed to production with the system, given that users are clearly informed about the provenance of all LLM-generated content.[8] Moreover, each metadata record in OERSI includes a "Report record" button that opens a generic contact form with a freetext field. This creates a Gitlab issue which is then manually reviewed. In this way, users are able to easily provide feedback on the content.

Nevertheless, we observe several recurring issues that limit the reliability of generated content and highlight areas for further care. These issues include hallucinated content, inconsistencies, minor language errors, and difficulty generating labels from a controlled vocabulary.

4.1. Hallucinations and Other Errors

One of the most prominent issues are the so-called hallucinations, in which the LLM states as fact information that is not supported by the source material. This occurs in a range of forms, from minor inaccuracies to major conceptual misrepresentations, although these appear to be rare in our application. The model may fixate on a single example or element from the source material and present it as the primary topic, even when this is not reflective of the broader context. Other problems involve subtle omissions. In one example, the generated description of a resource tailored for a specific educational platform did not mention that platform at all. While not incorrect per se, such omissions reduce the specificity and utility of the result. We have also observed occasional spelling mistakes or strange phrasing, particularly when mixing German and English languages. While prompt tuning and better instruction design seem to have helped mitigate this, some such inconsistencies are expected to persist.

However, with the possible exception of the language errors, each of these issues still persists in even much larger state-of-the-art models [10]. As such, hallucinations, omissions, and other such inconsistencies are less isolated errors and more a systemic limitation of the current technology.

[8] See the Appendix for links to examples in production.

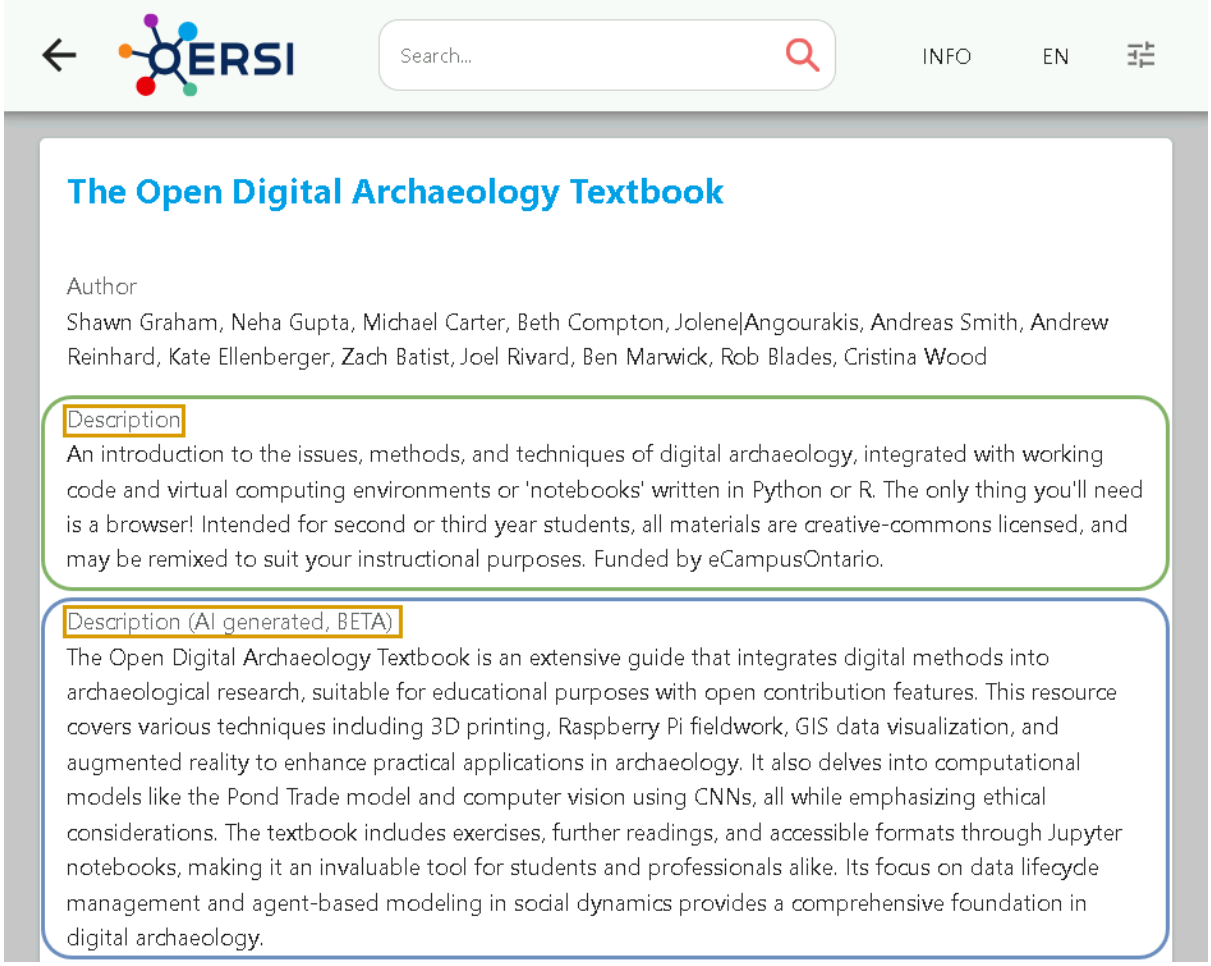

Figure 4: Frontend Display: Screenshot of an OERSI details page (from the test system) showing both the original description, in the green box, and the generated description, in the blue box, with the labels in orange. This screenshot has been cropped for brevity.

4.2. Subject Generation

In our case, subjects have proven to be the trickiest of the metadata fields to generate, due to their use of a controlled vocabulary. We use the Hochschulfächersystematik (or Higher Education Subject Classification) vocabulary, which is an extensive academic subject taxonomy for higher education[9]. When generating subjects by the LLM, only the top two levels of the hierarchy are included in the input, and the model is instructed to answer with only and exactly a term from this taxonomy. However, restricting LLM output to a controlled terminology via prompts remains an open area of research [11]. While effective at times, the labels generated by the LLM are often inaccurate. Sometimes the generated label is a valid subject term in our vocabulary, but an inaccurate classification of the resource. Other times, it generates a label which is not part of our vocabulary, and is therefore discarded. It may be that the meta-context of the request from within the sphere of OER metadata and the presence of so many subjects in the prompt is leading to unpredictable results. We have determined that LLM performance is still too varied in our system for this field to move to production.

5. Discussion and Lessons Learned

Although our automatic enrichment pipeline already delivers useful results, its deployment and scaling-up is still constrained by several factors. We describe them here to give the community a realistic

[9]It contains around 350 concepts. cf. https://w3id.org/kim/hochschulfaechersystematik/scheme

picture of the practical constraints that can limit the deployment and operation of LLM-based metadata pipelines.

5.1. Persistent Challenges

Language. One challenge we have encountered is the mixed use of languages across both resource content and metadata. As a Germany-based service indexing a variety of repositories with different intended audiences, OERSI contains resources which span a wide array of world languages, with English and German being the most common. It is not unusual for a resource to be in either English or German and its associated metadata to be in the other language. All enrichment prompts in our system are issued in English, but we aim to generate metadata in the same language as the resource content. This is particularly challenging given that even state-of-the-art LLMs can produce inconsistent output when handling mixed-language input [12]. One possible approach would be to translate prompts into the resource language, but this would introduce too much additional complexity at this time. Adapting the system for each supported language would also substantially complicate prompt engineering and validation logic. As a practical compromise, our current implementation restricts processing to English and German resources, allowing us to focus on high-quality generation and validation pipelines in the most common languages while acknowledging the need for future multilingual expansion.

Length. Controlling the length of generated metadata - particularly descriptions - remains an ongoing challenge in our enrichment pipeline. While the generated card descriptions are consistently the desired single concise sentence, generating substantially longer and more detailed content has proved inconsistent. This is particularly problematic for summaries, which serve as the foundation for generating all other metadata fields – making a high-quality summary a critical prerequisite for effective downstream generation. Prompting strategies to influence output length – such as requesting a specific number of words or sentences, or instructing the model to follow a particular structure (e.g., academic abstract) – have resulted in only minor and inconsistent changes. This suggests that prompt-based solutions alone are insufficient for achieving reliably longer and more detailed outputs.

Quality Control. Another open challenge is the task of evaluating the quality of the LLM output. We have followed an example-driven assessment during the development process, primarily relying on our own judgments, and organized one small workshop to solicit impressions from experienced colleagues. This manual evaluation is neither scalable nor reproducible. However, we have also found it challenging to identify quantitative metrics which meaningfully capture the "quality" of LLM-generated metadata, particularly for free-text fields like descriptions or titles, in which clarity, relevance, and informativeness are essential but hard to formalize. Automatic metrics such as BLEU or ROUGE could be computed, but our hypothesis is that lexical overlap would not meaningfully reflect quality in this case. Moreover, this would also require the development of a gold standard dataset. For now, we see a user study as the most practically feasible assessment method, although this is not an ongoing solution.

One potential suggestion is to ask an LLM to evaluate whether the generated metadata fits the resource content, and if not, flag that record for manual review. Recent research supports the feasibility of using LLMs to evaluate LLM output, but it's unclear whether these methods are ready for a production environment. And while in theory this approach would be scalable, it would also require adding yet another LLM request to our pipeline [13, 14].

GDPR Concerns. We have already taken note of several instances in which the LLM generated content with author names. Because we operate in Germany, we must consider General Data Protection Regulation (GDPR) laws concerning any personal data, which includes the names of authors, scholars, or any other individuals. While some contexts (e.g., summarizing a history lecture that names historical figures) are unambiguously allowed, other cases, such as incorrectly attributing ideas or generating misleading information about real people, pose legal and ethical risks. Currently, there is no automated mechanism to reliably detect such occurrences, as the acceptability of name mentions is highly context-dependent. For now, we provide users with a generic "Report record" function to flag problematic content, and will blacklist individual records from LLM content generation if necessary.

5.2. Evolution of MOP

Structured vs Unstructured Response. In the initial implementation of our pipeline, we generated all metadata fields via a single request, instructing the model to return a well-formed JSON object. However, this approach proved to be impractical as the model produced outputs that deviated from the expected structure. Extracting and validating the relevant fields required brittle post-processing logic that, when it failed, affected the entire resource. As a result, we transitioned to issuing separate requests for each individual metadata field. While this increases the total number of LLM calls per resource, it also minimizes the impact of failure – if one request fails or produces invalid output, the rest of the metadata for that record remains unaffected.

This also simplified prompt engineering, as each prompt can be tailored to a specific field without the need for complex formatting constraints or the inclusion of input data which is not relevant for all fields. Additionally, when generating metadata fields such as titles or keywords, we found that supplying too much information tended to produce generic or vague outputs. Our best results were achieved by providing only the generated summary and the resource language, which likely helped the model focus on salient content without distraction.

Full Text vs Summary. Our initial pipeline used the full text of each resource as input for all LLM-generation tasks. However, we found that using the full text could degrade the quality of the generated metadata: for concise fields like descriptions or keywords, too much input led to vague, overly generic, or noisy outputs. To address this, we introduced an intermediate summarization step. Now, we first generate a condensed summary of the full text, which is then used as input for subsequent metadata generation. We found that this improves output quality by focusing the model's attention on the most salient information, although the system still requires some refinement – especially in generating longer and more detailed summaries (see subsection 5.1 for more discussion around output length).

One point of discussion this has raised is the difference between a summary and description. When considering LLM output, for videos only a few minutes long or text resources of a few pages, there may be minimal differentiation between a summary and a description. However, for longer resources, such as lecture videos or textbooks, the summary should contain substantially more detail. Aside from length, there is still another distinction to be made between a summary and a description: summaries should capture the structure, scope, and key content points of a resource; a description may contain all of those points, but it might also describe the resource at a higher level: framing its purpose, audience, or relevance. In other words, even the LLM-generated description behaves more like a summary. For now, we maintain the label of "description" for this generated field but may rename it.

Subject Classification. We have experimented with providing the subject taxonomy in various formats and states of completeness. Initially, we supplied the full subject hierarchy as a JSON structure, including both subject labels and their internal IDs. However, this approach led to confusion in the outputs: the model sometimes mixed up labels and IDs or failed to choose appropriate subjects, likely due to the length and complexity. We then tested a simplified version using only the top level of the taxonomy and without IDs, which improved output significantly. Currently, we pass the top two levels of the taxonomy without IDs, and map the strings to IDs within our module. One possible solution to address this is via an iterative classification approach: first generate a general subject, and then narrow down to a subfield by making a new request passing only the subfields under that subject.

Evaluator Class. To support internal development and iteration on prompt design, we introduced a lightweight evaluator class for assessing the generated metadata. This component is not part of the core MOP architecture but serves as a development utility. Its primary function is to calculate the mean length of generated content for a given metadata field, in order to compare the effects of prompt variations or model configurations over time. Length may also serve as a useful proxy for certain goals, such as ensuring summaries are sufficiently informative.

Display of Generated Content. In designing the presentation of the LLM-generated metadata in the frontend system, we considered how to balance clarity, usability, and transparency. Some initial ideas were separating fields into tabs (e.g., toggling between original and generated content) and making

LLM-content visually distinct in cards using a colored background. In the end, we opted for a simpler but clearly demarcated approach. All content is shown on the detail page, with LLM-content shown following the original content and a label clearly marking it as AI-generated. An example can be seen in the screenshot in Figure 4.

Availability of Generated Content. As described in subsection 3.3, the schema used by OERSI is the product of a joint project with multiple stakeholders. Therefore, the question of whether and how to expose the LLM-generated content via the API was also a point of discussion. We recognized that limiting access to the frontend could create confusion, as users who see metadata in the interface may reasonably expect to retrieve it via the API as well. From a transparency and usability standpoint, we concluded that making the enriched metadata available through the API is preferable. However, this change requires modifying the OERSI schema. As OERSI is a collaborative project, any schema change must be evaluated for potential impact on existing infrastructure and approved by relevant stakeholders. Our current plan proposes adding a separate `generated_content` object to the OERSI-specific schema and introducing an optional API parameter that allows clients to explicitly include or exclude LLM-generated fields in their results.

5.3. Resources Required

Compute availability remains the principal obstacle to production-ready deployment of our metadata generation pipeline. As a publicly funded institute, we operate under certain budget constraints, so procuring a dedicated GPU with more than 24GBs of VRAM is currently infeasible. Reliance on shared infrastructure has already exposed hard limits: during the first week of May 2025, every job submitted to the internal shared GPU cluster failed because the queue was saturated, halting LLM content generation for several days. Community services such as Ollama offer an interim fallback, yet our throughput is tightly coupled to external demand patterns we cannot influence. While we have recently been granted an allocation on the academic HPC system Kisski to supplement our internal resources, the usable quota is still fairly limited. In consequence, we cannot sustain the GPU-hours required for the full end-to-end workflow on all 90,000 some resources indexed in OERSI; instead, we have reduced scope to an MVP that generates descriptions only for a curated, impact-ranked whitelist of resources.[10] We feel that such issues are not unique to us or our use case, and sharing them with the community also sheds light on the real-life applicability and constraints that users face.

6. Related Work

Several recent studies combine LLMs or NLP with knowledge graphs to enrich, generate, or validate metadata, or to align schemas. Kumar et al. propose an enterprise framework that uses LLMs to unify heterogeneous data sources into an activity-centric knowledge graph, automating entity/relation extraction and "semantic enrichment" [15]. Taboada et al. introduce MILA [16], an LLM-based ontology matching pipeline that uses a retrieve-and-prompt strategy to align schema entities. Others argue that LLMs can significantly accelerate core KG and ontology engineering tasks [17], including modeling, alignment, and population. Leal et al. [18] proposed an LLM-based zero-shot approach for named entity linking in educational texts, using retrieval-augmented prompts to connect content to knowledge organization systems. In library contexts, LLMs have been used to generate descriptive metadata; for instance, Huang et al. [19] applied GPT-4 to web archive collections, auto-generating titles and abstracts. They reported cost savings but also noted that LLM outputs can be lower quality than human-curated metadata and that hallucinations remain a challenge.

In the education domain, LLMs are also applied for metadata generation. Viswanathan et al. [20] used GPT-4 to segment and summarize lecture transcripts, extracting learning objectives, key definitions, and questions. Beyond generation, a key challenge is addressing metadata consistency and integrating heterogeneous educational resources. Tavakoli et al. [21], analyzing OER metadata, underscored that

[10] See the Appendix for links to examples in production.

high-quality, consistent metadata is critical for search and recommendation. The heterogeneity of educational metadata, with diverse schemas and vocabularies, poses a persistent problem for federation, leading to issues like imprecise term definitions and incomplete conventions [22]. Linked Data approaches have sought to address this interoperability: for example, mEducator [23] demonstrated publishing OERs in RDF and linking to external vocabularies, while Pereira et al. [24] surveyed how Linked Data can enhance resource interoperability and personalization. Integration projects like LinkedUp [25] have aimed to aggregate learning data into unified KGs. Similarly, Telnov et al. [26] implemented a semantic educational portal using RDF triplestores. Liang et al. [27] also tackled metadata harmonization for language resource repositories using Linked Data. Despite these efforts, aligning schemas across diverse domains often requires considerable mapping or transformation.

Our work with MOP extends these lines of research by combining LLM-based content analysis with semantic integration mechanisms. It aims to automatically infer metadata from resource content, demonstrating a practical application effective even within resource-limited public institutes.

7. Conclusion

In this paper we have presented MOP, our domain-agnostic intermediate layer for automatically generating metadata fields tailored for incorporation into scientific knowledge graphs. The modular architecture is comprised of three primary components: the loader, the generator, and the updater. The loader module fetches metadata from a data source – we use OERSI, a federated knowledge graph system, as an example use case – and then extracts the full text of the resource via its direct download link. In the next step, the generator module queries an LLM to generate a summary of the full text for use as input in all downstream tasks. Subsequently, the LLM generates any of a configurable set of metadata fields, which are validated against a predefined JSON schema to ensure compliance with the knowledge graph schema. Finally, the updater module uses an API provided by OERSI to update the LLM generated fields within the system. This process runs asynchronously to OERSI's metadata harvesting process, allowing for flexible updates. Although development has been done using OERSI and focusing on OER, the code, available in its entirety on GitLab, is fully reusable and adaptable to other usage scenarios.

While our workflow is already productive and deployed on our production system, the coverage has been complicated by the availability of compute resources. The resources required for the entire pipeline to run over the more than 90,000 resources indexed in OERSI is not available to us at this time. Therefore, we have launched our MVP on our production system with a small number of resources enriched with descriptions, with more to be added on an ongoing basis. (See the Appendix for links to some examples.) If and when additional infrastructure becomes available, the selection of resources and metadata fields can easily be expanded. Additional future work includes refining the generation of subject labels and potentially customizing LLMs with QLORA [28], which could lead to better results for our use case with little need for more compute resources.

We share our insights and experience in this work to shed more light on how language models can be applied to enhance knowledge graph metadata, used on a smaller scale with limited resources, and with this highlight both the limitations and the feasibility.

Acknowledgments

We thank Yaser Jaradeh, Allard Oelen, and Sebastian Peters for generously consulting on this project and sharing their expertise and experience.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: polish sentences, rephrase. Further, the authors used Gemini in order to: search for and/or explain relevant literature. After using

these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. Mishra, Open educational resources: Removing barriers from within, Distance education 38 (2017) 369–380.

[2] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it's done, Queue 17 (2019) 48–75.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[4] F. Marulli, L. Campanile, M. S. de Biase, S. Marrone, L. Verde, M. Bifulco, Understanding readability of large language models output: an empirical analysis, Procedia Computer Science 246 (2024) 5273–5282.

[5] C. Bizer, T. Heath, T. Berners-Lee, Linked data: Principles and state of the art, in: World wide web conference, volume 1, Citeseer, 2008, p. 40.

[6] O. Suominen, C. Mader, Assessing and improving the quality of skos vocabularies, Journal on Data Semantics 3 (2014) 47–73.

[7] A. Zouaq, F. Martel, What is the schema of your knowledge graph? leveraging knowledge graph embeddings and clustering for expressive taxonomy learning, in: Proceedings of the international workshop on semantic big data, 2020, pp. 1–6.

[8] B. L. Mbaiossoum, How to choose the best AI LLM: A guide to navigating the diversity of models, J. Inf. Syst. Eng. Manag. 10 (2025) 221–232.

[9] B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 1 (2020) 3.

[10] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, ACM Trans. Inf. Syst. 43 (2025). URL: https://doi.org/10.1145/3703155. doi:10.1145/3703155.

[11] B. Li, Y. Wang, T. Meng, K.-W. Chang, N. Peng, Control large language models via divide and conquer, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 15240–15256. URL: https://aclanthology.org/2024.emnlp-main.850/. doi:10.18653/v1/2024.emnlp-main.850.

[12] L. Zhang, Q. Jin, H. Huang, D. Zhang, F. Wei, Respond in my language: Mitigating language inconsistency in response generation based on large language models, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 4177–4192.

[13] Z. Guo, R. Jin, C. Liu, Y. Huang, D. Shi, L. Yu, Y. Liu, J. Li, B. Xiong, D. Xiong, et al., Evaluating large language models: A comprehensive survey, arXiv preprint arXiv:2310.19736 (2023).

[14] R. Awasthi, S. Mishra, D. Mahapatra, A. Khanna, K. Maheshwari, J. Cywinski, F. Papay, P. Mathur, Humanely: Human evaluation of llm yield, using a novel web-based evaluation tool,

medRxiv (2024). URL: https://www.medrxiv.org/content/early/2024/12/14/2023.12.22.23300458. doi:10.1101/2023.12.22.23300458.
[15] R. Kumar, K. Ishan, H. Kumar, A. Singla, Llm-powered knowledge graphs for enterprise intelligence and analytics, arXiv preprint arXiv:2503.07993 (2025).
[16] M. Taboada, D. Martinez, M. Arideh, R. Mosquera, Ontology matching with large language models and prioritized depth-first search, arXiv preprint arXiv:2501.11441 (2025).
[17] C. Shimizu, P. Hitzler, Accelerating knowledge graph and ontology engineering with large language models, Journal of Web Semantics (2025) 100862.
[18] R. Leal, A. Ahola, E. Hyvönen, Using llms for enriching metadata with links to kos and knowledge graphs: Case finnish named entity linking (2024).
[19] A. Y. Huang, A. Nair, Z. R. Goh, T. Liu, Web archives metadata generation with gpt-4o: Challenges and insights, arXiv preprint arXiv:2411.05409 (2024).
[20] S. Asthana, T. Arif, K. C. Thompson, Field experiences and reflections on using llms to generate comprehensive lecture metadata, in: NeurIPS'23 workshop on generative AI for education (GAIED), 2023.
[21] M. Tavakoli, M. Elias, G. Kismihók, S. Auer, Metadata analysis of open educational resources, in: LAK21: 11th International Learning Analytics and Knowledge Conference, 2021, pp. 626–631.
[22] G. Alemu, B. Stevens, P. Ross, Semantic metadata interoperability in digital libraries: a constructivist grounded theory approach, in: ACM/IEEE Joint Conference on Digital Libraries, Ottawa (Canada), volume 13, 2011, pp. 7–16.
[23] S. Dietze, D. Taibi, H. Q. Yu, N. Dovrolis, Al inked d ataset of medical educational resources, British Journal of Educational Technology 46 (2015) 1123–1129.
[24] C. K. Pereira, S. W. M. Siqueira, B. P. Nunes, S. Dietze, Linked data in education: A survey and a synthesis of actual research and future challenges, IEEE Transactions on Learning Technologies 11 (2017) 400–412.
[25] E. Herder, S. Dietze, M. d'Aquin, Linkedup-linking web data for adaptive education., in: UMAP Workshops, 2013.
[26] V. P. Telnov, Semantic educational web portal, in: CEUR-WS, ANALYTICS AND DATA MANAGEMENT IN DATA-INTENSIVE FIELDS, 2017, pp. 80–86.
[27] Z. Liang, Harmonizing metadata of language resources for enhanced querying and accessibility, in: 2024 5th International Conference on Computers and Artificial Intelligence Technology (CAIT), IEEE, 2024, pp. 642–650.
[28] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: efficient finetuning of quantized llms, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Curran Associates Inc., Red Hook, NY, USA, 2023.

A. Example Records

As MOP has only recently been integrated into the production system, there are currently few records with LLM-generated content available. For convenience, we provide links here to collections in which all (or, in the final case, most) records contain this content.

- Collection of texts and reference works about archaeology and ancient history.
- Collection of videos explaining German laws around data privacy and security.
- Collection of videos demonstrating principles of chemistry.
- Collection of textbooks about medical terminology.

B. LLM Comparison

We performed a small comparison of three LLMs which were recommended for our use case by the LLM Integration Lead of our institute. The full output results from this comparison are available here.