Moving towards a General Metadata Extraction Solution for Research Data with State-of-the-Art Methods

Benedikt Heinrichs[a] and Marius Politze[b]
IT Center, RWTH Aachen University, Seffenter Weg 23, Aachen, Germany

Keywords: Research Data Management, Metadata Extraction, Metadata Generation, Linked Data.

Abstract: Many research data management processes, especially those defined by the FAIR Guiding Principles, rely on metadata for making it findable and re-usable. Most Metadata workflows however require the researcher to describe their data manually, a tedious process which is one of the reasons it is sometimes not done. Therefore, automatic solutions have to be used in order to ensure the findability and re-usability. Current solutions only focus and are effective on extracting metadata in single disciplines using domain knowledge. This paper aims, therefore, at identifying the gaps in current metadata extraction processes and defining a model for a general extraction pipeline for research data. The results of implementing such a model are discussed and a proof-of-concept is shown in the case of video-based data. This model is basis for future research as a testbed to build and evaluate discipline-specific automatic metadata extraction workflows.

1 INTRODUCTION

Research Data Management (RDM) is a central field in today's universities and is something every researcher should come across. With current state-of-the-art methods institutions are trying to up their field in providing good ways for researchers to make their data findable, accessible, re-usable and accessible according to the FAIR Guiding Principles described by (Wilkinson et al., 2016). Recent works like the creation of a draft for an interoperability framework by the EOSC, described by (Corcho et al., 2020), show the interest and necessity of standards. They describe the need of fulfilling the FAIR Guiding Principles even when focusing on semantic interoperability for describing data in a way that other machines can understand its purpose and meaning. Especially looking at re-usability, there is therefore a clear need of having a description of the content of data, so that people or machines finding it know what they are looking at. The current state on what such a person will find is however administrative things like when the data was created, to which organization it is assigned or who holds the rights to it as schemas like Dublin Core, described by (DCMI Usage Board, 2002), are widely used standards for it. While this information can be very useful, it does not help in making the data searchable or understandable on a deeper level. The only way of figuring out what it is about is by looking at the data directly, contacting the creator or with some luck stumbling across some documentation file or paper that describes it. This further creates an issue when looking at the research data-life-cycle research data moves through shown in figure 1 and described by (Schmitz and Politze, 2018). Since data moves without trace between some of those phases it becomes even more important to have a good representation of the data whenever the opportunity presents itself for having some chance to try and understand the path it took. Therefore, there is a clear need for descriptive research data that has a description of the content for tackling these problems. Since this is furthermore a very tedious or near impossible task to do manually, automatic ways would greatly reduce the workload on a researcher and improve the quality of research data. However, research data is not uniformly, so while domain-dependent automatic methods might work on some research data, a general automatic method that can include domain-dependent methods is necessary in tackling the issues for every data type. This paper therefore will focus on the general automatic creation of so-called metadata that also describes the content in hopes of being a start to solve the mentioned issues.

[a] https://orcid.org/0000-0003-3309-5985
[b] https://orcid.org/0000-0003-3175-0659

Heinrichs, B. and Politze, M.
Moving towards a General Metadata Extraction Solution for Research Data with State-of-the-Art Methods.
DOI: 10.5220/0010129502270234
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 1: KDIR*, pages 227-234
ISBN: 978-989-758-474-9

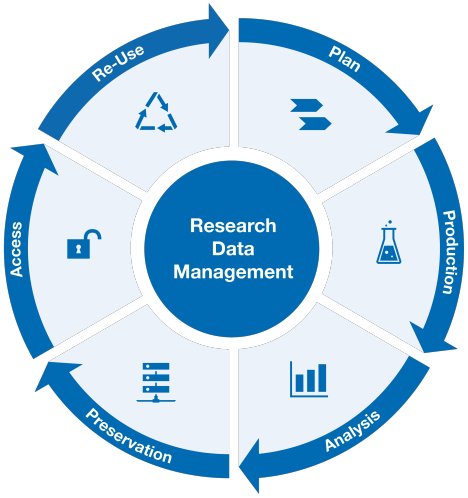

Figure 1: Research data-life-cycle of the RWTH Aachen University.

2 CURRENT STATE AND RESEARCH GOAL

The creation of metadata is a significant task to make the described research data re-usable and findable. Metadata can mean many things, however in this case it is meant as the description of an entity in the research data or the accumulation of any number of such descriptions. In this paper, metadata is divided into three distinct types: administrative, technical and descriptive. The definitions are described by (Lubas et al., 2013) and detailed in the following:

- Administrative Metadata.
  Description of the administrative elements that are necessary in determining e.g. who is responsible for a certain data entity.
- Technical Metadata.
  Description of the necessary information that describes how a data entity came about, where it is currently located and how to use it.
- Descriptive Metadata.
  Description of the elements that describe a data entity directly. This means that these elements can be file attributes like file size, file name and other delivered properties like creation date or more detailed information like creator and title. Furthermore, for this paper especially the content information as metadata is significant to be described in logical triples like subject “Sample AS001”, predicate “uses chemical” and object “Tetraethylorthosilicate” which are derived from a data entity and currently manually described in projects like (Politze et al., 2019b).

Following the discussed problems, the metadata type being focused on is descriptive metadata. However, it can be noted that certain created information as descriptive metadata can help the creation of administrative or technical metadata.

2.1 Current State on Metadata Extraction

Current methods of metadata extraction more than often focus on the researchers to describe their research data manually as described in (Politze et al., 2019a). Such values can be discipline-specific and include what title a research project has, who has the rights to it or what subject area it belongs to but can also specify e.g. the microscope which was being used, depending on the schema. In (Jane Greenberg, 2004) and (Mattmann and Zitting, 2011) the researchers even create descriptive metadata automatically based on the attributes of data as files or using META tags. This however leads to the content of research data being largely ignored in the process. Therefore, a lot of information that could be derived by a description of the content as metadata is lost. For this reason, researchers in single domains try to look into the issue of extracting more descriptive metadata from their research data by analyzing the content. In (Burgess and Mattmann, 2014), (Rodrigo et al., 2018) and (Grunzke et al., 2018) the researchers present their success in their specific domains by creating specific models and generating descriptive metadata for their use-case. The problem with this however is, that this works as long as the research data being analyzed is uniformly similar. In the broad context of research data management this however is not the case, which is why the case for general metadata extraction is being made, so that metadata can be extracted from every research data entity.

2.1.1 Current State on Metadata Representation

Metadata can be represented in widely different ways and be in different shapes or forms. However, looking specifically at research data, a trend can be detected in the usage of linked data and representing metadata in the RDF (Resource Description Framework) format, described by (Cyganiak et al., 2014). Standards, like Dublin Core described in (DCMI Usage Board, 2002) or EngMeta which is specifically created for research data from the Engineering Sciences and described in (Iglezakis and Schembera, 2019), are organized in this format, giving an additional benefit for using it. Furthermore, metadata in this format can be seen as a knowledge graph that can be searched and

contains information about the represented data entity. In this paper, metadata will be therefore looked as represented by RDF.

2.2 Automatic Metadata Extraction

Since there are multiple methods metadata can be extracted from different types of data, some general methods are discussed in the following.

2.2.1 Generic Metadata Extraction

Generic metadata extraction is here currently understood as the extraction of descriptive metadata based on file attributes. This means the focus lies on fields like creator, creation date and many more. One tool for extracting this kind of metadata is called Apache Tika. It was first presented by (Mattmann and Zitting, 2011) and is a tool that can extract the attributes of a file and represent them in a JSON format. Furthermore, given a text-based format, or a suitable adapter for a format, it can extract the text content of a file. The output however does not conform to the RDF format and has to be transformed to be represented as linked data.

2.2.2 Metadata Extraction from Text

Since the representation of a file can be at many times a text or at least understood as text, methods which create metadata based on text are significant and one is called Pikes. It is a tool which was first presented by (Corcoglioniti et al., 2016) and uses natural language processing techniques to convert text to a rich metadata representation in RDF as linked data. This representation includes the content relation using NLP (natural language processing) techniques, links from entities to DBPedia described by (Lehmann et al., 2015) and much more. Such metadata can make the text understandable by machines and can be seen as a representation of descriptive metadata which contains the content information.

2.3 Text Extraction

Since a metadata extraction method that can create the wanted descriptive metadata from text is established, methods that represent other data types as text are looked at to retrieve further descriptive metadata from them. Research and work has already been done on certain types and is presented in the following.

2.3.1 Image Text Extraction

Images, for example from microscopes, might contain hints on the current state of the microscope or the looked entity during an experiment. Therefore, it is important to extract the text from such images. This is not a new task and solutions exist for this problem. One solution is a tool called Tesseract which was first presented by (R. Smith, 2007) and provides a so called OCR engine that can extract text from images. Training this engine correctly can result in a good method for this and further use cases. Furthermore, Tesseract can be integrated in Apache Tika, which was described in section 2.2.1, and be used to get the text representation of an image file.

2.3.2 Audio Text Extraction

With a lot of focus on text-to-speech services, the other speech-to-text way is also being explored more and more. In (Kumar and Singh, 2019) the researchers look at different models and represent the current state-of-the-art. Furthermore, with cloud providers like Google (https://cloud.google.com/speech-to-text) presenting the option, it was never so simple to use such a method and work with the results.

2.4 Challenges

The aim of this paper is to build a general workflow to extract machine-readable metadata from the content of a research data entity. From the description of the current state-of-the-art and this aim, the following challenges could be detected:

- Manual metadata creation for every data entity is not a feasible task
- Solutions for some domain-dependent metadata extraction exist, but there is no solution which tries to combine them
- There are many data formats where no specific metadata extraction method exists
- Even if text can be represented as descriptive metadata, the challenge still remains to see how some data formats can be represented as text
- Generic metadata extraction solutions like Apache Tika do not produce an output which is represented as linked data

3 APPROACH

Based on the current state-of-the-art and identified research gaps in this chapter a highly configurable model is proposed for extracting descriptive metadata. The idea is that a pipeline is created which receives a data entity, represented as any number of files, as an input and stores the collected metadata as the output directly in a metadata store. Since research data in particular can be unique and many data types are necessary to consider, even application specific data types, the model is intended to be as configurable and generic as possible so that custom extractors can be used to extend the pipeline with custom data formats. This in particular tackles the issue of previous research only tackling very specific use cases by opening this method up to any data type imaginable. Furthermore, since a lot of data can be represented as text, a custom extraction method does only have to provide the text representation and not even deal with the metadata representation since a metadata from text extraction method can always be added to the pipeline. However, if in any step metadata is produced, it is important the corresponding RDF format is added as an output as well, so that it can be processed in the last step. The proposed model goes through the following steps:

1. The data entity starts with the generic extraction step, extracting metadata based on file attributes and text, if possible, by any number of configurable extractors.
2. Based on the data type any number of configurable MIME-Type-based extraction methods are called for extracting metadata and a text representation of the input.
3. The text representation runs through any number of configurable text-based metadata extractors for getting the final content-based metadata.
4. The produced metadata gets mapped to a defined format and verified for correctness (e.g. missing entries).

A visualization of this model can be seen in figure 2 and the concrete steps are described in the following.

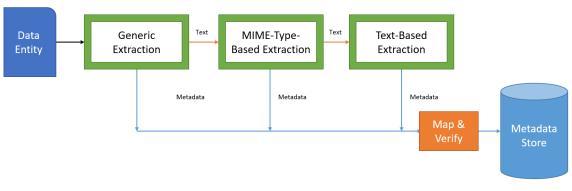

Figure 2: Data Entity Extraction Pipeline.

3.1 Generic Extraction

The generic extraction step is initially the most important step in the pipeline, since this step shapes all further steps. The requirements for a generic extraction method are that it accepts as input a file, an identifier and a collection of configuration values. Furthermore, a generic extraction method has to fulfill multiple requirements for the outputs it has to return, which are:

- Extract the descriptive metadata from a file using the file attributes.
  This means that elements which a file might already provide information for, like creator, creation date, file name and many more, are collected and represented in a metadata schema.
- Detecting the MIME-Type of a file.
  The MIME-Type of a file represents in what format the data is structured. Detecting this is essential for continuing the process since it determines which MIME-Type-based extractors will be called next.
- Extracting the text from text-based MIME-Types. This is optional, however it greatly reduces complexity in the next steps when the generic extraction method already extracts the textual elements it finds from text-based MIME-Types.

3.2 MIME-Type-based Extraction

The specific custom implementations all follow the same interface which requires them to register their targeted MIME-Types to the pipeline. Depending on if any of their targeted MIME-Types are represented by the current data entity, they are executed. They receive as input a file, an identifier and a collection of configuration values. The main goal of an extraction method is either to extract text from the data by using domain knowledge or extracting metadata directly. As examples there can be text in images which can be extracted, an image can be described with the displayed objects or audio which contains speech can be converted to text by state-of-the-art methods. These results, especially the text representation of the certain input type, are used and moved to the last step and therefore that representation becomes even more important to be extracted by the MIME-Type-based extraction method.

3.3 Text-based Extraction

The final extraction step is the text-based metadata extraction. The idea here is that after the data entity has passed through the other steps, a text representation of some kind for the content exists, even if

the input was not textual data before. Therefore, every text-based extraction method receives additionally to the file, an identifier and a collection of configuration values also the currently extracted text. A lot of research has been done on extracting metadata, especially as linked data, from text, therefore this property can be exploited in this step. Such methods make use of natural language processing and in general use of the relations and grammar in a text. An implementation here should use the text and convert it to a metadata representation that describes what important information the content contains. An example can be to retrieve the most important topics using topic extraction or describing the relations of certain nouns in the text by utilizing the natural language grammar.

3.4 Metadata Representation

The final resulting metadata has to be represented uniformly, so that any use can be drawn out of it by machines. The input for this final step are a number of metadata sets in any RDF format that first have to be merged into a singular metadata set. As previously discussed, the metadata is represented as linked data and therefore as triples which have a subject, predicate and object. Every entry in the metadata is linked to the subject that describes the incoming file. However, a filename is not a unique identifier, so the model requires an additional identifier to be present, otherwise a random identifier will be generated. Using a combination of some specified prefix and the identifier, the subject is created. This creates a linked data knowledge graph about the input data entity. It is however not possible for every extraction method to create metadata with the same quality or quantity since different data types and extraction methods produce vastly different results. Therefore, it is necessary for every extraction method to produce and describe their representation of data following commonly accepted structures. Furthermore, certain fields can be defined in there as requirements for the metadata set, so that some common ground can exist between created metadata sets. An option for such a requirement enforcement is the Shapes Constraint Language (SHACL) which was specified by (Knublauch and Kontokostas, 2017). This language can be used to form a so-called application profile which defines the requirements on the metadata set and with it, the resulting graph can be verified. Since the graph might not be in the needed structure, an implementation can be provided to restructure and map the triples in the graph to the needed ones. There are a couple of entries from the Dublin Core schema which should be always present like http://purl.org/dc/elements/1.1/creator, http://purl.org/dc/elements/1.1/title or http://purl.org/dc/elements/1.1/identifier so it is checked, if they exist in the graph. Then furthermore, a configured ruleset as application profile can be taken and applied to the graph for verifying if the resulting metadata conforms to what is required. If this is true, the metadata should be stored.

4 PRELIMINARY RESULTS

This section will discuss the first implementation of the proposed model as a proof-of-concept and the preliminary results of it. The used tools for each step will be discussed and the usage of them and modifications on them will be explained. It is to be noted that this will not cover every single implementation, only the most general one and one concrete example in the case of video files.

4.1 Current Concrete File Extraction Pipeline

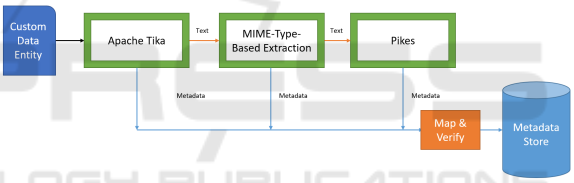

Figure 3: Concrete File Extraction Pipeline.

In figure 3 the concrete implementation of the proposed model is shown. There are two concrete implementations listed which replace the proposed steps, “Apache Tika” for “Generic Extractors” and “Pikes” for “Text-Based Extraction” since both of them follow the requirements. The MIME-Type-based Extraction still remains open, since this ensures compatibility to every data type.

4.2 Case Study - Video

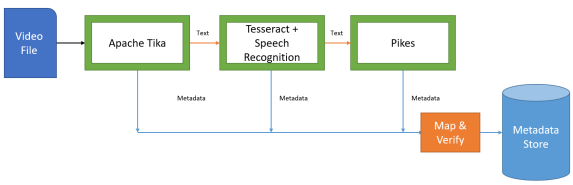

Figure 4: Video File Extraction Pipeline.

As a proof-of-concept a video file is used which contains text and spoken audio for showing the steps in a realistic scenario. For choosing a video, research

results from a recent published work is being used. The concrete video file being used is from the work of (Heilmann et al., 2020) and is available as the second supplement material at https://doi.org/10.23641/asha.12456719.v1. The resulting pipeline of using a video file can be seen in figure 4. The MIME-Type-based Extraction step utilizes Tesseract and Speech Recognition. Each step and what the output is, will be discussed in the following.

4.2.1 Usage of Apache Tika

As described in section 2.2.1, Apache Tika can make an ideal fit for the generic extraction. How it performs regarding the requirements is discussed in the following:

- Extract the descriptive metadata from a file using the file attributes.
  When passing a file to Apache Tika, it locates all descriptive metadata a file has to offer using the file attributes and returns them. This is in this proof-of-concept information like “dcterms:created” with its value “2014-05-30T16:13:18Z”.
- Detecting the MIME-Type of a file.
  Apache Tika returns the MIME-Type of a file, even if it was not specified by the filename extension. In this proof-of-concept it is “video/mp4”.
- Extracting the text from text-based MIME-Types.
  One of the return values from Apache Tika is the content which represents the file as text. Furthermore, with the use of technologies like Tesseract, text can also be retrieved from images with Apache Tika. Since for this proof-of-concept a video file is being used, there is however no text extracted.

The requirements are therefore all fulfilled. One thing which however still needs to be done is to transform the output. Apache Tika produces a JSON structure which does not completely conform to the RDF format. There are however some links which are done by some keys that represent an ontology like Dublin Core. For utilizing this, the values containing such a key are linked to the subject of the file directly as predicate (key) and object (value). In this proof-of-concept one such triple can look like: “file:{identifier} dcterms:created ’2014-05-30T16:13:18Z”’ where “file” is an example prefix. A challenge is to convert every other key-value-pair so that it can be described in the RDF format and stored in a metadata store. One idea here for the proof-of-concept is that the keys which do not conform to an ontology and are not URLs are added to a constant prefix, e.g. “http://example.org/tika/{key}”. This can then be used in the proof-of-concept with a triple like: “file:{identifier} tika:Content_Type ’video/mp4”’. Further work still has to be done on mapping the results from Apache Tika accurately to an ontology and verifying the results using an application profile. After the conversion, a concrete metadata entry can be pushed further to the metadata store.

4.2.2 Usage of Tesseract and Speech Recognition

It is a difficult task to extract metadata from video files normally, since the information can only be distinguished frame by frame, but even then there is the issue of audio and images which need to be analyzed. For this proof-of-concept, the model gets split into an audio and image part so that the input can still be processed. First the audio gets analyzed by specific audio file extractors which transform the audio into a text representation of the spoken text by using speech-to-text methods. This results in sentences being extracted like “This short video is an example of [...]”. Next, the images of every frame are passed to image file extractors which extract e.g. the text shown in an image with Tesseract and describe the objects shown in an image in a metadata representation. With respect to frames being similar over the cause of time, the metadata and text representations of the images are combined. In this proof-of-concept this results in extracted text like “CONVERSATION - SCHOOL AGE” as seen in figure 5 and metadata like “file:{identifier} foaf:depicts imageobject:clock” and “imageobject:clock imageobject:count 1” where “imageobject” is an example prefix and links to the frame shown in figure 6.

Figure 5: Video frame with the text “‘CONVERSATION - SCHOOL AGE”.

Figure 6: Video frame with a clock.

4.2.3 Usage of Pikes

As a text-based extraction method, the tool Pikes is being used as described in section 2.2.2. The accumulated text gets pushed to Pikes either all at once, or depending on the number of sentences and a pre-configured threshold per batch. The received output contains the content relations, however also includes a lot of grammar information and definitions on where the words are placed in the text. Since the interest is here on describing the content and the value is low on the natural language semantics and concrete position of something, the output is filtered to only include the content relations, e.g. subject "http://pikes.fbk.eu/#child", predicate "http://dkm.fbk.eu/ontologies/knowledgestore#mod" and object "attr:school-age", and definitions for certain entities e.g. for "http://pikes.fbk.eu/#math" a DBPedia entry ("dbpedia:Mathematics") exists and is linked to it. Every relation is represented by Pikes in fact graphs, however storing them as numerous fact graphs and linking them to a file is creating a large number of triples without much benefit. Therefore, the fact graphs are merged into one fact graph with the prefix being a static one e.g. "http://example.org/factgraph/" combined with the file identifier. This factgraph is then linked to the file, completing text-based extraction.

4.2.4 Metadata Representation

In this proof-of-concept the created metadata does not have to be further modified and can be validated against an application profile which requires certain Dublin Core values to be present but is open to any number of further triples. This succeeds and the metadata is stored. Therefore, the model allows describing information extracted from a video as structured metadata that could be indexed by semantic search engines.

5 CONCLUSION

This paper describes the need of a general way to extract descriptive metadata which works regardless of the type and can be extended. As a starting point, a pipeline is proposed that allows the extraction of RDF-based descriptive metadata to represent a provided data entity. It offers extension points to become agnostic to the representation of the information in the data entity, ranging from textual data to audiovisual representations. The proposed model provides a solution in generalizing the ways metadata can be extracted and clear requirements, necessary modules and a structure for making it work are discussed. A first implementation of the model is shown with state-of-the-art technology and the case of video files is shown as a proof-of-concept.

5.1 Identified Research Gaps

The created pipeline was an experimental setup which can now be tried in practice. This paper describes the setting and provides the model as a first way of dealing with the current challenges. However, some further gaps which need to be looked into were discovered and are formulated as questions in the following:

1. Following the generic extraction methods, how should the resulting metadata look like and how should the results be represented or mapped?
2. How can such a proposed pipeline be evaluated?
3. There are many data types which such a proposed pipeline could support, but which data types really are the ones which are needed for strictly working with research data?
4. The here used proof-of-concept extracts metadata from a video file. During this process the pipeline extracts images and extracts text and metadata from them. How can such a video file be compared to an image using the metadata and how can it be detected that they are similar and one is stemming from the other?
5. Since there is a large variety of data, how can a mapping of the resulting metadata to an application profile be created?

5.2 Future Work

Building on this work, the extraction methods will be enhanced so that more and more data types are supported. Deep learning could furthermore be used if a model can be created which represents research data as an input and metadata about it as an output. In an analysis it shows that specifically source code and numerical data are significant aspects for current research data and therefore should be included. Furthermore, the nature of the created pipeline allows for previous discipline-specific methods to be integrated. Looking at the resulted metadata, methods to map them more accurately to application profiles and specifying a concept in which often occurring predicates are mapped using specified rules to common standards will greatly increase the quality of the output. This can help to utilize the produced metadata in other domains like checking with such a representation the similarity to another data entity without the

need of looking at the whole content. Furthermore, reducing the produced metadata to only the most relevant and uniquely representing parts will increase the interpretability and comparability. Lastly, collecting such metadata on research data for a research project could be used to act as a knowledge graph of the whole research to identify what the research is actually about and discovering the novelty of it.

REFERENCES

Burgess, A. B. and Mattmann, C. A. (2014). Automatically classifying and interpreting polar datasets with apache tika. In Joshi, J., editor, *2014 IEEE 15th International Conference on Information Reuse and Integration (IRI)*, pages 863–867, Piscataway, NJ. IEEE.

Corcho, O., Eriksson, M., Kurowski, K., Ojsteršek, M., Choirat, C., van de Sanden, M., and Coppens, F. (2020). Eosc interoperability framework (v1.0): Draft for community consultation. https://www.eoscsecretariat.eu/sites/default/files/eosc-interoperability-framework-v1.0.pdf.

Corcoglioniti, F., Rospocher, M., and Aprosio, A. P. (2016). Frame-based ontology population with pikes. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3261–3275.

Cyganiak, R., Lanthaler, M., and Wood, D. (2014). RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

DCMI Usage Board (2002). Dcmi metadata terms. https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.

Grunzke, R., Hartmann, V., Jejkal, T., Kollai, H., Prabhune, A., Herold, H., Deicke, A., Dressler, C., Dolhoff, J., Stanek, J., Hoffmann, A., Müller-Pfefferkorn, R., Schrade, T., Meinel, G., Herres-Pawlis, S., and Nagel, W. E. (2018). The masi repository service — comprehensive, metadata-driven and multi-community research data management. *Future Generation Computer Systems*.

Heilmann, J., Tucci, A., Plante, E., and Miller, J. F. (2020). Assessing functional language in school-aged children using language sample analysis. *Perspectives of the ASHA Special Interest Groups*, 5(3):622–636.

Iglezakis, D. and Schembera, B. (2019). EngMeta - a Metadata Scheme for the Engineering Sciences.

Jane Greenberg (2004). Metadata extraction and harvesting. *Journal of Internet Cataloging*, 6(4):59–82.

Knublauch, H. and Kontokostas, D. (2017). Shapes constraint language (SHACL). W3C recommendation, W3C. https://www.w3.org/TR/2017/REC-shacl-20170720/.

Kumar, Y. and Singh, N. (2019). A comprehensive view of automatic speech recognition system - a systematic literature review. In *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pages 168–173.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2015). Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Lubas, R. L., Jackson, A. S., and Schneider, I. (2013). Introduction to metadata. In Lubas, R. L., Jackson, A. S., and Schneider, I., editors, *The Metadata Manual*, Chandos Information Professional Series, pages 1–15. Chandos Publishing.

Mattmann, C. and Zitting, J. (2011). Tika in action.

Politze, M., Bensberg, S., and Müller, M. (op. 2019a). Managing discipline-specific metadata within an integrated research data management system. In Filipe, J., editor, *Proceedings of the 21st International Conference on Enterprise Information Systems ICEIS 2019, Heraklion, Crete - Greece, May 3 - 5, 2019*, ICEIS (Setúbal), pages 253–260, [S. l.]. SciTePress.

Politze, M., Schwarz, A., Kirchmeyer, S., Claus, F., and Müller, M. S. (2019b). Kollaborative Forschungsunterstützung : Ein Integriertes Probenmanagement. In *E-Science-Tage 2019 : data to knowledge / herausgegeben von Vincent Heuveline, Fabian Gebhart und Nina Mohammadianbisheh*, pages 58–67, Heidelberg. E-Science-Tage, Heidelberg (Germany), 27 Mar 2019 - 29 Mar 2019, Universitätsbibliothek Heidelberg.

R. Smith (2007). An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 629–633.

Rodrigo, G. P., Henderson, M., Weber, G. H., Ophus, C., Antypas, K., and Ramakrishnan, L. (2018). Sciencesearch: Enabling search through automatic metadata generation. In *2018 IEEE 14th International Conference on e-Science (e-Science)*, pages 93–104.

Schmitz, D. and Politze, M. (2018). Forschungsdaten managen – bausteine für eine dezentrale, forschungsnahe unterstützung. *o-bib. Das offene Bibliotheksjournal / Herausgeber VDB*, 5(3):76–91.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3:160018.