Humanities & Social Sciences Communications

ARTICLE

https://doi.org/10.1057/s41599-025-04491-x OPEN

Building sustainable information systems and transformer models on demand

Thomas Asselborn [1,2]✉, Sylvia Melzer [1,2]✉, Simon Schiff[3], Magnus Bender [1], Florian Andreas Marwitz [1,2], Said Aljoumani[2], Stefan Thiemann[4], Konrad Hirschler[2] & Ralf Möller[1]

The growing practice of archiving research data in repositories reflects an upward trend. However, storing data in an RDR (Research Data Repository) does not guarantee that the archived data will always be readily reusable, even if this fulfils the FAIR (Findable, Accessible, Interoperable Reusable) principles. To ensure sustainable RDM (Research Data Management), archiving must consider the future potential for data reuse in a low-threshold fashion. In this article, we demonstrate the utilisation of straightforward methods to implement a so-called warm or hot archiving for research data within an RDR, as opposed to the conventional cold archiving approach. We explore the additional value of using research data in the humanities, emphasising the advantages of maintaining data accessibility and relevance over time. In the humanities, evaluating numerous data sets efficiently is crucial for current and future projects. Reviewing and evaluating relevance is important, particularly when dealing with a substantial number of data sets. Rapid evaluation facilitates profound decisions on the utility of the data for one's ongoing or upcoming projects. For hot archiving, this means that in addition to the research data, the data should be available in a human-friendly way, i.e., a viewer application to visualise the data should be easily accessible. However, as rapid developments in the IT sector mean that after a few years, it cannot be guaranteed that these viewers or other tools will work, we also show how data can be viewed in a user-specific way via the RDR and how sustainable viewing can be integrated into the RDR. This article presents a generic approach to building sustainable viewers, which we call information systems, or transformer models on demand using data from pre-modern Arabic. In addition, we show that the easy-to-use chatbot ChatGPT can alternatively be context-specifically prepared to deliver more precise results and associated resources in the field of humanities. On the one hand, we have achieved a substantial reduction in the development time of an information system, from months to seconds, as well as the ability to fine-tune BERT (Bidirectional Encoder Representations from Transformers) models without specific knowledge in selecting models or tools. On the other hand, we have developed a chatbot that not only provides project-specific responses but also references the sources.

[1] Institute of Humanities-Centered Artificial Intelligence (CHAI), University of Hamburg, Hamburg, Germany. [2] Centre for the Study of Manuscript Cultures (CSMC), University of Hamburg, Hamburg, Germany. [3] Institute of Information Systems (IFIS), University of Luebeck, Schleswig-Holstein, Germany. [4] Center for Sustainable Research Data Management, University of Hamburg, Hamburg, Germany. ✉email: thomas.asselborn@uni-hamburg.de; sylvia.melzer@uni-hamburg.de

Introduction

In the digital era marked by the increasing significance of digital information systems, the ability to create sustainable and reusable solutions on demand has become important. In the field of humanities, a wealth of data has already been curated within RDRs (**R**esearch **D**ata **R**epositories), signifying a substantial resource. While archiving research data in an RDR typically adheres to the FAIR (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable) principles, their reusability is not guaranteed. This is due to constraints such as i) time limitations for converting data sets into the required formats for analysis, ii) challenges in getting the specific tools to function, or iii) a lack of knowledge about which tools are suitable for the task. Reusing data contributes to sustainability, in contrast to data that is merely archived and no longer needed. Such unused or minimally used data is defined as *cold* data, constantly used data is called *hot* data, and *warm* data as data with regular access (Pernet et al. 2023). Therefore, a sustainable approach involves considering what to do with data as they become colder over time.

On-demand information systems play a crucial role in ensuring easy access to research data, effectively mitigating the concern of data deterioration over time. Researchers can quickly access and use their data when needed, which is critical for long-term usability and sustainability (Melzer et al. 2023). These systems can provide custom data visualisation capabilities that facilitate the communication of research results and data-driven insights. An important aspect of on-demand information systems lies in their user-centric orientation, which places the needs and preferences of researchers at the forefront (Schiff et al. 2022). Building an ISoD (Information System on Demand) encourages researchers to actively use and maintain their data and promotes a sense of responsibility for its sustainability. These systems also promote interdisciplinary collaboration by facilitating cooperation between researchers from different fields, leading to more comprehensive and sustainable RDM (**R**esearch **D**ata **M**anagement) strategies (Peukert et al. 2023).

This article describes how information systems can be built on demand to ensure data reusability by providing tools and infrastructure to reuse archived data. This minimises duplication of effort and improves the sustainability of data resources. The systems are designed to be scalable, adapt to changing research needs, and accommodate the growing volume of research data.

In this article, we show how manifest files like METS (Metadata Encoding & Transmission Standard) can be used to save the configuration data and the appropriate viewers for a data set and to automate the execution of the FToD (Fine-Tuning on Demand) process from the RDR.

The utilisation of transformer models for analysing historical data, such as pre-modern Arabic texts, offers a new opportunity to use information systems as a source to fine-tune models such as BERT (Devlin et al. 2018). The challenge, however, is to enable scholars themselves to fine-tune models. There are already tools that are user-friendly and efficient for fine-tuning user-specific models, such as "AutoTrain" developed by HuggingFace[1], which is not for specific user groups, while the approach presented in Asselborn et al. (2023) is for RDR users. To enable researchers to use this function, we argue that it should be possible to perform such fine-tuning directly from an RDR. In addition, the corresponding configuration parameters, such as the appropriate labels used for a model, should be archived together with the research data. Correct labelling is crucial to ensure that the results are improved (and not degraded) after fine-tuning. For example, in Arab countries, names are often adapted to modern naming conventions, featuring a first name followed by a surname (e.g. first name: *Hamza*, surname: *ibn Omar ibn Mustafa*). However, early Arabic texts, as well as texts from various global regions, exhibit diverse naming patterns. These names may encompass tribal affiliations, place of origin, or even associations with a legal school or occupational title. Person names can also include a place name, requiring appropriate labelling. Additionally, places are sometimes paraphrased, e.g. *the Golden Mosque*. Furthermore, indicating a date follows a different format than the Gregorian calendar, as illustrated in the following sentence: *"Hamza ibn Omar ibn Mustafa went to Marrakesh on the ninth day of the month of Shawwal."*

Chatbots like ChatGPT are well-suited for users who prefer to interact with a data set by formulating queries in natural language. However, ChatGPT or other chatbots may not consistently provide correct answers. These chatbots can be improved by including project-specific content to enhance the rate of correct answers. Some tools have already been implemented, such as *h2oGPT*(H2O.ai 2023), so that such a chatbot can meet the needs of various users from different disciplines. Project-specific sources must also be given as citations for the answers. Especially in the humanities, the citation of texts is a core element of analyses (Schiff and Möller 2023). The data from the RDR, which is used as input by tools like ChatGPT, and the answers contain additional citations, is also a sustainable approach. In the humanities, it is common for annotations to be added to the texts, which can be individual words or short texts that must also be considered in the analysis. If, for example, there are two dates for an artefact, an explanation of the dating is usually added in a commentary field, which can be seen as an annotation. In this article, we introduce ChatHA (Humanities Aligned Chatbot), which provides answers with source references and utilises annotations in the response process to deliver more personalised (subjective) results.

The synergy between on-demand information systems, transformer models, and data reuse strategies exemplifies a comprehensive and forward-thinking approach to sustainable RDM.

In this article, we present the three concepts ISoD (Section "Information systems on demand"), FToD (Section "Fine-tuning on demand"), and ChatHA (Section "ChatHA") of making (research) data reusable. Along with the introduction of the three concepts, we also provide answers to the following key questions. First, how can research data be made available in a human-friendly way for hot archiving, considering the potential obsolescence of viewer applications over time? Second, what methods can be implemented to ensure sustainable warm or hot archiving within RDM systems and RDRs? Third, what generic approach can be used to build sustainable viewers and transformer models on demand? Fourth, how can tools like ChatGPT be context-specifically prepared to deliver more precise results and associated resources in the field of humanities? Fifth, how can chatbots be developed to provide project-specific responses and reference sources? Finally, how can development time for information systems be reduced, and BERT models be fine-tuned without requiring specialised knowledge in model selection or tools?

This article is structured as follows: Preliminaries that can help in understanding the concepts presented in the article are provided in Section "Preliminaries". Section "Information systems on demand", Section "Fine-tuning on demand", and Section "ChatHA" introduce the concept of ISoD, FToD, and ChatHA. The application and the results of the three concepts are described in Section "Application and results". Finally, Section "Conclusion and outlook" presents the conclusion and outlook for developing sustainable information systems and transformer models on demand.

Sustainable research data

Research data are all data generated within the framework of a scientific project (Thiemann 2019) and is a collection of information gathered through various methods and techniques. The data can be classified into different types based on the methods

used for collection, the characteristics of the retrieved information, and the way it is presented. Types of research data are statistics, results of experiments, measurements, observations resulting from fieldwork, survey results, interview recordings, and images (EUR-Lex 2019).

With national and international authorities issuing a series of reports, policies and guidelines to deal with big data in research institutions, university libraries have responded by helping researchers meet the new requirements. The requirements include, for example, facilitating access to data (Pryor 2012). Since then, strategies have been developed for efficient and effective RDM at research institutions. A contemporary focus within the realm of RDM is to make research data sustainable; in other words, sustainability of research data refers to their long-term preservation, accessibility, interoperability, and warm or hot reuse. While books in libraries can be preserved in the long term, digitised resources face the challenge of ensuring long-term sustainability (Lavoie 2012).

At the University of Hamburg, the sustainability of research data is ensured by uploading research data to the university's RDR (short: RDR@UHH) because it is typically designed to provide long-term preservation for uploaded data. Robust data backup and storage strategies ensure that data in the RDR remains intact and accessible over a long period, often far beyond the completion of a research project. In addition, RDRs are structured to offer consistent and reliable accessibility to authorised users, including researchers or the public. Sustainable research data often adhere to common data standards and formats, making it easier for different researchers and systems to work with the data. This interoperability, in principle, promotes the continued usability of the data, even as technologies and tools evolve. However, it also happens that scripts illustrating research data can become outdated or no longer exist. We therefore propose to archive these scripts in the RDR as well. This way, researchers can look at the data at a later point in time or even continue to work with it efficiently without having to rebuild the scripts first. In this article, we show how data can be reused from an RDR to visualise research data, rebuild an information system and fine-tune transformer models.

Related work

An innovative approach known as DoD (Database on Demand) has been devised specifically for proteomics researchers, offering an effortless method to generate custom databases. PRIDE (Vizcaíno et al. 2011) stands as one of these DoD systems that allows users to construct databases, yet its scope is confined to proteomics. Nonetheless, it is essential to note that PRIDE does not operate automatically by extracting data from an RDR. Instead, it involves a more manual and customised effort on the part of users. Despite this, the concept of DoD has proven its adaptability, extending its utility beyond proteomics to other disciplines, including the humanities; see (Schiff et al. 2022). For scholars in the humanities, the DBoD (DataBasing on Demand) process was successfully adopted because it was beneficial for their research endeavours (Melzer et al. 2022). In general, automating the DBoD approach is therefore forward-looking and supports the sustainable use of research data from an RDR.

In various fields, LLM (Large Language Model)s are commonly used to fine-tune existing models to obtain more accurate answers to user queries. Fine-tuning, a process that involves adjusting model parameters and training on domain-specific data, helps optimise the performance of these models. Additionally, for humanities scholars, there is a desire to automate the fine-tuning process directly from an RDR, where research data is archived. The automated process would enable users to effortlessly fine-tune models, for example, with a simple button click, eliminating the need for in-depth involvement in the fine-tuning procedures and tools. The function "AutoTrain" developed by HuggingFace is a forward-looking approach. However, users need to familiarise themselves a little to perform the fine-tuning. In this article, this approach is a bit more simplified so that a user can perform the fine-tuning directly by selecting the data set in an RDR with one button click.

ChatGPT is suitable for generating human-like texts and conducting conversations in natural language. Humanities scholars can ask questions and follow-up queries in this form and receive answers. While ChatGPT does not include the source information, Perplexity.ai[2] does. However, these references do not refer to user-specific sources. H2O.ai developed the tool *h2oGPT*(H2O.ai 2023) to display user-specific sources in the search results. In this article, we present a similar approach by employing SCDs (Subjective Content Descriptions) (Kuhr et al. 2019) so that existing (subjective) annotations are also integrated into the query evaluation to obtain refined search results.

Preliminaries

The following preliminaries provide some foundational concepts relevant to the approaches presented. This section is more technical and may be skipped for now. The abbreviations and concepts used in the following sections can be looked up here.

Definition 1. **R**esearch **D**ata **R**epository (RDR)

An RDR is a storage location for digital objects and provides access to research data. It serves as a secure and organised storage solution for data generated or used in scientific research, ensuring that the data is preserved, findable, and accessible for future use and analysis. Examples of RDRs are Zenodo (Research and OpenAIRE 2013) and the RDR@UHH (Universität Hamburg 2022).

Both the FToD process and ChatHA are based on transformer models. Thus, some terms related to transformer models need to be introduced first.

Definition 2. Transformer model

Transformer models, or transformers in short, are computational architectures capable of automatically converting one form of input into another form of output. These models were first introduced in (Vaswani et al. 2017) in 2017. Transformers are particularly useful for processing, e.g. sequential data, such as natural language text, because they can learn context and meaning by tracking relationships between elements in the sequence.

Definition 3. **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (BERT)

BERT is a transformer model introduced by Devlin et. al. (Devlin et al. 2018) in 2019. BERT comprises a stack of multiple encoder blocks that harness the attention model to achieve a deeper comprehension of language context. The input text is divided into tokens, and each token gets transformed into a vector within BERT's output. BERT is trained using both the MLM (Masked Language Model) for language generation and the NSP (Next Sentence Prediction) task concurrently. The MLM modifies the traditional language model to consider the bidirectional context in its prediction. For each sequence, 15% of tokens are replaced with a [MASK] token. The model is then trained to predict the masked words based on the context provided by the remaining non-masked words in the sequence. The NSP task uses the output embedding of the first token of the input sequence ([CLS]) that prepends the sequence to predict whether the second sentence follows the first or is from a different document. (Song et al. 2019)

Definition 4. **G**enerative **P**re-trained **T**ransformers (GPT)

GPT (Generative Pre-trained Transformer) is a family of large-scale language models developed by OpenAI for NLP (Natural Language Processing) tasks. GPT models are based on the Transformer architecture and are pre-trained on massive amounts of text data using unsupervised learning. The pre-training process involves predicting the next word in a sequence of text, given the previous words. (Radford et al. 2018)

Definition 5. ChatGPT

ChatGPT is a chatbot that uses an advanced language model developed by OpenAI, designed to generate human-like text and engage in natural and coherent conversations with users. The public version at the time of writing this article is built upon the GPT-3.5 architecture. It can maintain context in conversations and respond in a contextually relevant manner, making it suitable for a wide range of applications. It can also be fine-tuned for specific tasks, domains, or industries, allowing developers to tailor ChatGPT's responses to particular applications and user needs. However, it should also be noted that ChatGPT's answers are not necessarily correct.

Definition 6. Llama 2

Llama 2 is an open-source LLM developed by Meta (Touvron et al. 2023). It is similar to GPT, but the weights are open-sourced, so we can use the Llama 2 model more easily. In the following, we will identify Llama 2 with the term Llama.

Definition 7. **R**etrieval-**A**ugmented **G**eneration (RAG)

RAG (Retrieval-Augmented Generation) is a method for providing additional information to an LLM (Lewis et al. 2020). As a pre-processing step, RAG indexes a corpus into a vector database by embedding. The original query is modified before asking the LLM: The query is embedded as well, and the k nearest sentences to it in the vector database are extracted and prepended to the query as context. The developer chooses a whole positive number value for k and determines the most suitable one through experimentation. We can also apply RAG in another way: Instead of augmenting the query, we can augment the output. In this case, the output of the LLM is embedded, and the nearest sentences are extracted and appended to the output. We use the first method throughout this paper.

Definition 8. HuggingFace Transformers library

HuggingFace Transformers is a library developed by HuggingFace[3] allowing easy download, use and upload of transformer models (Wolf et al. 2020).

Definition 9. **S**ubjective **C**ontent **D**escriptions (SCDs)

For ChatHA, we will use SCDs (Kuhr 2022; Kuhr et al. 2019). An *SCD* can be thought of as a sticky note attached to a sentence in a document. Such a sticky note contains additional content, which can be of any form, e.g. text, images, or audio. An SCD can be attached to multiple similar sentences if this SCD can cover all the sentences' contents. The referenced sentences are referred to as the *windows* of the SCD. Each SCD is associated with a probability distribution of words formed over its windows. Multiple SCDs build an *SCD matrix* in which each SCD covers a row with its distribution. Thus, an SCD can be seen as being associated with a probability distribution over words. The USEM (Unsupervised Estimator of SCD Matrices) (Bender et al. 2023) algorithm can be used to infer SCDs from a *corpus*, that is, a set of documents. Later, we can use the MPS^{2}CD (Most Probably Suited SCD) algorithm to find the most probable SCDs for a given sentence by finding the SCDs with the highest probability of being generated by the given sentence and SCD matrix. In summary, SCDs can be automatically estimated for a corpus, and later, these SCDs can be associated with new sentences. Hence, the SCDs associated with the new sentences build references to the initial corpus.

For the execution of a process on demand, we use a configuration file, here in METS format, which will be introduced in more detail.

Definition 10. **M**etadata **E**ncoding & **T**ransmission **S**tandard (METS)

Manifest files contain optional metadata (configuration data) for the associated data set and are available in XML format. METS is a recognised standard that facilitates the exchange of digitised documents among cultural heritage institutions. It operates as an XML schema specially designed for the generation of digital objects. In this context, a digital object may encompass one or multiple digital files, potentially in varying formats, and can provide an intricate internal structure description. A METS file consists of seven major sections. Here, an excerpt is presented of the fields important for ISoD (see Section "Information systems on demand") or also FToD (see Section "Fine-tuning on demand"). Other fields are described in the standard [4].

METS Header: The METS Header is a section within the METS file that contains metadata about the METS file itself. It includes information such as the creator and editor of the file (see Listing 1).

Listing 1: METS Header

```
<mets:metsHdr CREATEDATE="2023-10-16T09:30:00"
              RECORDSTATUS="Complete">
 <mets:agent ROLE ="CREATOR"
             TYPE="INDIVIDUAL">
  <mets:name>
   Konrad Hirschler
  </mets:name>
 </mets:agent>
 <mets:agent ROLE ="ARCHIVIST"
             TYPE="INDIVIDUAL">
  <mets:name>
   Invenio Packager Webapp
  </mets:name>
 </mets:agent>
</mets:metsHdr>
```

Descriptive Metadata: The descriptive metadata section <dmdSec> in METS can either point to external descriptive metadata sources or contain internally embedded descriptive metadata. It provides information about the digital object being described, such as its title, author, and subject (see Listing 2).

Listing 2: metsdmdSec

```
<mets:dmdSec ID="DC">
 <mets:mdWrap MIMETYPE ="text/xml"
              MDTYPE="DC"
              LABEL="Dublin Core Metadata">
  <mets:xmlData>
   <dc:dc>
    <dc:creator>
     Konrad Hirschler
    </dc:creator>
   </dc:dc>
  </mets:xmlData>
 </mets:mdWrap>
</mets:dmdSec>
```

File Section: The file section in METS lists all the files that comprise the digital object. These files are represented by < file > elements, which can be organised into < fileGrp > groups based on different versions or categories of the object. Additionally, it includes functions, for example, what data is to be harvested (e.g. CSV file) (see Listing 3).

Listing 3: File Section

```
<mets:fileSec>
 <mets:fileGrp ID ="media">
  <mets:file ID="acp.csv"
             MIMETYPE="text/csv"
             SIZE=148826>
   <mets:FLocat LOCTYPE="URL"
                xlink:href="acp.csv">
   </mets:FLocat>
  </mets:file>
 </mets:fileGrp>
</mets:fileSec>
```

Structural Map: The structural map < structMap > is a crucial component of a METS file. It outlines the hierarchical structure of the digital library object, defining the relationships between different elements. It also links the elements to the corresponding content files and metadata. The structural map is built in a tree-like structure using multiple nested DIV elements, with the root DIV element containing all other DIV elements (see Listing 4).

Listing 4: Structural Map

```
<mets:structMap TYPE ="logical">
 <mets:div TYPE ="root">
  <mets:fptr FILEID="acp"/>
 </mets:div>
</mets:structMap>
```

Behavior: The < behaviorSec > section in METS enables the association of executable behaviours with content in the METS object. Each behaviour is defined by an interface definition element, representing an abstract definition of a set of behaviours. Additionally, each behaviour has a mechanism element that identifies the executable code module responsible for implementing and running the defined behaviours (see Listing 5). For instance, in Listing 5, the viewer Heurist is chosen, and the process ISoD will be executed.

Listing 5: Behavior to execute the ISoD process and present the data in a Heurist database instance

```
<mets:behaviorSec>
 <mets:behavior ADMID="heurist">
  <mets:mechanism LABEL="isod"/>
 </mets:behavior>
</mets:behaviorSec>
```

Information systems on demand

Information systems play a crucial role in today's digital age, not only in business, government, education, and healthcare but also in the humanities. In (Laudon and Laudon 2013), information systems are defined from a technical perspective "as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision-making and control in an organisation." The humanities scholars increasingly store their collected research data in RDRs. To ensure long-term integrity and facilitate reproducibility of research data, we have recognised the need for a more tailored but generic approach. This approach envisages loading data directly from an RDR, e.g. into a database and creating an ISoD on top of the database. We have developed a DBoD framework to simplify the processes. With this generic framework, users can seamlessly access, configure, and create customised, project-specific database instances on demand, all while requiring minimal resources. This framework allows researchers to create information systems based on the database instances that meet their individual research and academic needs. The tool Heurist (HEURIST 2022), an open-source database management system with a web front-end, allows researchers without prior IT knowledge to develop data models, store data, search, and publish data on a website. In some projects, research data in hundreds to thousands of JSON or TEI (Text Encoding Initiative) files or other machine-readable formats could be transferred to such a database instance of Heurist using the DBoD process (Melzer et al. 2022). During the DBoD process, the data is read from archived formats and converted into the Heurist XML (HML) format, which is then manually loaded into a database instance. A website also needs to be created manually to become an information system from the database instance. To simplify the process and automate the creation of an information system altogether, we have executed a pre-processing and an archiving process (see Fig. 1) to generate the input data for the ISoD process (see Fig. 2).

Pre-processing and archiving. In certain projects, like the NETamil project (NETamil Group 2014), research data on the manuscripts was initially stored in a Microsoft Word document (as DOCX format). The research data needed to be transformed into a CSV file to enable integration with the DBoD process. The pre-process proceeds as presented in Fig. 1 and described in the following: A DOCX document is essentially a zip archive that includes a file named document.xml. To extract and parse the content within the document.xml file, we defined a project-specific grammar and implemented a parser in JavaScript using ANTLR4 (Parr 2013), a tool that automatically generates source code for a parser based on a given grammar. This source code provides an application programming interface (API) for accessing specific parts of the text based on the defined grammar. The end result is a TXT document containing the contents of the original DOCX document, devoid of any XML tags. Subsequently, we employ a parser we have developed using ANTLR4 to parse the TXT document. This parser is generated as a Java code snippet by ANTLR4, also derived from a specified grammar. The parsed documents are transformed into CSV.

A configuration file is essential to streamline the automation of the ISoD process. This file should contain the data required to set up an information system. To fulfil this need, we employ a manifest file known as the METS file (see Section "Preliminaries").

In this article, we describe how a user does not need to actively create a METS file. The METS file is created in the background. For this purpose, we have implemented an Archiver Web Application, which is used to create the research data with the configuration data as a ZIP archive, which the researcher can then upload into an RDR.

Information system on demand. The main process of the ISoD process can be broken down into three key steps, as illustrated in Fig. 2.

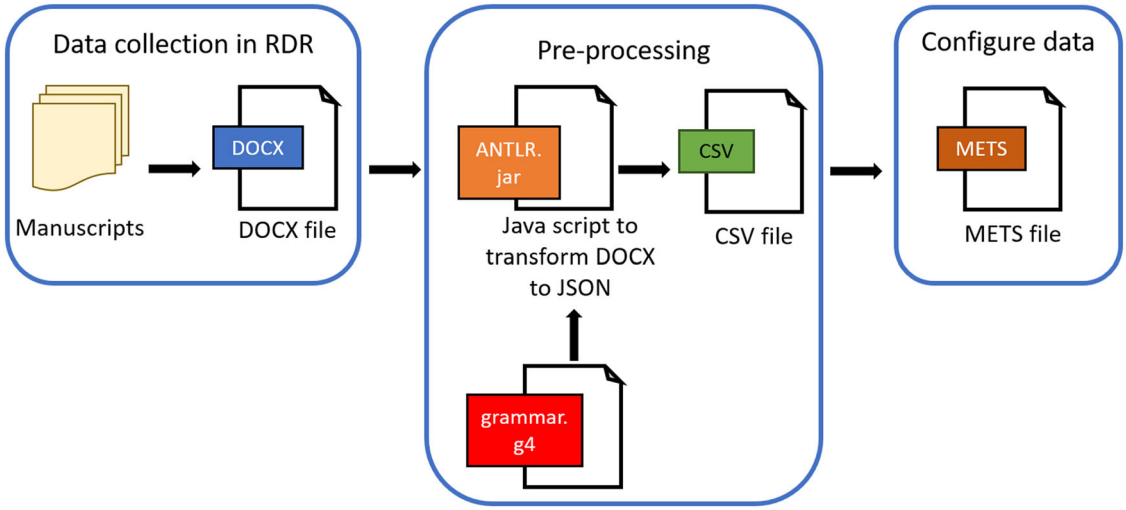


Fig. 1 Pre-processing and archiving process with three steps. i) data collection in RDR, ii) pre-processing, and iii) configure data.

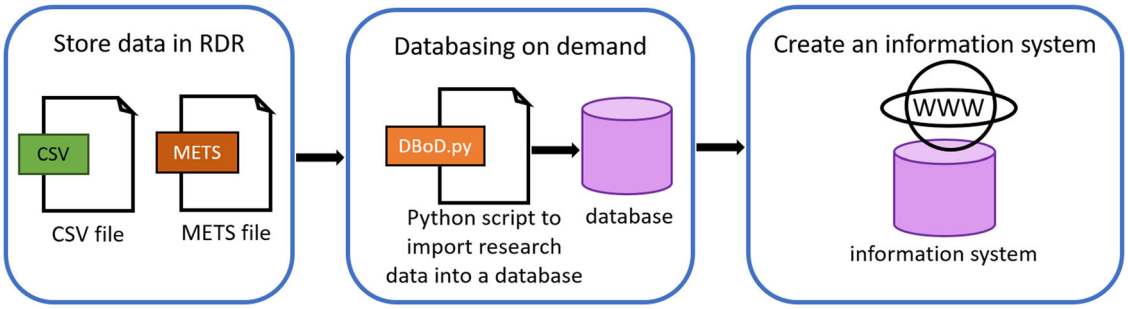


Fig. 2 The process for building information systems on demand. The process consists of three steps: i) store data in RDR, ii) databasing on demand, and iii) create an information system.

i. Store data in RDR: Users initiate the process by uploading a ZIP archive containing research data in CSV format, along with a METS file, into the RDR.
ii. Databasing on demand: The DBoD process is activated through the RDR, accessible via a new button or a process link. This step involves the execution of a Python script that imports research data from the CSV file into a database. Simultaneously, it generates and populates data tables in the background.
iii. Create an information system: An information system, such as a website, is constructed on top of the database. This is done with a specific focus on how the data should be presented. If Heurist is employed to create a web page, a template (typically a *.tpl file) can be integrated into the data set to customise the presentation.

In a prototype implementation, the ISoD process has been integrated into a local version of the RDR, making it easier for researchers to reuse archived data in the RDR with minimal effort, especially when dealing with standardised or commonly used formats. There are plans to migrate this process to the production version of RDR@UHH. This approach allows for the creation of numerous information systems as needed, effectively on demand.

Example information system: audition certificates platform. Audition certificates contain an unprecedented wealth of historical data that has received little scholarly attention. They are written records on handwritten books that confirm the authorised transfer of texts from teacher(s) to student(s). Specifically, the texts were recited aloud, either by a teacher or one of the students, and after such a reading session, a member of the group wrote the audition certificate onto the book. Through their involvement, all students gained the privilege to serve as teachers in subsequent reading sessions. These certificates may encompass various details, such as the names of the teacher(s), student(s), reader, writer of the certificate, book owner, reading date, reading location, and more.

The project "Reading the Scholarly Archive in the Pre-Ottoman Arabic Middle East" (RFE14)[5] which worked on the ACP (Audition Certificates Platform) wants to identify, among others, the role of women and slaves as historical actors. Furthermore, topographical information, such as names of buildings, kinship relations, prices, historical events, and designations for various trades and professions, are to be investigated. An information system is suitable for searching for this information.

During the ACP projects, information systems were implemented in two different ways for different target groups. The one way was: We used the database management tool Heurist[6] to analyse the collected research data and make it accessible to users through a website (see Fig. 3). Once the project is completed, the information system can be exported in CSV, XML, or JSON formats and archived in an RDR. The other way was to create a web page manually, directly from the original files, using JavaScript[7].

So, there are two different viewers for presenting research results. The METS file was used to specify which viewer should be used to display the research results. For displaying the data in a Heurist database instance, we generated the METS file as

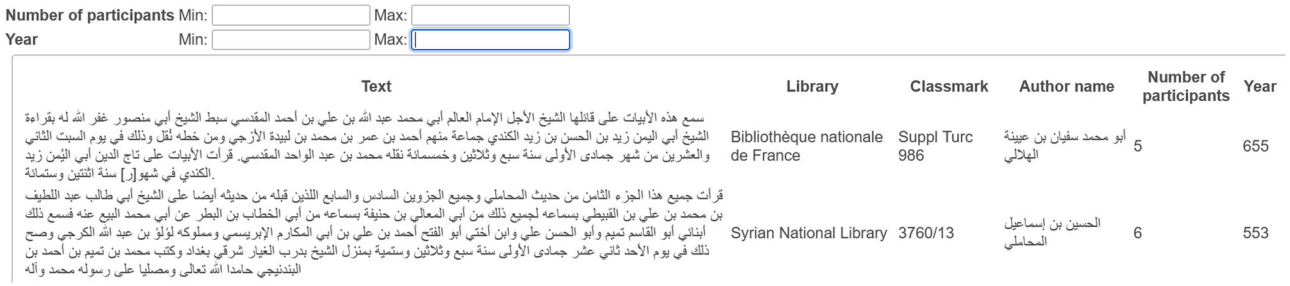


Fig. 3 Automatically generated table view with the columns "Text", "Library", "Classmark", "Author name", "Number of participants", and "Year" in Heurist. The entries can be min/max filtered by the "Number of participants" and "Year" (see above).

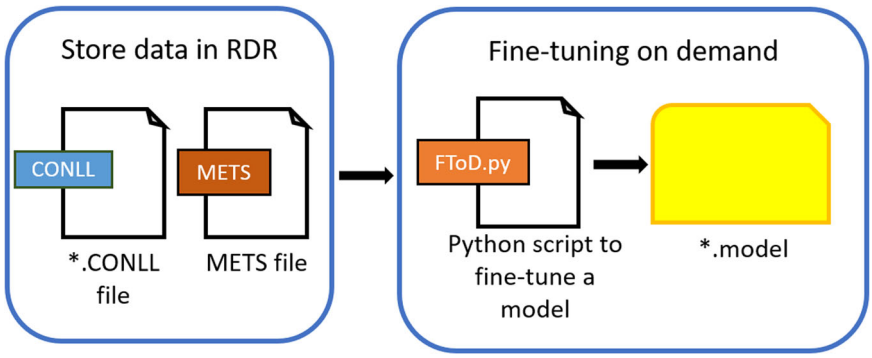


Fig. 4 Fine-tuning on demand process. The process consists of two steps: i) store data in RDR, ii) Fine-tuning on demand.

presented in Listings 5. The output of the ISoD process is presented in Fig. 3.

Fine-tuning on demand

Data that is produced and uploaded to an RDR can also serve the purpose of another process, which we call FToD. This process allows users to fine-tune a transformer model directly within the RDR, eliminating the requirement for proficiency in programming languages and necessary libraries (such as Python with HuggingFace Transformers). The initial prototype is tailored for NER (Named Entity Recognition) tasks using BERT models.

The resulting model can be used for multiple further tasks, like manually creating graphical user interfaces for specific purposes or general usage with services like ChatHA (Section "ChatHA"). Figure 4 describes the main process for fine-tuning models on demand in two steps.

i. Store data in RDR: Similarly to the ISoD process, data needs to be stored in an RDR. We have decided to use the CoNLL-2003 format[8] for the prototype. While the original format also included part-of-speech (POS) tags and syntactic chunk tags, we have decided for now to only allow CoNLL-2003 files where only the named entity tag is present besides the word itself. In our first implementation, we expect the user uploading the data set to provide two files. One is for the training data itself, and the other is for the test data. We plan to change that in the future to change the training/testing split dynamically. Another future improvement is to allow different file formats, like the JSONL format[9] that can be exported directly from labelling tools like Doccano (Nakayama et al. 2018).
In addition to the data set itself, a METS file introduced prior must also be present. It specifies that the FToD process needs to be used as well as the used model (from HuggingFace Hub) and used parameters like learning rate and number of epochs, etc. Parameters not specified in METS file will be set to the default values as specified in the HuggingFace library[10]. While it is still necessary to create the METS file manually, the creation will later be integrated into the Archiver Web Application.
ii. Fine-tuning on demand: The second main step is the fine-tuning itself. Figure 5 shows how a screen in the RDR may look like once everything has been uploaded correctly and the user has permission to perform the fine-tuning. Once the user has started the process by clicking on "Train my model" (see the button in Fig. 5 below), it may take some time to finish the process. Depending on the number of users and the available hardware resources, some scheduling algorithms known from the domain of operating systems (Horn 1974) may be implemented to share the hardware more fairly. Thus, the system should e-mail the registered address, telling the user that the model has been successfully fine-tuned. Afterwards, the user can see the results of the fine-tuning by being redirected to the screen depicted in Fig. 6. This shows the results graphically to the user and allows the users to compare multiple models to pick the best one based on the scholar's needs and not necessarily based on standard metrics like precision, recall, or F1-score. As a next step, it is also planned to allow directly from within this screen to persistently store the selected models in the RDR for reusing them. Another scientist with a similar task can then use the model to annotate their own data, saving time and effort.
The code for the fine-tuning itself is written in Python 3.11 and based on the HuggingFace Transformers library. Fundamentally, it is not different from other code for fine-tuning BERT models like the one from Andrew Marmon (Andrew Marmon 2021), which was the basis for our code.

ChatHA

Using a similar approach as presented for FToD, we have developed a first prototype of ChatHA (Asselborn et al. 2023), a

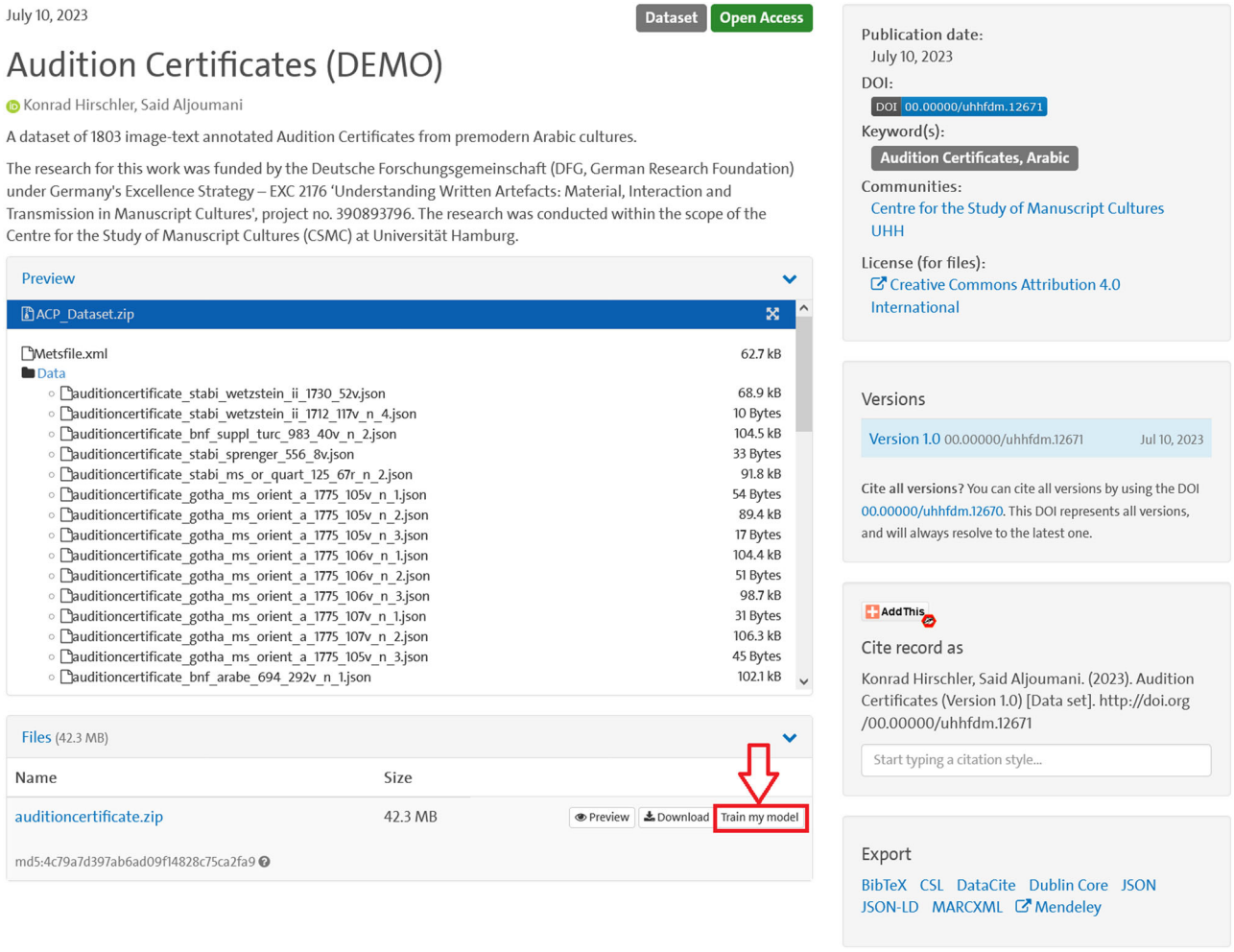


Fig. 5 Research data repository with demo data. The new "Train my model" button is shown with a red arrow and red border.

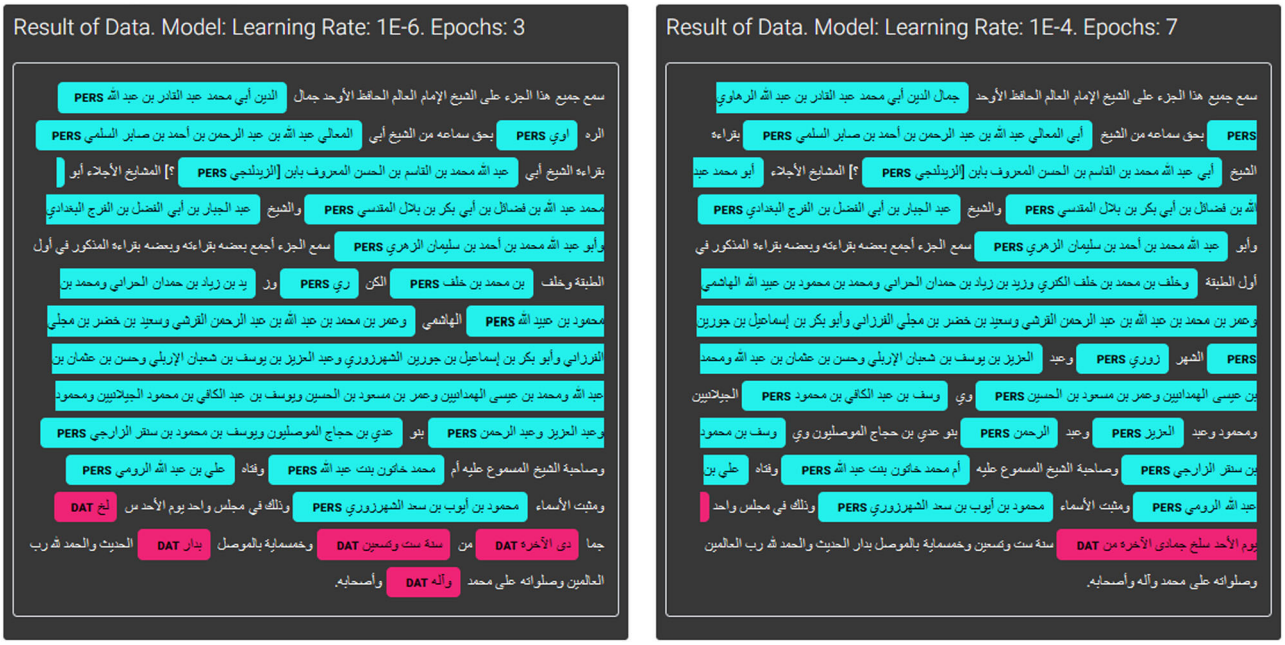


Fig. 6 After fine-tuning is completed, this screen is shown to verify a model. It can be used to compare multiple models to find the one that promises the best results. Text from Bibliothéque nationale de France, arabe 3481, fol. 229r. Link: https://gallica.bnf.fr/ark:/12148/btv1b110029593/f230.item.zoom.

chatbot designed for use in the field of humanities. The goal is to integrate the chatbot into an RDR to allow humanities scholars to interact with research data in the repository using natural language queries. This allows the RDR to be not only a storage of research data but also a tool that can improve the way scholars use and reuse data (possibly hidden in the number of entries).

One of the obstacles to using generative AI like ChatGPT in a scholarly field is the fact that these systems can return an answer that sounds convincing but is factually wrong. This phenomenon is often also called "hallucinations" (Child et al. 2019; Dao et al. 2022; OpenAI 2023). ChatHA is designed to avoid them by being grounded on the data present in the RDR. Thus, ChatHA can provide citations for generated text, potentially increasing users' trust in the system and making it verifiable. In contrast to using any source from some random page on the internet, sources from within the RDR are provided by other researchers from universities, which should also follow certain scientific practices. This approach can also lead to the system being more trustworthy than other systems. Additionally, ChatHA is also able to solve a second problem scholars can be faced with. There are special tools for certain tasks that humanities scholars face. For instance, the "Maktaba Shameela"[11] contains a multitude of texts from Arabic history, Gallica contains texts from the Bibliothèque nationale de France[12] and EFES (EpiDoc Front-End Services) (EpiDoc 2022) is for displaying EpiDoc (Epigraphic Documents in TEI XML) files (Bodard and Yordanova 2020) in a human-readable format. Every tool works differently and provides different functionalities, which makes it necessary to get used to them when needed. Working with natural language that aligns with a scholar's thought process eliminates the hurdle of learning a new GUI (Graphical User Interface) for each new project and/or task. To overcome these challenges, we have developed the following process presented in Fig. 7. The process is quite analogous to FToD with BERT. For FToD, we build a model for given data and need to build a viewer for the resulting model. For ChatHA, we also build a model for given data, but accompany the resulting model with a chatbot. We now describe the process for building the chatbot:

1. Data collection in RDR: This step is, in principle, identical to the first step in ISoD as well as FToD. Since ChatHA relies on the data in the RDR, it must first be stored. However, unlike the other processes, a METS file is unnecessary because no project-specific configuration must be stored at upload time. There is a need for a configuration file at the side of the user querying the system specifying, e.g. the specific entries one wants to have in the context of ChatHA, which may be in the form of METS as well with details being worked on at a later stage. Our prototype implementation works with data in the form of PDF files, and other file types are already planned to be compatible at a later stage.
2. ChatHA: The second step describes the process itself. Here, the text from the PDFs is extracted and used on the one hand for generating SCDs through the USEM algorithm (Bender et al. 2023). This process is done offline before the system is used and does not require recomputation at every query but only when the user decides to use a different set of PDF files to be used. On the other hand, the text is also used to augment the underlying generative model. We have used Llama as our basis because it is Open Source, provides a clear license allowing use in both commercial and non-commercial projects and provides good performance overall (Touvron et al. 2023). Other generative AIs can also be used if necessary, e.g. in the future when a new model is released, improving speed, memory usage, performance, etc. The augmentation is then done using the RAG method. When a user enters a query, this approach takes a set of supporting documents from the selected corpus, i.e. the text from the selected PDF files, and concatenates them with the original user query to form a new query given to the base generative AI, in our case Llama (Lewis et al. 2020). As a final step, the output from the augmented Llama and the SCDs are combined to provide the citations. This is done by also generating SCDs from the output. The most probable SCDs can be found using the MPS^{2}CD algorithm. The output with citations will then be presented to the user.

The system in its current form will also return an output when it finds no SCDs. We have identified a dilemma between

1. giving answers that are not backed by SCDs, which may be correct or wrong but can give important clues for further research or lead to potential trust issues in the system and
2. only giving answers backed by SCDs which are verifiable by the source but potentially giving an answer similar to "Sorry, I can not help you with the query." which may be frustrating and may result in not using the system because it will be seen as unhelpful.

The better option will be discussed later; real-world user tests must be performed to find that.

Integration into the RDR: a concept. With ChatHA, interacting with the chatbot is as important as the technical background. Thus, a focus on ease of use is essential. In particular, ChatHA needs to be straightforwardly accessible from within the RDR and directly usable.

Therefore, we add a new button "Add to ChatHA" to each data set in the RDR analogue to the button "Train my model" shown in Fig. 5. Pressing this button, the data set is added to the set of files used to augment the underlying generative model. In the backend, SCDs are generated for the new set of files. Future queries to ChatHA are also answered with the added data set.

Moreover, we add a user interface to ChatHA to the view of each data set in the RDR. When ChatHA is asked through this interface, it is temporarily extended by the currently viewed data set. Thus, on the one hand, we help scholars by providing a way to chat with each data set with ease. On the other hand, we further help them to connect the data set with the others in the RDR already added to ChatHA.

Application and results

In this section, we present the application and the results of the processes ISoD, FToD, and ChatHA in four humanities projects. This section shows that all approaches work with already existing data, which can help to keep the data hot and sustainable.

Information system on demand. We have already applied the ISoD approach in several projects in the field of humanities. Here we present four, namely ACP (see Section "Information system on demand"), Beta masāheft[13], NETamil (see Section "Pre-processing and archiving"), and EDAK (Epigraphische Datenbank zum Antiken Kleinasien)[14], to show that, in detail, there are different requirements for the data formats in which research data is stored and the researcher's requirements for representing this data in an information system.

In project ACP, the research data was accessible in the JSON format; in the projects Beta masāheft and NETamil in the TEI format, and in project EDAK in the EpiDoc format. To automatically transfer this data to a Heurist database instance, a CSV file with the results and a METS file were first created for

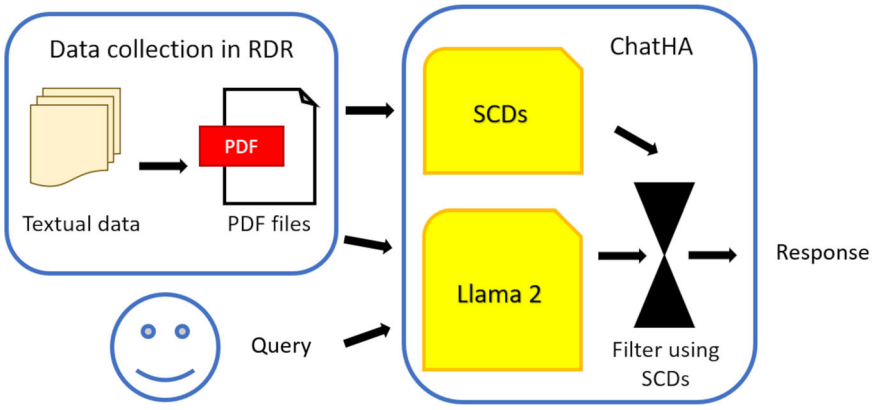


Fig. 7 Overview of ChatHA's workflow. The process consists of two main steps: i) data collection in RDR, ii) the ChatHA process. ChatHA takes a user query as input and generates a response.

each project. For projects where the project-specific, machine-readable conventions encoded in XML must be more readable for humans, the extensible stylesheet language transformations (XSLT) stylesheets (Elliott et al. 2008-2017) can be used for EpiDoc and TEI. As shown in Fig. 8, if the XSLT stylesheets are used, the ISoD processes take significantly longer per file compared to the case where the stylesheets are not used.

The long processing time is because every XML tag in every document is parsed, which takes time as the number of documents increases. As we can also see from the Fig. 8 using the Beta masāheft project as an example, three-quarters of the time can be saved without the use of XSLT compared to the use of XSLT. The numbering here is for illustrative purposes. "1" in Beta masāheft 1 indicates that XSLT is not used, and "2" in Beta masāheft 2 means that XSLT is used. The same applies to NETamil 1 and 2.

For the implementation in a productive system, this means that an upper limit should be set for the number of files or entries in the CSV files; otherwise, the execution of the ISoD process can lead to an overload and also take too long.

Fine-tuning on demand. The FToD process was applied to the ACP data in JSON format found in the RDR (Hirschler and Aljoumani 2023). 3519 audition certificates are present in the repository. They were pre-processed manually to have them in the needed CoNLL-2003 format. During that process, around 80% of the samples were put into the training data set while the remaining 20% were put into the testing data set. Because this project deals with Arabic texts, a model pre-trained on Arabic texts has been used. In our case, this is CAMeLBERT (Inoue et al. 2021). CAMeLBERT is a class of BERT models containing four individual models for different dialects of Arabic Table 1.

Table 2 shows the parameter combinations we have used during our experiments. All other parameters not mentioned in the table are kept to the default values as described previously.

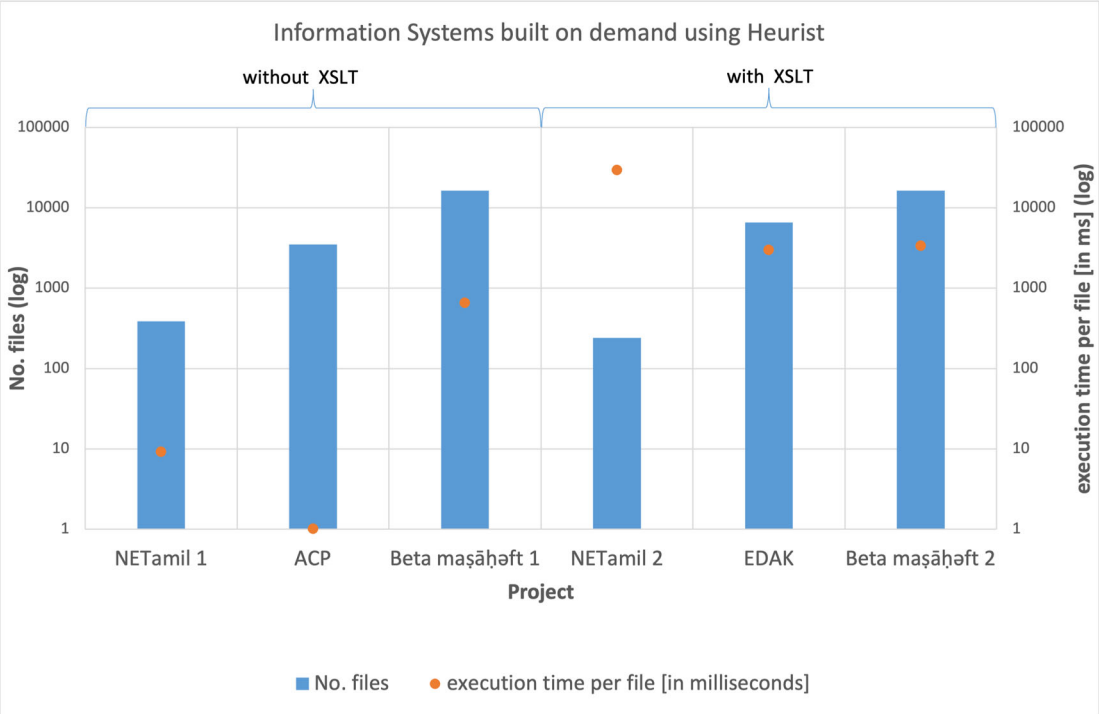
As a first step, we compared the four given CAMeLBERT models to find the best candidate for further experiments. To do this, we have picked the middle values for both weight decay, meaning 1.0000×10^{-5}, and for epochs, i.e. five epochs. The results are shown in Fig. 9, where we compared the metrics precision, recall, F1 and accuracy of the NER task evaluated on the testing dataset. CAMeLBERT-CA outperformed all other models for all metrics, while CAMeLBERT-DA was the worst. Thus, we proceeded with further experiments using the model for classical Arabic only.

Nine experiments were carried out on the model for classical Arabic. While the parameter for weight decay was kept at 1.0000×10^{-4}, the number of epochs was set to three, five and seven, respectively, and for each number of epochs, the learning rate was set to 1.0000×10^{-4}, 1.0000×10^{-5}, and 1.0000×10^{-6}. The results are depicted in Fig. 10. It can be seen that a learning rate of 1.0000×10^{-4} was providing the highest metrics across all epochs. When increasing the number of epochs, performance increased so far for all learning rates tested although most, when the learning rate was lowest.

Problems and challenges. The goal of the FToD process is for users to obtain a fine-tuned model without needing to learn programming. Ideally, it should work out of the box, even when users upload less-than-ideal data. For example, the CoNLL-2003 file should only have two columns: the word itself and the named entity tag. If the user decides to generate a file with three columns or makes a different mistake, e.g. by missing a tag on one word, the process should still function or give at least a reasonable error message. Finding all possibilities for things that can go wrong is a task we need to focus on in the future to make the process as smooth as possible. Additionally, the choice of hyperparameters has an impact on the performance of the final model (Feurer and Hutter 2019). Thus, a user picking the wrong hyperparameters may be underwhelmed by the outcome of the fine-tuning. To mitigate this, libraries like Optuna (Akiba et al. 2019) can be integrated into the process to automatically find hyperparameters that are good for the specific purpose. However, this now increases the execution time and resource usage for fine-tuning, which may be problematic when multiple users are waiting for their results simultaneously.

ChatHA. We make ChatHA available through interfaces directly in the RDR. Moreover, we provide a chat interface just like ChatGPT. The scholar can chat with ChatHA and ChatHA answers with the files added as additional context. ChatHA allows users to chat directly with a chatbot capable of searching the documents. Thus, it eliminates the need for a researcher to have expert knowledge in the specific field of interest.

Another advantage of ChatHA in contrast to a regular LLMs is the ability to take a subset of the RDR into account and provide citations. Therefore, we evaluate exactly this behaviour: We ask the same query to Llama and ChatHA and compare the output. We also post-process Llama with SCDs to compare the provided citations. The focus of our evaluation is twofold: First, we examine the ability to take the corpus into account, which leads to, second, a demonstration that the citations returned in ChatHA are correct and verifiable. To do so, we first run queries of the kind *Complete this sentence:* ... with the answers shortened to the relevant parts:



(a)

Project	File Type	XSLT Used	No. of Files	Execution Time per File [in ms]
NETamil	CSV	no	388	9.2784
ACP	JSON	no	3,516	1.0239
Beta maṣāḥəft 1	TEI	no	16,337	661.0761
NETamil 2	TEI	yes	242	29,752.0661
EDAK	EpiDoc	yes	6,570	3,013.6986
Beta maṣāḥəft 2	TEI	yes	16,337	3,415.5598

(b)

Fig. 8 Execution time comparison for different projects. a ISoD using Heurist. The diagram shows the duration of an ISoD process without (left) and with (right) the usage of XSLT stylesheets for six projects. **b** The table shows for six projects the execution time per file (in ms) of an ISoD process considering the file type, if XSLT is used or not, as well as the number of files.

Table 1 Summary of CAMeLBERT models.

Model	Description
CAMeLBERT-CA	This is the BERT model pre-trained for classical Arabic (CA) https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-ca
CAMeLBERT-DA	This is the BERT model pre-trained for dialectic Arabic (DA) https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-da
CAMeLBERT-MSA	This is the BERT model pre-trained for Modern Standard Arabic (MSA) https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-msa
CAMeLBERT-MIX	This is the BERT model pre-trained for a mixture of the other three variants https://huggingface.co/CAMeL-Lab/bert-base-arabic-camelbert-mix

1. Query: "Complete this sentence: The poem's layout features one stanza per line, although there is no particular 'stop' (kituo) sign between the vipande, hence it gives the general"
2. Llama's answer: "The poem's layout features one stanza per line, although there is no particular 'stop' (kituo) sign between the vipande, hence it gives the general impression of a continuous flow of thoughts and ideas."
3. ChatHA's answer: "impression of each stanza being one full line. [PDF Volume 17: p. 74 sent. 14] …"

Based on its training data, Llama makes up some information and completes the sentence with the most probable words. ChatHA continues the sentence correctly and provides a reference where the sentence is found. The citation is missing in Llama, too. Therefore, ChatHA can use the information in the RDR and provide references. On the downside, ChatHA takes approximately 8.3 seconds to answer the query on a single Nvidia DGX2 GPU, while Llama only takes approximately 2 seconds.

We can perform the test with another query as well:

- Query: "Complete this sentence: This assumption is supported by the fact that, despite the Swahili manuscript not bearing a title, this second part of"
- Llama's answer: "This assumption is supported by the fact that, despite the Swahili manuscript not bearing a title, this second part of the text is written in a distinctive style that is different from the rest of the manuscript. [PDF Volume 17: p. 74 sent. 24] …"

- ChatHA's answer: "manuscript 333 displays its own title on the top of the first page, where the number '2' occurs (supposedly meaning 'second part'). [PDF Volume 17: p. 74 sent. 24] …"

Once again, Llama continues the sentence based on its training data. But ChatHA correctly continues the sentence and provides a reference to it, although the reference is one page off. Llama takes approximately 6 seconds to answer, while ChatHA needs approximately 11.1 seconds.

For researchers, a query type like *Where is XYZ mentioned?* is more interesting, as this can lead to new connections. Therefore, we conduct further experiments with this query type.

- Query: "Where is 'Ajami Correspondence Letters' mentioned?"
- Llama's answer: "I apologise, but I cannot find any mention of 'Ajami Correspondence Letters' in the text of the article. [PDF Volume 18: p. 123 sent. 3] It is possible that the term is used in a different context or article. [PDF Volume 18: p. 21 sent. 1] Can you provide more information or context about where this term is used? [PDF Volume 18: p. 21 sent. 1]"
- ChatHA's answer: "Amulet compilation. The amulet compilation is a unique feature of Swahili oral practice that has been 'translated' into the manuscript page. The forms of the amulets are characterised by a mix of Arabic and African influences, reflecting the blending of cultures in the Swahili coastal region. The traces of usage can be seen in the variations in spelling, grammar, and syntax, which are reminiscent of the spoken language. The amulet compilation is found in several other texts of the codex, indicating its importance in Swahili culture. The compilation includes a variety of amulets, each with its own unique properties and uses. Some of the amulets are protective, while others are meant to bring good luck or to ward off evil. The term "Ajami Correspondence Letters" is mentioned in the codex, indicating that the amulet compilation is related to other texts that use the Ajami script. The Ajami script is a form of Arabic script used in the Swahili coastal region, and it is believed to have been introduced by Arab traders and scholars in the 10th century. [PDF Volume 19: p. 165 sent. 16] The use of the Ajami script in the codex suggests that the amulet compilation is part of a larger corpus of texts that were written in this script. [PDF Volume 19: p. 165 sent. 16] In conclusion, the amulet compilation in the codex is a unique feature that reflects the blending of cultures in the Swahili coastal region. The traces of usage on the forms suggest that the amulets were an important part of Swahili oral practice, and their inclusion in the codex indicates their significance in Swahili culture. The mention of 'Ajami Correspondence Letters' in the codex suggests that the amulet compilation is related to other texts that use the Ajami script, further highlighting its importance in the region."

Table 2 Parameters and their values.

Parameter	Value
model	CAMeLBERT-CA, CAMeLBERT-MSA, CAMeLBERT-DA, CAMeLBERT-Mix
learning rate	1.0000×10^{-4}, 1.0000×10^{-5}, 1.0000×10^{-6}
epochs	3, 5, 7
weight decay	1.0000×10^{-5}

Some combinations were used during our experiments.

Here, ChatHA outperforms Llama again. Llama even refuses to answer. However, the given citation in the answer by ChatHA is incorrect. Here, we only print the first windows of an SCD, so other windows are more likely to fit. Llama takes only 2.7 seconds, while ChatHA needs 16.7 seconds.

Nevertheless, Llama and ChatHA fail to answer correctly on other query types. But, in this subsection, we can see evidence for ChatHA taking the RDR into account. Therefore, adapting to other query types is likely only further optimisations away. A possible optimisation would be, e.g. increasing the number of

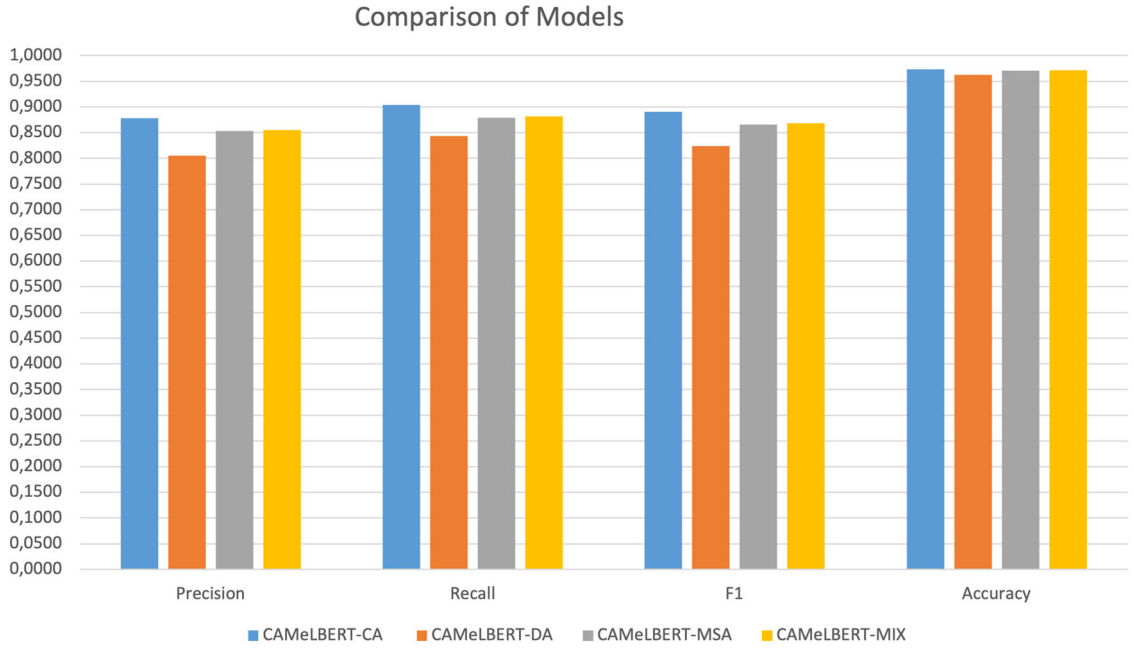
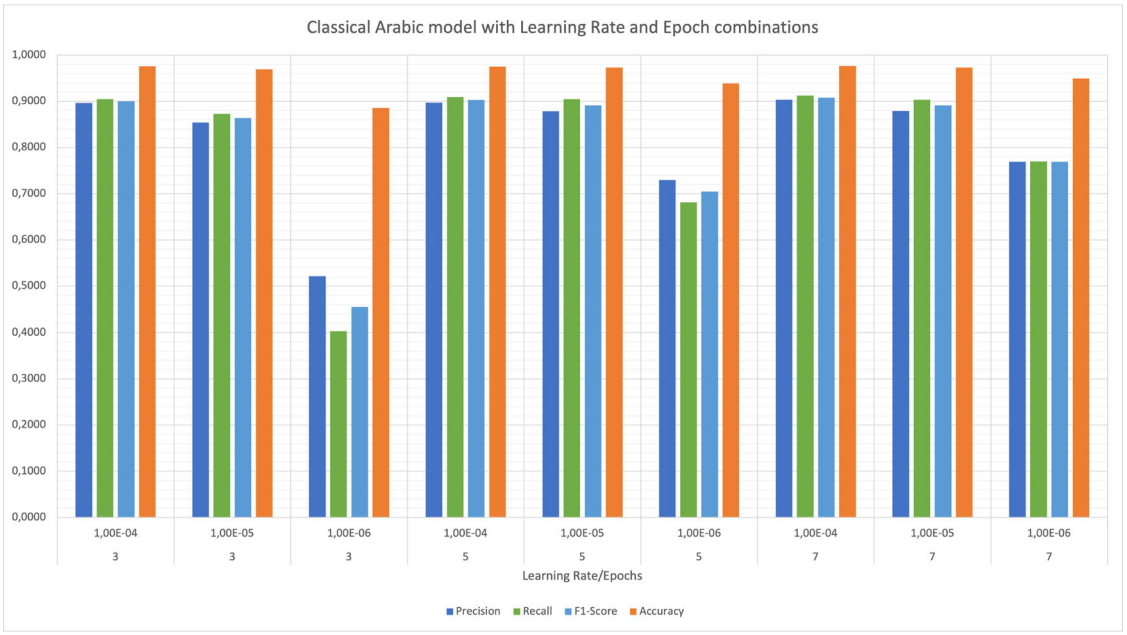


Fig. 9 Comparison of performance for four different CAMeLBERT models. The learning rate was fixed at 1.0000×10^{-5} and the number of epochs to 5. A user would pick the one with the best performance for further fine-tuning (in this case, CAMeLBERT-CA).



(a)

Epochs	Learning Rate	Precision	Recall	F1-Score	Accuracy
3	1.0000×10^{-4}	0.8961	0.9043	0.9002	0.9754
3	1.0000×10^{-5}	0.8542	0.8729	0.8634	0.9694
3	1.0000×10^{-6}	0.5215	0.4029	0.4546	0.8853
5	1.0000×10^{-4}	0.8967	0.9093	0.9029	0.9748
5	1.0000×10^{-5}	0.8781	0.9043	0.8910	0.9733
5	1.0000×10^{-6}	0.7293	0.6815	0.7046	0.9387
7	$\mathbf{1.0000 \times 10^{-4}}$	**0.9036**	**0.9121**	**0.9079**	**0.9763**
7	1.0000×10^{-5}	0.8788	0.9034	0.8909	0.9733
7	1.0000×10^{-6}	0.7691	0.7694	0.7693	0.9497

(b)

Fig. 10 Performance metrics for different configurations of Epoch and Learning Rate. a Comparison of performance for all experiments done using CAMeLBERT-CA for classical Arabic. The X-axis shows all combinations of chosen learning rates and epochs. **b** The row printed in boldface denotes the parameters with the best overall performance.

extracted sentences in the RAG approach up to giving whole related documents or articles into the context window of the LLM.

As a result, we have provided some insights into the sustainable development of on-demand systems by demonstrating the use of warm or hot archiving methods for research data within a RDR in the humanities. This approach emphasises the importance of maintaining the accessibility and relevance of data over time and simplifies the efficient analysis of numerous datasets for current and future projects. Other important findings highlighted in this article are the development of the sustainable usage of viewers (information systems) and transformer models. We have shown that tools such as ChatGPT can be contextualised to provide more accurate results and related resources in the humanities, ultimately reducing development time and improving data accessibility through the RDR integration.

Conclusion and outlook

In conclusion, developing and implementing the ISoD is essential for fulfilling the need for researchers to access and use data from an RDR. The integration of this functionality into the RDR ensured its long-term usability and sustainability. This article has presented how on-demand information systems can be strategically designed to ensure the reusability of data, providing essential tools and infrastructure for the reuse of archived data. Integrating the transformer model BERT opens new avenues for analysing historical data like pre-modern Arabic texts. While the challenge persists in enabling various users to fine-tune models, we argue for the necessity of performing fine-tuning directly from the RDR, emphasising the importance of archiving configuration parameters using METS, including suitable labels and viewer, with the research data to ensure optimal results. Chatbots like ChatGPT are becoming valuable tools for users who want to interact with data sets in natural language. Given the limitations in providing correct answers from time to time, improvements can be made by including project-specific content. The introduction of ChatHA extends this capability by providing answers with references and using annotations to provide personalised and subjective results. The developed prototypes validate that the ideas presented in the research can be effectively implemented in real productive systems. They demonstrate the feasibility and practical application of the proposed methods in enhancing data accessibility and relevance in research data management within the humanities field and beyond.

At the time of conducting the experiments presented in this article, the models for ChatHA and FToD were among the best and most recent ones. Since then, more recent and more advanced models, like Llama 3[15], have been released. These models may perform better in some terms of our evaluation.

However, our proposed solutions to problems, like not containing custom data or missing custom fine-tuned models, are still applicable.

In the future, we plan that the DBoD and FToD approaches become part of the function in all productive RDRs to help scholars' scientific work. Scholars in different fields would benefit from easily accessible data. Moreover, we also plan to improve ChatHA by integrating more techniques for using SCDs, like building an SCD information system or efficiently removing incorrect content from a repository using the approaches proposed in (Bender et al. 2024).

Data availability

Data for FToD: https://doi.org/10.25592/uhhfdm.13525. Selected data for ChatHA: https://www.csmc.uni-hamburg.de/publications/mc.html. Data from the EDAK project used for ISoD: https://doi.org/10.25592/5mx6-1k15. Data from NETamil project used for ISoD: https://doi.org/10.25592/mdq0-7x79.

Received: 15 November 2023; Accepted: 30 January 2025;
Published online: 15 February 2025

Notes

1 https://huggingface.co/autotrain
2 https://www.perplexity.ai/
3 https://huggingface.co/docs/transformers/index
4 https://www.loc.gov/standards/mets/METSOverview.v2.html
5 https://www.csmc.uni-hamburg.de/written-artefacts/research-fields/field-e/rfe14.html
6 https://heurist.fdm.uni-hamburg.de/index_en.html
7 https://www.audition-certificates-platform.org/
8 Details about the format can be found here: https://www.clips.uantwerpen.be/conll2003/ner/
9 https://jsonlines.org/
10 https://huggingface.co/docs/transformers/main_classes/trainer#transformers.TrainingArguments
11 https://shamela.ws/
12 https://gallica.bnf.fr/accueil/
13 https://www.betamasaheft.uni-hamburg.de/
14 https://www.epigraphik.uni-hamburg.de/content/index.xml
15 https://ai.meta.com/blog/meta-llama-3/

References

Akiba T, Sano S, Yanase T, et al. (2019) Optuna: A next-generation hyperparameter optimization framework. arXiv https://arxiv.org/abs/1907.10902
Andrew Marmon (2021) united-nations-ner. https://github.com/acoadmarmon/united-nations-ner, GitHub repository, Accessed 01 November 2023
Asselborn T, Melzer S, Aljoumani S, et al. (2023) Fine-tuning bert models on demand for information systems explained using training data from pre-modern arabic. In: Proceedings of the Workshop on Humanities-Centred Artificial Intelligence (CHAI 2023). CEUR Workshop Proceedings
Bender M, Braun T, Möller R, et al. (2023) Unsupervised estimation of subjective content descriptions. In: Proceedings of the 17th IEEE International Conference on Semantic Computing (ICSC-23) https://doi.org/10.1109/ICSC56153.2023.00052
Bender M, Braun T, Möller R, Gehrke M (2024) Enhancement of Subjective Content Descriptions by using Human Feedback. arXiv https://arxiv.org/abs/2405.15786
Bodard G, Yordanova P (2020) Publication, testing and visualization with efes: A tool for all stages of the epidoc editing process. Studia Universitatis Babes Bolyai Digitalia 65:17–35
Child R, Gray S, Radford A, et al. (2019) Generating long sequences with sparse transformers. arXiv https://arxiv.org/abs/1904.10509
Dao T, Fu DY, Ermon S, et al. (2022) Flashattention: Fast and memory-efficient exact attention with io-awareness. arXiv https://arxiv.org/abs/2205.14135
Devlin J, Chang M, Lee K, et al. (2018) BERT: pre-training of deep bidirectional transformers for language understanding. CoRR. http://arxiv.org/abs/1810.04805
Elliott T, Au Z, Bodard G, et al. (2008-2017) EpiDoc Reference Stylesheets (version 9). Available: https://sourceforge.net/p/epidoc/wiki/Stylesheets/, accessed November 06, 2023
EpiDoc (2022) EFES: EpiDoc Front-End Services. https://github.com/EpiDoc/EFES, GitHub repository, Accessed 10 November 2023
EUR-Lex (2019) DIRECTIVE (EU) 2019/1024 of the European Parliament and of the Council of 20 June 2019 on open data and the re-use of public sector information. Official J Eur Union L 172/56. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019L1024
Feurer M, Hutter F (2019) Hyperparameter optimization. Automated machine learning: Methods, systems, challenges pp 3–33
H2O.ai (2023) h2oGPT. https://github.com/h2oai/h2ogpt, GitHub repository, Accessed 08 November 2023
HEURIST (2022) A unique solution to the data management needs of Humanities researchers. https://heuristnetwork.org/, accessed 25 July 2023
Hirschler K, Aljoumani S (2023) Audition certificates platform. https://doi.org/10.25592/uhhfdm.13525
Horn W (1974) Some simple scheduling algorithms. Naval Res Logistics Q 21:177–185
Inoue G, Alhafni B, Baimukan N, et al. (2021) The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop. Association for Computational Linguistics, Kyiv, Ukraine (Online)
Kuhr F (2022) Context is the Key: Context-aware Corpus Annotation using Subjective Content Descriptions. PhD thesis, University of Lübeck, PhD thesis
Kuhr F, Braun T, Bender M, et al. (2019) To Extend or not to Extend? Context-specific Corpus Enrichment. In: Proceedings of AI: Advances in Artificial Intelligence pp 357–368. https://doi.org/10.1007/978-3-030-35288-2_29
Laudon KC, Laudon JP (2013) Management Information Systems: Managing the Digital Firm, 13th edn. Pearson
Lavoie BF (2012) Sustainable research data, Facet, p 67–82. https://doi.org/10.29085/9781856048910.005
Lewis P, Perez E, Piktus A et al. (2020) Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv Neural Inf Proces Syst 33:9459–9474
Melzer S, Schiff S, Weise F, et al. (2022) Databasing on demand for research data repositories explained with a large epidoc data set. CENTERIS
Melzer S, Thiemann S, Schiff S, et al. (2023) Implementation of a federated information system by means of reuse of research data archived in research data repositories. Data Sci J, https://doi.org/10.5334/dsj-2023-039
Nakayama H, Kubo T, Kamura J, et al. (2018) doccano: Text annotation tool for human. https://github.com/doccano/doccano, software available from https://github.com/doccano/doccano
NETamil Group (2014) NETamil. https://www.efeo.fr/base.php?code=811, accessed 25 May 2024
OpenAI (2023) GPT-4 Technical Report. arXiv http://arxiv.org/abs/2303.08774
Parr T (2013) The Definitive ANTLR 4 Reference. Pragmatic Bookshelf, Dallas/Raleigh, USA
Pernet C, Svarer C, Blair R, et al. (2023) On the long-term archiving of research data. Neuroinformatics 21(2):243–246. https://doi.org/10.1007/s12021-023-09621-x
Peukert H, Melzer S, Jacob J, et al. (2023) Das forschungsdatenzentrum der universität hamburg: Auf dem weg zu einem gemeinsamen verständnis im umgang mit forschungsdaten in den natur-, sozial- und geisteswissenschaften. Bausteine Forschungsdatenmanagement (3). https://doi.org/10.17192/bfdm.2023.3.8562
Pryor G (ed) (2012) Managing Research Data. Facet, https://doi.org/10.29085/9781856048910
Radford A, Narasimhan K, Salimans T, et al. (2018) Improving language understanding by generative pre-training. https://openai.com/research/language-unsupervised
Research EOFN, OpenAIRE (2013), Zenodo. https://doi.org/10.25495/7GXK-RD71
Schiff S, Melzer S, Wilden E, et al. (2022) TEI-Based Interactive Critical Editions. In Uchida S, Barney E, Eglin V (Eds.), 15TH IAPR INTERNATIONAL WORKSHOP ON DOCUMENT ANALYSIS SYSTEMS (pp. 230–244). https://doi.org/10.1007/978-3-031-06555-2_16
Schiff S, Möller R (2023) Persistent Data, Sustainable Information. Data Linking Workshop 2023: Computer Vision and Natural Language Processing - Challenges in the Humanities. https://doi.org/10.25592/uhhfdm.13099
Song K, Tan X, Qin T, et al. (2019) Mass: Masked sequence to sequence pre-training for language generation. In: Chaudhuri K, Salakhutdinov R (eds) ICML, Proceedings of Machine Learning Research, vol 97. PMLR, pp 5926–5936, http://dblp.uni-trier.de/db/conf/icml/icml2019.html#SongTQLL19
Thiemann K (2019) Broschüre Nachhaltiges Forschungsdatenmanagement. https://doi.org/10.25592/uhhfdm.699
Touvron H, Martin L, Stone K, et al. (2023) Llama 2: Open foundation and fine-tuned chat models. arXive https://arxiv.org/abs/2307.09288
Universität Hamburg (2022) Research Data Repository. Available: https://www.fdr.uni-hamburg.de/, accessed October 13, 2023
Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, et al. (eds) Advances in Neural Information

Processing Systems, vol 30. Curran Associates, Inc., https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
Vizcaíno JA, Reisinger F, Côté R, et al. (2011) PRIDE and "Database on Demand" as Valuable Tools for Computational Proteomics. vol 696. Humana Press, pp 93–105, https://doi.org/10.1007/978-1-60761-987-1_6
Wolf T, Debut L, Sanh V, et al. (2020) Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Online, pp 38–45, https://doi.org/10.18653/v1/2020.emnlp-demos.6

Acknowledgements

The research for this article was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2176 'Understanding Written Artefacts: Material, Interaction and Transmission in Manuscript Cultures', project no. 390893796. The research was conducted within the scope of the Centre for the Study of Manuscript Cultures (CSMC) at University of Hamburg. We acknowledge financial support from the Open Access Publication Fund of University of Hamburg.

Author contributions

Thomas Asselborn, Sylvia Melzer, Simon Schiff, Magnus Bender and Florian Marwitz contributed equality to this work. Said Aljoumani has acquired and prepared the data for the FToD process (Section "Fine-tuning on demand"). Stefan Thiemann, Konrad Hirschler and Ralf Möller have fundamentally supervised this article.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Ethical approval

Ethical approval was not required as the study did not involve human participants.

Informed consent

This article does not contain any studies with human participants performed by the authors.

Additional information

Correspondence and requests for materials should be addressed to Thomas Asselborn or Sylvia Melzer.