JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by

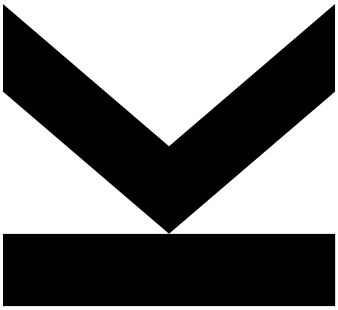
Jan Paffenholz, k12232332
Ardit Saliji, k11716015
Clemens Scheuchenpflug, k118031777
Felix Stadler, k12043299
Simon Ulmer, k12043331
Sebastian Wallner, k12005861
Anil Yildiz, k51912477

Submitted at

Institute of
Business Informatics -
Information
Engineering

META ANALYSIS

A supportive tool for universities



Seminar Paper

as part of

Information Engineering 256.404, Summer term 2024

in the Master's Program

Business Informatics

JOHANNES KEPLER
UNIVERSITY LINZ
Altenberger Str. 69
4040 Linz, Austria
www.jku.at
DVR 0093696

JOHANNES KEPLER UNIVERSITY LINZ

TABLE OF CONTENTS

1. Introduction 5
2. Methodology 6
3. State of the Field 6
   - 3.1. Metadata 7
     - 3.1.1. Types of metadata 7
     - 3.1.2. Derivation to research question 9
   - 3.2. Metadata extraction 9
     - 3.2.1. Rule-Based methods 9
     - 3.2.2. Machine Learning methods 9
     - 3.2.3. Deep Learning Methods 10
     - 3.2.4. Derivation to research question 10
   - 3.3. Information gaining from metadata 10
     - 3.3.1. Qualitative methods 10
     - 3.3.2. Quantitative methods 11
     - 3.3.3. Derivation to research question 11
   - 3.4. Dashboarding 12
   - 3.5. Conclusion of the Chapter 12
4. System implementation 12
   - 4.1. Problem identification and motivation 12
   - 4.2. Objectives of a solution 13
   - 4.3. Design and Development 13
     - 4.3.1. Technologies 13
     - 4.3.2. Backend 14
       - General information 14
       - Metadata 15
       - Metadata extraction 16
       - Information gaining from metadata 19
     - 4.3.3. Frontend 19
     - 4.3.4. Dashboarding 19
     - 4.3.5. Database 19
     - 4.3.6. Data model 20
     - 4.3.7. Process model 22
   - 4.4. Demonstration 23
   - 4.5. Evaluation 23

5. Discussion 25
6. Conclusion 26
7. References 27
8. List of Tables 31
9. List of Figures 32
10. Appendices 33

Abstract (EN)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim.

Abstrakt (DE)

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim.

1. Introduction

Universities try to combine the academic learning with practical application and therefore involve students in practical projects. These projects often lead to the creation of seminar papers in different contexts.

The sum of scholarly research papers may provide valuable information for future research or applications such as institutional repositories (Fan et al., 2019; Liu et al., 2018). However, due to the unstructured nature of research documents, it can be difficult to directly query their information (Fan et al., 2019; Herrera-Urtiaga et al., 2023). To gather information efficiently from past and future papers, extracting metadata is a compelling method (Cheung & Vijayakumar, 2016; Liu et al., 2018).

Metadata refers to data which provides information about other data, in this case, seminar papers (Herrera-Urtiaga et al., 2023). It helps users understand and manage datasets by providing descriptive details. The definition of metadata for documents aids in searching and filtering (Herrera-Urtiaga et al., 2023). Furthermore, it also enables more efficient and comprehensive meta-analyses of large collections of academic research (Adefowoke Ojokoh et al., 2009).

The appropriate metadata types play a key role in the gathering of useful information (Han et al., 2003). Various metadata standards have been proposed and implemented in different literature search engines (Han et al., 2003). Due to the consistent scope of available documents, a more precise selection of metadata is feasible for this research project.

In recent years, multiple methods for the extraction of metadata from research documents have been proposed (Hasan Choudhury et al., 2021; Tkaczyk et al., 2015). Available tools support in different fields of application, such as analyzing scanned documents, image metadata extraction, or bibliography metadata extraction (Hasan Choudhury et al., 2021). These tools can be powered by techniques such as regular expressions, rule-based parsers or machine learning (Adefowoke Ojokoh et al., 2009; Han et al., 2003; Liu et al., 2018).

The goal of this paper is to develop a research concept to extract a maximum of useful information out of the metadata derived from academic papers. The gathered information should be visualized to enable a better understanding and pictorial analysis of the results.

This will answer the research question: How can metadata be extracted automatically from academic papers to derive useful information for the use in a university environment?

To allow for a more structured approach, the research question is split into multiple sub-questions:

- Which metadata should be extracted for the analysis of information?
- How can metadata be automatically extracted from research papers?
- How can information be derived from the extracted metadata?
- How can the gathered information be visualized?
- ~~How can future work be integrated in this process? (based on the feedback from MS2)~~

To address these research questions, a comparative literature review was conducted in the context of meta-analyses, automatic metadata extraction, information derivation from metadata, and information display. In addition, a system that analyzes research documents and displays information derived from their metadata was developed.

This seminar paper is structured as follows. Chapter two depicts the methodology used for this literature review. Chapter three discusses the current state of the field in the context of metadata extraction, information gained from metadata and dashboarding of analyzed documents. Chapter four describes the selected metadata and the chosen data extraction process. In chapter five, an overview of the developed systems for information extraction and visualization is given as well as the possibility to integrate future work. Finally, chapter six summarizes the results, discusses the limitations and gives recommendations for future work.

2. Methodology

The methodology used in this seminar paper is the Design science research method (DSR) based on Peffers et al. (2006). It is a methodology which consists out of six process steps. At the beginning there is the problem identification and motivation. So, a problem needs to be defined and it has to be highlighted why this problem is of importance. Following this step the objectives of a solution must be determined. This includes the creation of an artefact and highlighting which improvements an artefact would bring to the current problem and status quo. After this process step the design and development phase of the artefact is taking place. Subsequently, a demonstration of the artefact takes place in a suitable context to show the desired functionality and how the artefact solve the problem of its purpose. Following the demonstration an evaluation takes place where the effectiveness and efficiency of the provided solution is observed. If this is not yet sufficient an iteration back to the design phase is possible. At the end of the DSR-method is the communication where the artefact and this documentation get published in form of a paper (Peffers et al., 2006).

<@David will send visual @Sebastian>

Pfeffers et al. (2018) compared five different genres of the Design science research. Out of those five the design science research methodology is the most suited one to adapt to this seminar paper and that's why we followed its characteristics. The focus of this genre is to develop an applicable artefact for the defined problem of the work. The whole process is flexible and not very concerned with strict design or process guidelines. Also, the role of theory is quite generalizable and its main focus point is to deliver supporting arguments for the potential success of an artefact. In addition to that the evaluation of this DSR genre is outcome orientated and practical (Peffers et al., 2018).
All these main aspects of the DSR methodology genre reoccur in this seminar paper and how the development of the artefact is approached.

3. State of the Field

Based on the given problem mentioned in the Introduction (see chapter 1.), the aim of this chapter is to describe the current state of the field. Consequently, this would provide an overview of the topic and its related fields as well as highlight the importance of the problem in the current literature. Therefore, this chapter begins with the topic metadata by describing it in general and continues with the related topics metadata extraction and how to gain information from metadata. In conclusion, the chapter ends with the topic dashboarding, where the aim is to present information extracted from metadata.

3.1. Metadata

This section provides an overview of metadata, characterizing it as essential data about data across various fields of applications. It discusses the classifications of metadata as outlined by the Dublin Core standard (Weibel & Koch, 2000).

Metadata can be very diverse, because basically everything that is stored to describe other things can be described as metadata. The classic and roughly formulated definition is that metadata is "data about data" (Riley, 2017). Therefore metadata is a very important component for a broad spectrum of applications, which helps to stitch together content and services and make them more visible to users (Weibel & Koch, 2000). Based on literature research this paper will give an overview about the types of metadata and how they are categorized. Furthermore, an insight in the work from the Dublin Core (Weibel & Koch, 2000) is given as this was established as a standard in the past.

3.1.1. Types of metadata

Riley (2017) separated metadata into descriptive metadata, administrative metadata, structural metadata, and markup languages. Descriptive metadata refers to all data which is relevant for finding or understanding a resource (Riley, 2017). Administrative metadata deals with technical details such as file formats and encoding schemes, ensuring proper decoding, and rendering of files for users (Riley, 2017). Therefore, the administrative metadata can be separated into technical metadata (decoding and rendering), preservation metadata (long-term management of files) and rights metadata (intellectual property rights from the referred content) (Riley, 2017). Structural metadata provides information about the organization and relationships within resources, such as the sequence of pages or chapters within a document (Riley, 2017). Markup languages integrate metadata and flags for other structural or semantic features within content, allowing the identification and interpretation of elements like headings and paragraphs (Riley, 2017).

The Dublin Core Metadata Initiative (Weibel & Koch, 2000) has developed a widely used standard for metadata which defines 15 elements for resource description. These 15 defined elements are shown in the following table (Weibel & Koch, 2000):

DCMES Element	Element Refinement(s)	Element Encoding Scheme(s)
Date	Alternative	-
Creator	-	-
Subject	-	LCSH MeSH DDC LCC UDC
Description	Table of Contents Abstract	-
Publisher	-	-
Contributor	-	-
Date	Created Valid Available	DCMI Period W3C-DTF

	Isued Modified	
Type	-	DCMI Type Vocabulary
Format	Extent Medium	- IMT
Identifier	-	URI
Source	-	URI
Language	-	ISO 639-2 RFC 1766
Relation	Is Version Of Has Version Is Replaced By Replaces Is Required By Requires Is Part Of Has Part Is Referenced By References Is Format Of Has Format	URI
Coverage	Spatial Temporal	DCMI Point ISO 3166 DCMI Box TGN DCMI Period W3C-DTF
Rights	-	-

Table 1: Dublin Core Elements (Weibel & Koch, 2000)

With respect to the elements defined in the Dublin Core (Weibel & Koch, 2000), Riley (2017) presented the following graph to also have the elements categorized according to the above mentioned types of metadata and stated the primary uses for the elements.

Metadata Type	Example Properties	Primary Uses
Descriptive metadata	Title Author Subject Genre Publication date	Discovery Display Interoperability
Technical metadata	File type File size Creation date/time Compression scheme	Interoperability Digital object management Preservation
Preservation metadata	Checksum Preservation event	Interoperability Digital object management

		Preservation
Rights metadata	Copyright status License terms Rights holder	Interoperability Digital object management
Structural metadata	Sequence Place in hierarchy	Navigation
Markup languages	Paragraph Heading List Name Date	Navigation Interoperability

Table 2: Metadata Elements categorized by type (Riley, 2017)

Riley (2017) shows that there is a differentiation within the main keyword metadata with respect to which data is referred to. Additionally, there is the Dublin Core model (Weibel & Koch, 2000) which states the most important elements when talking about metadata.

3.1.2. Derivation to research question

With respect to this work and explicitly to answer the defined research question “Which metadata should be extracted for the analysis of information?”, it is necessary to identify the most relevant data for the stakeholders which will use the developed artefact. So therefore, there will be more focus on descriptive metadata as this might be more useful to gain information about projects done as the other types of metadata.

3.2. Metadata extraction

Metadata extraction plays a pivotal role in scientific literature management and analysis (Adefowoke Ojokoh et al., 2009). The process of manually extracting data from research papers for meta-analysis is extremely time-consuming, especially when dealing with a wide variety of papers (Shah-Mohammadi & Finkelstein, 2024). In order to address this problem and support and speed up the data collection automated metadata extraction has become imperative (Shah-Mohammadi & Finkelstein, 2024). In this chapter state-of-the-art techniques of metadata extraction are reviewed.

3.2.1. Rule-Based methods

As the name already states rule-based methods rely on pre-defined rules (determined by domain experts – with consideration to the format and traits of the documents) that are used to extract the metadata from a scientific paper by identifying patterns (Hong et al., 2021; Nasar et al., 2018). An example for a rule-based extraction system is PDFX that extracts metadata from papers in PDF format and transforms it into XML (Tkaczyk et al., 2015). While being interpretable for humans, these methods struggle with variations in paper formatting and require significant manual effort to create and maintain rules for diverse paper styles (Guo et al., 2023).

3.2.2. Machine Learning methods

Machine Learning methods relies on models with machine learning algorithms that are trained on a corpus of scientific papers in order to learn features of this documents (Guo et al., 2023; Nasar et al., 2018). Based on the learned features of the training papers, metadata can be extracted from other papers (Guo et al., 2023). Popular methods for metadata extraction include Support

Vector Machines (SVMs) for classifying text segments (Tkaczyk et al., 2015). An example for a system based on machine learning methods is CERMINE by Tkaczyk et al. (2015) which allows structured metadata extraction from scientific articles (Tkaczyk et al., 2015). Machine learning methods offer improved adaptability compared to rule-based approaches, but require significant (labelled) training data, which can be expensive to curate (Guo et al., 2023).

3.2.3. Deep Learning Methods

Recent advancements in deep learning have shown significant promise for metadata extraction from scientific papers (Guo et al., 2023). For this method deep neural networks are used (Guo et al., 2023), especially pre-trained large language models that perform zero-shot metadata extraction (Kartchner et al., 2023; Shah-Mohammadi & Finkelstein, 2024). Examples for large language models that can be used for metadata extraction are GPT, Bard or Llama (Guo et al., 2023). Deep learning methods achieve high accuracy and therefore outperform rule-based and machine learning methods, however the results should be used cautiously (Guo et al., 2023; Kartchner et al., 2023).

3.2.4. Derivation to research question

Various approaches in the literature emphasize the importance of conducting information gain through metadata extraction, but the diversity in papers complicates matters, as not every paper is structured the same way (e.g. different title page) (Guo et al., 2023). Therefore, leveraging pre-trained large language models for metadata extraction, presents a promising solution as the data can be extracted via prompt engineering (Shah-Mohammadi & Finkelstein, 2024). This approach accommodates the variability in paper structures and streamlines the extraction process, ultimately enhancing the efficiency of meta-analysis and advancing scientific knowledge discovery (Guo et al., 2023).

3.3. Information gaining from metadata

In the context of this work, two possible starting points for the generation of information based on the extracted metadata have emerged. On the one hand qualitative methods and on the other hand quantitative methods (Formentini et al., 2022). A combination of the two approaches would also be possible (Lindner, 2020; Palinkas, 2014). Consequently, the aim of this section is to describe the process of deriving information from metadata through both qualitative and quantitative methods. This should give an overview of how the different approaches can be used to gain information.

3.3.1. Qualitative methods

Qualitative methods are used, among other things, to gain an in-depth understanding of a topic, which includes, for example, the perspectives of individuals (Palinkas, 2014). According to Lindner (2020), this allows for the exploration of topics that have not been investigated before. Brüsemeister (2008) describes these methodologies as discovery methodologies. In addition, qualitative methods use few data sets which are subsequently interpreted by the researcher (Brüsemeister, 2008; Lindner, 2020). However, a disadvantage is that, according to Lindner (2020), this interpretation is marked by a high subjectivity. He argues that such subjectivity can lead to different outcomes by different persons.

One method is the qualitative content analysis (Brüsemeister, 2008; Kuckartz, 2018; Mayring, 2010, 2015; Mayring & Fenzl, 2019). Mayring (2015) defines three types of qualitative content analysis. The first type which he defines is the structured/deductive analysis. According to his

work, this type of analysis aims to filter out a certain structure of the text based on a predefined category system. He says that this predefined category system should be derived by the research question and proven by the theory. The next analysis type he defines is the explicative analysis which further describes a certain part of the text with additional content. This content should, according to Mayring, expand the knowledge about the text part. The last analysis type he defines is the summary/inductive content analysis. This type of analysis should, according to his work, reduce the material to the most important parts furthermore generalize it. In conclusion, the qualitative content analysis is a method which could be used to gain information from the extracted metadata. The researcher could choose one of the defined types of content analysis from Mayring (2015) based on the research aim.

Another method is the case study (Fidel, 1984; Harling, 2012; Heale & Twycross, 2018; Lindner, 2020; Schögel & Tomczak, 2009). According to Lindner (2020), a case study should expand the theoretical knowledge with practical findings. Harling (2012) defines a case study as follows: “A case study is a holistic inquiry that investigates a contemporary phenomenon within its natural setting.” He states that the defined phenomenon can be a wide variety of entities, ranging from programs to people. According to Heinrich et al. (2011), in the field of business informatics, the organization is a commonly investigated phenomenon. Case studies make it possible to build more knowledge about a specific phenomenon (Heale & Twycross, 2018). Fidel (1984) states the following about case studies: “When applied as a research method, case studies are usually carried out to generate findings of relevance beyond the individual cases.” Such a method could include one or multiple cases (Harling, 2012; Heale & Twycross, 2018; Schögel & Tomczak, 2009). In addition, to gather information about the case, several empirical survey techniques, such as interviews, observations, etc., are used (Heinrich et al., 2011). A case study can be an appropriate method in the field of metadata research, as it allows for the incorporation of a metadata extraction system in a company and the subsequent gathering of information from it.

3.3.2. Quantitative methods

Quantitative methods on the other hand are methods that provide numbers and metrics that make products assessable on the basis of these metrics and are therefore used as engineering design tools (Formentini et al., 2022). The data is collected in a standardized and structured way, which is why a larger number of data sets are used (Brüsemeister, 2008; Lindner, 2020). According to Lindner (2020), quantitative methods facilitate the generalization and comparability of the research. As case studies are collecting qualitative or quantitative data (Heale & Twycross, 2018), it could also be included in this subsection. However, since this method is described in the previous subsection, it will not be discussed in this subsection.

One method is the quantitative content analysis (Rössler, 2017; Schneijderberg et al., 2022). In contrast to the qualitative content analysis, this method measures the frequency of a particular material, such as the frequency of certain words in texts or gestures in videos (Rössler, 2017; Schneijderberg et al., 2022). In fact, this explanation is analogous to the definition of quantitative methods at the beginning of this subsection. Rössler (2017) mentions that the sole collection of the data is not enough to gain knowledge. Therefore, according to his work, the collected data are described descriptively in the next step. In summary, the quantitative content analysis could be a conceivable solution to analyze the frequency of the extracted metadata.

3.3.3. Derivation to research question

Since a meta-analysis includes, among other things, the statistical evaluation of several bundled research papers (Cheung & Vijayakumar, 2016), and the inclusion of human interpretations and

perspectives would exceed the scope of the seminar paper, the focus is largely on descriptive metadata, which is intended to provide information via quantitative approaches.

3.4. Dashboarding

This section provides an overview of dashboarding. Dashboarding is a very popular technique used to visualize data and make it look appealing. But besides that, there is more to that than just making chart look good. A solid visualization makes valuable inside about a large amount of data available and easily accessible. So, facilitating and monitoring conditions is the real benefit a well-designed dashboard is providing. On the one hand the creation of a dashboard generates valuable knowledge about your own data. On the other hand it also leads to being able to spread this knowledge within a whole organization, making it available for everyone who needs it (Wexler et al., 2017).

Especially in use-cases where a large amount of data is available and needs to be organized and displayed dashboards are typically the first choice to use. Benefits of dashboards are the improved decision making and performance. For example, to easily identify either positive or negative trends. Additionally, the provided knowledge displayed in a dashboard enables an informed and data-driven decision process. Furthermore, the use of dashboards increases productivity due to an easy understanding as well as less effort to update and maintain them (Rasmussen et al., 2012).

The data which is displayed within a dashboard can be categorized in two different types, either metrics or key performance indicators (KPI). A KPI is a subset of a metric. So, every KPI is a metric but not every metric is a KPI. A metric is defined as a regular unit of something which can be measured e.g. any kind of values, parameters or measures. A KPI are metrics which are particularly important for the organization. Because of that most of the metrics displayed on dashboards are KPI's (Rasmussen et al., 2012).

To address the research question, it is essential to use a dashboard to effectively display the defined KPIs. Dashboards categorize data into metrics and key performance indicators (KPIs). By simplifying complex information, dashboards make data easily understandable and manageable.

3.5. Conclusion of the Chapter

This chapter has dealt extensively with the state of the field of the research topic. It builds a foundation for the following chapters and for the processing of the research goals as well as the research question. At first, this chapter dealt with the topic metadata. The second topic was metadata extraction. The third topic was information gaining from metadata. The topic dashboarding concludes this chapter. Thus, this chapter builds a basis to conclude with the results of this research.

4. System implementation

4.1. Problem identification and motivation

See chapter "Introduction" of this work.

4.2. Objectives of a solution

For this paper the about to be mentioned objectives have been defined and will be presented and explained in the following.
Firstly, a system for the metadata extraction has to be developed. That includes designing and creating an information system for extracting metadata from academic papers as a proof of concept. This information system should consist of a backend for handling the data and a frontend for visualizing the processed data.
To be able to create such an information system the relevant metadata must be determined beforehand. By doing that the potential impact on future research or practical applications within a university environment must be considered.
The metadata extraction process shall be performed by a semi-automated method to extract the correct metadata from the research documents. Adding to that methods must developed to derive meaningful information from the extracted metadata. The information system should enable insights that can support decision-making processes or future/further research.
To present the gathered information in a clear and intuitive manner a visualization should be designed. That facilitates understanding and enables stakeholders to perform in-depth analysis of the results. Also, users / stakeholders are enabled to adapt the displayed metrics within the dashboard to be able to handle new requests as well as looking at the data from another point of view through drill downs or other / additional forms of visualization.
The information system should allow users to change extracted data for the case of wrong extraction. Furthermore, the option to enter additional metadata about the paper work must be available.
The operability and sufficient functionality of the solution has to be tested with seminar papers in the archive of the Institute of Business Informatics – Information Engineering. To better evaluate the effectiveness and usability of the information system iterative feedback rounds should be conducted together with all relevant stakeholders. Throughout this whole process the methodology used in the design and implementation process as well as the findings obtained through evaluations to get valuable insights have to be documented.

4.3. Design and Development

This chapter explains the practical solution, therefore the used technologies, the data and process model and the structure of the resulting information system.

4.3.1. Technologies

The information system consists of a frontend, a backend and a database that are connected through REST APIs, therefore the components communicate with Http. In the following, the technologies for the implementation are presented with their reason for use.

The frontend is implemented in Next.js, an open-source web application framework that is built on top of React.js. The framework is used due to prior knowledge and experience in the project group.

For the backend Python in combination with Django, a high-level Python web framework, is used. Python as the technology for the backend is used to simplify the connection with the chosen large language model and due to the experience in the project group. Django, being an open-source framework, is freely available, making it attractive and its simplicity in developing REST interfaces further adds to its appeal.

Llama 3, an open-source large language model developed by Meta, is employed for metadata extraction. It is integrated with Ollama to function as a chatbot. The decision to utilize Llama 3 is driven by its capability to run locally within a Docker container, coupled with the team's existing experience with the model.

For the management of the academic papers and the extracted metadata PocketBase, which includes a real time relational database as well as a file storage, is used. Furthermore, PocketBase is useful for the information system due to its REST API, which allows a seamless integration with the back- and frontend.

In order to ensure version control and seamless collaboration in the implementation process withing the project group Git in combination with GitHub is used.

4.3.2. Backend

In the following, the structure of the Django Python backend is presented with the different views (REST API) and also the algorithm for the metadata extraction through the large language model.

General information
The backend consists of two views that also represent the interface to the frontend (as you can see in the process model).

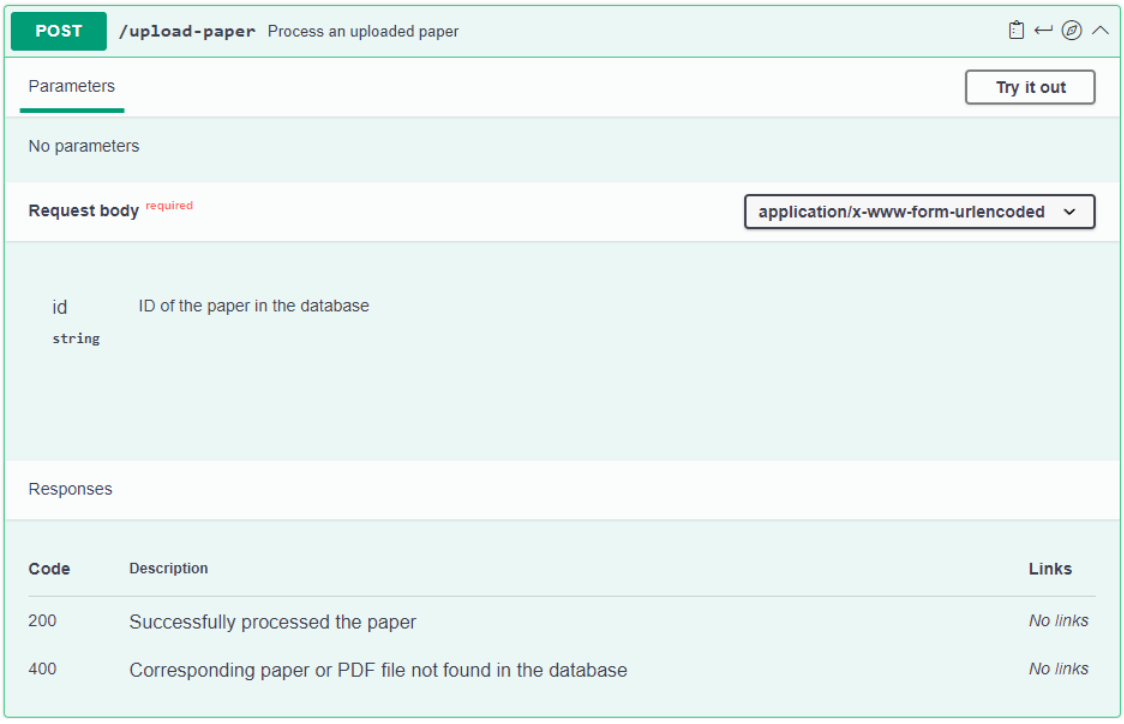


Figure 1: View "upload-paper"

The view “upload-paper” gets a post-request from the frontend with the id of the database object for the new paper (see Figure 1). With this id the database object can be loaded from the database and therefore also the id for the corresponding pdf file from the fileserver of the database. In order to access the loaded pdf file in the processing function the file needs to be stored. Then the processing of the file gets triggered in this view and afterwards the results are written back to the database.

Listing 1: View “upload-paper” (views.py)

Metadata

In order to store the metadata correctly all the classes of the data model were adopted in the backend as classes. See Listing 2 for the class “Literature” – the other classes have the same structure according to their corresponding attributes in the data model.

Listing 2: Data model classes (models.py)

```
# Class for the literature in a paper
class Literature():
    def __init__(self, title, id=None, authors="", date="", doi="", url=""):
        self.id = id
        self.title = title
        self.authors = authors
        self.date = date
        self.doi = doi
        self.url = url
```

Furthermore, to correctly interact with the http-requests the data needs to be serialized or deserialized in/from JSON format. Therefore, a function for data serialization and a deserialization function for each class of the data model is implemented (see Listing 3 – only the deserializer for the class literature is shown due to the similarity for the other classes).

Listing 3: Serializer and deserializer (serializers.py)

```
import json

# Function to serialize objects into JSON format
def serialize_object(obj):
    return json.dumps(vars(obj))

# Function to deserialize JSON format data of a paper
def deserialize_literature(jsonData):
    # Deserialize JSON into a Python dictionary
    data = json.loads(jsonData)

    return Literature(
        title=data["title"],
        id=data["id"],
        authors=data["authors"],
        date=data["date"],
        doi=data["doi"],
        url=data["url"]
    )
```

Metadata extraction

As already mentioned earlier, the metadata extraction process gets initially triggered in the "upload-paper" view with calling the function "processPaper" (see Listing 1).

Listing 4: Function "processPaper" (metadataExtraction.py)

In Listing 4 the function "processPaper" is presented. In the first step the pdf file, that was already loaded from the file storage and stored on the machine in the "uploadPaper" view, gets loaded with the PyPDFLoader class of the langchain_community package. This module splits the pdf into a list of Documents. The defined metadata should be within the first five pages of each paper and therefore to save time with the processing and also to prevent misinterpretations of the large language model with the sources only the first five pages are used. Furthermore, for the extraction of the sources the list of literature gets extracted (also see Listing 5) and also the whole pdf file without the list of literature gets extracted to calculate the word count.

Listing 5: Function "find_references" (metadataExtraction.py)

```
# Function to find the references in the pdf file
def find_references(pdfFile):
    for i, page in enumerate(pdfFile):
        text = page.page_content
        lowercaseText = text.lower()

        # Check if the first 40 characters include a number, a point,
        # "Literaturverzeichnis", "Literature", or "References",
        # and a newline character '\n' with optional spaces in between
        if any([char.isdigit() for char in lowercaseText[:40]]) and \
                ("literaturverzeichnis" in lowercaseText[:40] or "literature"
in lowercaseText[:40] or "references" in lowercaseText[:40]) and \
                "\n" in lowercaseText[:40]:
            return i

    # Return 5 if no such index is found - just use everything after the
      # metadata pages
    return 5
```

In the next steps the first five pages of the pdf file are converted into tokens with the RecursoveCharacterTextSplitter class of the langchain module. The tokens are then converted into embeddings with the OllamaEmbeddings class by the large context length text encoder (embeddings model that can only be used to generate embeddings) nomic-embeded-text and afterwards stored in a chroma database. This database is then used as context for the metadata extraction with the large language model. In the next step, the Retrieval-Augmented Generation (RAG) system is set up, which integrates a document retriever with a language model to enhance the generation of responses by providing contextually relevant information. At first, with the method as_retriever() the vectorstore (from the chroma database) is transformed into a retriever

capable of fetching relevant documents or information based on a query. This transformation is crucial for enabling the retrieval component of the RAG system. Then the large language model Llama3 is configured using the ChatOllama class (the other parameters are not important – yield similar results independent of the set parameters). In the next step a chain is created with the class RetrievalQA that combines the large language model and the retriever. This chain enables the retrieval of relevant documents, which the language model then utilizes to generate informed and contextually accurate responses. Last but not least, the invoke method is called on the chain with an input dictionary containing the query. This method orchestrates the retrieval of relevant documents through the retriever and subsequently employs the large language model to produce a coherent response based on the retrieved information. In the question for the large language model, it is specified that the response should be in JSON format, therefore the metadata can be handled and stored as JSON objects. The extraction of the course often also contains other irrelevant information like “Business informatics” therefore the result is streamlined through if queries (see Listing 6).

Listing 6: Function “check_course” (metadataExtraction.py)

```
# Function to check the course
def check_course(course):
    course = course.replace('-', ' ').lower()

    if course in 'it projekt' or course in 'it project':
        return 'PJ IT-Projekt Wirtschaftsinformatik'

    if course in 'ps information engineering':
        return 'PS Information Engineering & Management'

    if course in 'se information engineering &':
        return 'SE Information Engineering & Management'

    if course in 'se information engineering':
        return 'SE Information Engineering'

    if course in 'se seminar in':
        return 'SE Seminar in Planung und Gestaltung der Digitalisierung'

    return None
```

For the persons and the literature only the id of the database objects is stored in the list of the paper object. Therefore, for each person and literature a check if performed if its already present in the database and then the object must either be created or the id must be read out and saved (see Listing 7 – for literature function handle_literature is alike). Furthermore, for persons a distinction between the different types of persons (supervisor or student) must also be made.

Listing 7: Function “handle_person” (metadataExtraction.py)

```
# Function to check if a person is already present in the database - if not
create it - and return their id
```

```
def handle_person(name, personType):
    # Make http request to get the person
    jsonTemp = requests.get(url=pocketbaseCollectionURL + '/person/records',
params={'filter': f'name="{name}"&&person_type="{personType}"'}).text

    # Deserialize JSON into a Python dictionary
    dataDict = json.loads(jsonTemp)

    # Extract the 'items' part
    items = dataDict.get('items')

    # Check if the items part is empty -> create the person
    if len(items) == 0:
        newPerson = Person(name=name, person_type=personType)
        response = requests.post(url=pocketbaseCollectionURL +
'/person/records/', headers=header, data=serialize_object(newPerson)).text
        dict = json.loads(response)
        return dict.get('id')

    # Convert the list into a dictionary
    dict = items[0]

    # Get the value of the 'id' key from the dictionary
    return dict['id']
```

In order to get the literature of paper the large language model is used with the pdf file with only the sources as context and a different question. The procedure for the execution of the large language model stays the same. Afterwards, the response with the different JSON objects is handled in the function extract_literature_objects (see Listing 8).

Listing 8: Function “extract_literature_objects” (metadataExtraction.py)

```
# Function to extract literature objects form the llm response
def extract_literature_objects(text):
    startIndex = text.find('[')
    endIndex = text.rfind(']') + 1

    # Extract the JSON portion
    jsonData = text[startIndex:endIndex]

    # Parse the JSON string into a Python object (list of dictionaries)
    literatureData = json.loads(jsonData)

    # Create instances of the Literature class for each dictionary in the list
    literatureObjects = []
```

```
    for item in literatureData:
        literature_object = Literature(
            title=item['title'],
            authors=item['authors'],
            date=item['date'],
            doi=item['doi'],
            url=item['url']
        )
        literatureObjects.append(literature_object)

    return literatureObjects
```

Last but not least, the page and the word count are calculated. Due to the fact that the PyPDFLoader splits the pdf file into a list of documents, where each documents represents a single page in the pdf file, the page count can be calculated by taking the length of the list. For the word count the text of each page is split into a list of the words (whitespace is used as separator) and then the length of the list is added to the total number (see Listing 9). Then the paper object is returned and stored in the database.

Listing 9: Function “calculate_word _count” (metadataExtraction.py)

```
# Function to calculate the word count
def calculate_word_count(pdfFile):
    word_count = 0
    for page in pdfFile:
        text = page.page_content
        words = text.split()  # Split text into words
        word_count += len(words)   # Add the number of words

    return word_count
```

Information gaining from metadata

4.3.3. Frontend

4.3.4. Dashboarding

4.3.5. Database

4.3.6. Data model

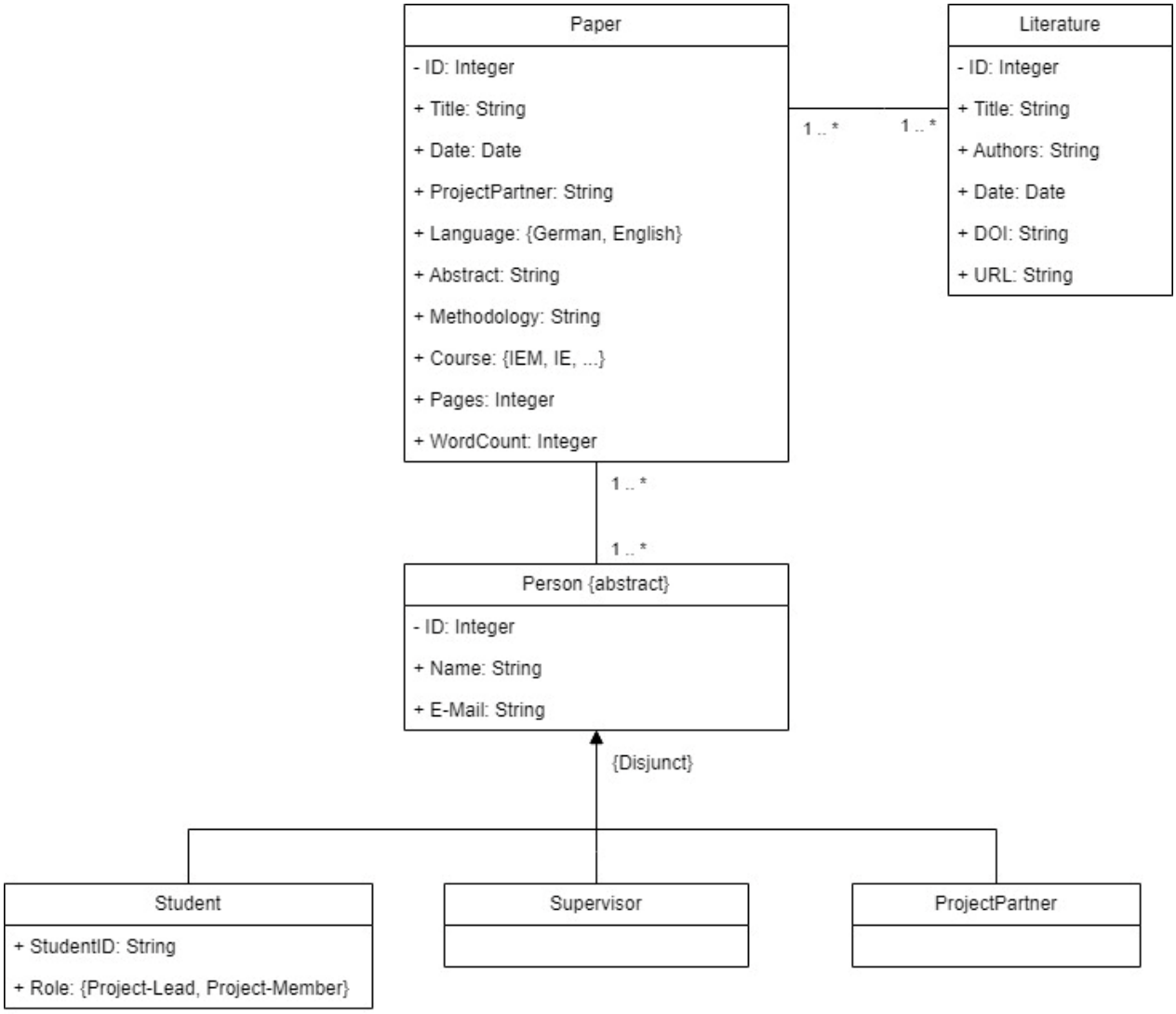


Figure 2: UML

The Figure 2 shows the data model of the project. First, the generalization of the persons can be seen here. The most essential data structures are recorded in the parent class “Person”. This class has a unique ID, name, and email address. The type of these data structures is self-explanatory. These data will be inherited by the child classes “Student”, “Supervisor”, and „ProjectPartner“. “Student” is the only child class which has additional data structures. The first data structure is “StudentID”. Since each student has a unique student ID, this data structure must also be unique. Additionally, the type of this data structure is a string which allows to append a “k” in the student ID. The next data structure “Role” marks the role of the student which can be either a project lead or a project member. Since a student can only have one of these two roles, the type of this data structure is an enumeration. The important thing about this generalization is that person instances are complete and disjoint, meaning that a person instance can only be either a student, supervisor, or a project partner.

The next table of the data model is “Paper” which stores essential information about the papers. A paper has a unique ID, title, publication date, assigned project partner, language, abstract, methodology, course name, number of pages and number of words. The language of a paper written by the students can only be in English or German. Therefore, the type of the data structure is an enumeration.

The association between a paper and a person is one-to-n on each side which means that a person can be involved in one or many papers and a paper could be written by one or many students.

The last table is named “Literature” which is about the literature references within a paper. The literature table has a unique ID, title, one or more authors, date, DOI, and a URL. The type of these data structures is also self-explanatory.

The associations between the “Literature” and “Paper” tables are also one-to-n on each side. This means that a paper can have one or many literature references and a literature could be referenced by one or many papers.

4.3.7. Process model

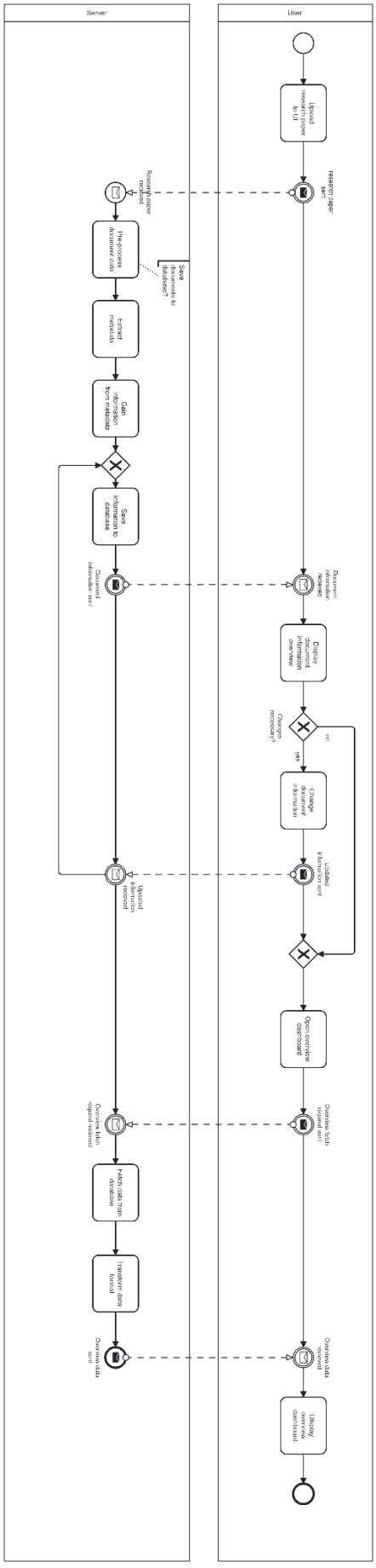


Figure 3: Process diagram

Figure 3 shows the main use case that is to be mapped as part of the partially automated meta-analysis. Essentially, the use case contains three different views. Course instructors and project managers, who interact with the system as users via the web client, and the server, which is responsible for extracting metadata and generating information.

The starting point for the illustrated process is the user who uploads a research paper via the web interface provided. The research paper is then transmitted to the server. As soon as the server has received the document, it is processed. The physical document is then first stored in the database. Extraction of the metadata then begins automatically.

As soon as this is complete, the server generates usable information from the metadata. The generated information is structured and stored in the database according to the database schema from chapter y. Once the information has been stored in the database, the server sends this information back to the client and thus to the user.

The user is shown the generated information and determines whether changes should be made manually or whether the information has been generated completely and correctly. If changes are required, the user adjusts the automatically generated information and then saves it. The saved information is then sent back to the server, where it is stored in the database in a structured manner. The information is then sent back to the user, who has the opportunity to adjust the data again.

If no changes are now required, the process is finished from the project manager's point of view. In the case of lecturers, it is also possible to open a dashboard view afterwards. This request is then sent to the server again. Once the server has received the information that the user wants to see the dashboard, the necessary data is loaded from the database. The data format is then adapted to meet the expectations of the dashboards on the web interface. This data is then sent back to the client, where the dashboard is displayed to the user once the data has been received.

4.4. Demonstration

Was wurde gemacht?

4.5. Evaluation

The evaluation of a design artifact is a central component of design science research and, together with the creation of the artifact, is one of the two most essential building blocks. Design science research requires evidence of the utility, quality and effectiveness of the artifact, which should be done using appropriate evaluation methods (Venable et al., 2016). An artifact is considered complete when it meets the requirements of the problem to be solved (Hevner et al., 2004).

In order to observe how suitable an artifact is for solving a particular problem, it is important to observe and measure this. This includes, among other things, comparing the defined objectives and the actual results of the artifact (Peffers et al., 2006). This in turn requires the definition of suitable metrics and the analysis of appropriate data (Hevner et al., 2004).

Once the artifact has been evaluated, there are several ways in which the project can continue. One possible outcome of the evaluation could be that the evaluation leads back to the design and development phase, where the artifact is improved (Peffers et al., 2006).

One of the challenges in the evaluation phase is to identify a suitable evaluation method (Elragal & Haddara, 2019). However, there are several methodologies in the literature that can be used for the evaluation of design artifacts. An excerpt of these can be seen in Figure 4.

The "Prototype" evaluation method was selected for this seminar paper. Prototypes are well suited to demonstrating the functionality, effectiveness and suitability of a solution to a specific problem (Peffers et al., 2012). A prototype represents possible proof that the design artifact functions as expected and that the required use cases can be represented (Peffers et al., 2012).

Logical Argument	An argument with face validity.
Expert Evaluation	Assessment of an artifact by one or more experts (e.g., Delphi study).
Technical Experiment	A performance evaluation of an algorithm implementation using real-world data, synthetic data, or no data, designed to evaluate the technical performance, rather than its performance in relation to the real world.
Subject-based Experiment	A test involving subjects to evaluate whether an assertion is true.
Action Research	Use of an artifact in a real-world situation as part of a research intervention, evaluating its effect on the real-world situation.
Prototype	Implementation of an artifact aimed at demonstrating the utility or suitability of the artifact.
Case Study	Application of an artifact to a real-world situation, evaluating its effect on the real-world situation.
Illustrative Scenario	Application of an artifact to a synthetic or real-world situation aimed at illustrating suitability or utility of the artifact.

Figure 4: Overview of evaluation methods (Peffers et al., 2012)

As part of this project, a prototype was developed to illustrate the semi-automated extraction and generation of information using an uploaded research paper. A web application was created for this purpose, which is made available to users to represent the relevant use cases.

…

5. Discussion

6. Conclusion

7. References

Adefowoke Ojokoh, B., Sunday Adewale, O., & Oluwole Falaki, S. (2009). Automated document metadata extraction. *Journal of Information Science*, *35*(5), 563–570. https://doi.org/10.1177/0165551509105195

Brüsemeister, T. (2008). *Qualitative Forschung: Ein Überblick* (2., überarb. Aufl). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-531-91182-3

Cheung, M. W.-L., & Vijayakumar, R. (2016). A Guide to Conducting a Meta-Analysis. *Neuropsychology Review*, *26*(2), 121–128. https://doi.org/10.1007/s11065-016-9319-z

Elragal, A., & Haddara, M. (2019). Design science research: Evaluation in the lens of big data analytics. *Systems*, *7*(2). Scopus. https://doi.org/10.3390/systems7020027

Fan, T., Liu, J., Qiu, Y., Jiang, C., Zhang, J., Zhang, W., & Wan, J. (2019). PARDA: A Dataset for Scholarly PDF Document Metadata Extraction Evaluation. In H. Gao, X. Wang, Y. Yin, & M. Iqbal (Eds.), *Collaborative Computing: Networking, Applications and Worksharing* (pp. 417–431). Springer International Publishing. https://doi.org/10.1007/978-3-030-12981-1_29

Fidel, R. (1984). The Case Study Method: A Case Study. *Library and Information Science Research*, *6*, 273–288.

Formentini, G., Boix Rodríguez, N., & Favi, C. (2022). Design for manufacturing and assembly methods in the product development process of mechanical products: A systematic literature review. *International Journal of Advanced Manufacturing Technology*, *120*(7–8), 4307–4334. Scopus. https://doi.org/10.1007/s00170-022-08837-6

Guo, M., Wu, F., Jiang, J., Yan, X., Chen, G., Li, W., Zhao, Y., & Sun, Z. (2023). Investigations on Scientific Literature Meta Information Extraction Using Large Language Models. *2023 IEEE International Conference on Knowledge Graph (ICKG)*, 249–254. https://doi.org/10.1109/ICKG59574.2023.00036

Han, H., Giles, C. L., Manavoglu, E., Zha, H., Zhang, Z., & Fox, E. A. (2003). Automatic document metadata extraction using support vector machines. *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, 37–48. https://doi.org/10.1109/JCDL.2003.1204842

Harling, K. (2012). An Overview of Case Study. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2141476

Hasan Choudhury, M., Jayanetti, H. R., Wu, J., Ingram, W. A., & Fox, E. A. (2021). Automatic Metadata Extraction Incorporating Visual Features from Scanned Electronic Theses and Dissertations. *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 230–233. https://doi.org/10.1109/JCDL52503.2021.00066

Heale, R., & Twycross, A. (2018). What is a case study? *Evidence Based Nursing*, *21*(1), 7–8. https://doi.org/10.1136/eb-2017-102845

Heinrich, L. J., Heinzl, A., & Riedl, R. (2011). *Wirtschaftsinformatik: Einführung und Grundlegung* (4., überarb. und erw. Aufl). Springer. https://link.springer.com/10.1007/978-3-642-15426-3

Herrera-Urtiaga, A. P., Torres-Cruz, F., Mendoza-Mollocondo, C. I., Paredes-Quispe, J.-R., & Chambi-Mamani, E. W. (2023). Automatic Extraction of Metadata Based on Natural Language Processing for Research Documents in Institutional Repositories. In Y. Iano, O. Saotome, G. L. Kemper Vásquez, M. T. de Moraes Gomes Rosa, R. Arthur, & G. Gomes de Oliveira (Eds.), *Proceedings of the 8th Brazilian Technology Symposium (BTSym'22)* (pp. 189–197). Springer International Publishing. https://doi.org/10.1007/978-3-031-31007-2_19

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. JSTOR. https://doi.org/10.2307/25148625

Hong, Z., Ward, L., Chard, K., Blaiszik, B., & Foster, I. (2021). Challenges and Advances in Information Extraction from Scientific Literature: A Review. *JOM*, *73*(11), 3383–3400. https://doi.org/10.1007/s11837-021-04902-9

Kartchner, D., Ramalingam, S., Al-Hussaini, I., Kronick, O., & Mitchell, C. (2023). Zero-Shot Information Extraction for Clinical Meta-Analysis using Large Language Models. In D. Demner-fushman, S. Ananiadou, & K. Cohen (Eds.), *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks* (pp. 396–405). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.bionlp-1.37

Kuckartz, U. (2018). *Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung* (4. Auflage, Issue ISBN: 9783779936824). Beltz Juventa. https://permalink.obvsg.at/AC15053340

Lindner, D. (2020). *Forschungsdesigns der Wirtschaftsinformatik: Empfehlungen für die Bachelor- und Masterarbeit*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-31140-7

Liu, R., Gao, L., An, D., Jiang, Z., & Tang, Z. (2018). Automatic document metadata extraction based on deep networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics), *10619 LNAI*, 305–317. Scopus. https://doi.org/10.1007/978-3-319-73618-1_26

Mayring, P. (2010). Qualitative Inhaltsanalyse. In G. Mey & K. Mruck (Eds.), *Handbuch Qualitative Forschung in der Psychologie* (pp. 601–613). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-531-92052-8_42

Mayring, P. (2015). *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (12., überarbeitete Auflage, Issue ISBN: 9783407257307). Beltz. https://permalink.obvsg.at/AC12177733

Mayring, P., & Fenzl, T. (2019). Qualitative Inhaltsanalyse. In N. Baur & J. Blasius (Eds.), *Handbuch Methoden der empirischen Sozialforschung* (pp. 633–648). Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-21308-4_42

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2018). Information extraction from scientific articles: A survey. *Scientometrics*, *117*(3), 1931–1990. https://doi.org/10.1007/s11192-018-2921-5

Palinkas, L. A. (2014). Qualitative and Mixed Methods in Mental Health Services and Implementation Research. *Journal of Clinical Child & Adolescent Psychology*, *43*(6), 851–861. https://doi.org/10.1080/15374416.2014.910791

Peffers, K., Rothenberger, M., Tuunanen, T., & Vaezi, R. (2012). Design science research evaluation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7286 LNCS*, 398–410. Scopus. https://doi.org/10.1007/978-3-642-29863-9_29

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). *The design science research process: A model for producing and presenting information systems research*. 83–106.

Peffers, K., Tuunanen, T., & Niehaves, B. (2018). Design science research genres: Introduction to the special issue on exemplars and criteria for applicable design science research. *European Journal of Information Systems*, *27*(2), 129–139. https://doi.org/10.1080/0960085X.2018.1458066

Rasmussen, N., Chen, C. Y., & Bansal, M. (Eds.). (2012). *Business Dashboards: A Visual Catalog for Design and Deployment* (1st ed.). Wiley. https://doi.org/10.1002/9781119197904

Riley, J. (2017). *Understanding metadata: What is metadata, and what is it for?* NISO Press.

Rössler, P. (2017). *Inhaltsanalyse* (3., völlig überarbeitete Auflage). UVK Verlagsgesellschaft mbH mit UVK/Lucius.

Schneijderberg, C., Wieczorek, O., & Steinhardt, I. (2022). *Qualitative und quantitative Inhaltsanalyse: Digital und automatisiert: eine anwendungsorientierte Einführung mit empirischen Beispielen und Softwareanwendungen* (1. Auflage). Beltz Juventa.

Schögel, M., & Tomczak, T. (2009). Fallstudie. In C. Baumgarth, M. Eisend, & H. Evanschitzky (Eds.), *Empirische Mastertechniken: Eine anwendungsorientierte Einführung für die Marketing- und Managementforschung* (pp. 79–105). Gabler Verlag. https://doi.org/10.1007/978-3-8349-8278-0_3

Shah-Mohammadi, F., & Finkelstein, J. (2024). Large Language Model-Based Architecture for Automatic Outcome Data Extraction to Support Meta-Analysis. *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*, 0079–0085. https://doi.org/10.1109/CCWC60891.2024.10427829

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., & Bolikowski, Ł. (2015). CERMINE: Automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, *18*(4), 317–335. https://doi.org/10.1007/s10032-015-0249-8

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: A Framework for Evaluation in Design Science Research. *European Journal of Information Systems*, *25*(1), 77–89. Scopus. https://doi.org/10.1057/ejis.2014.36

Weibel, S. L., & Koch, T. (2000). The Dublin Core Metadata Initiative: Mission, Current Activities, and Future Directions. *D-Lib Magazine*, *6*(12). https://doi.org/10.1045/december2000-weibel

Wexler, S., Shaffer, J., & Cotgreave, A. (2017). *The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios* (1st ed.). Wiley. https://doi.org/10.1002/9781119283089

8. List of Tables

Table 1: Dublin Core Elements (Weibel & Koch, 2000) 8
Table 2: Metadata Elements categorized by type (Riley, 2017) 9

9. List of Figures

Figure 1: View "upload-paper" 14
Figure 2: UML 20
Figure 3: Process diagram 22
Figure 4: Overview of evaluation methods (Peffers et al., 2012) 24

10. Appendices

User Manual

Data collected (in a raw form or tables)

Source code

…