universität innsbruck

Masterarbeit
zur Erlangung des akademischen Grades
Master of Science
der Fakultät für Mathematik, Informatik und Physik
der Leopold-Franzens Universität Innsbruck

SciKGTeX - A LaTeX Package to Semantically Annotate Contributions in Scientific Publications

Verfasser: Christof Bless
Matrikel-Nr: 11915408

Betreuerin: Assoc. Prof. Dr. Anna Fensel
Co-Betreuer: Dr. Oliver Karras
Institut für Informatik
Semantic Technology Institute

Abgabedatum: 25. Mai 2022

Abstract

The continuously increasing output of published research makes the work of researchers harder as it becomes impossible to keep track of and compare the most recent advances in a field. Scientific knowledge graphs have been proposed as a solution to structure the content of research publications in a machine-readable way and enable more efficient, computer-assisted workflows for many research activities. Crowdsourcing approaches are used frequently to build and maintain such scientific knowledge graphs. Researchers are motivated to contribute to these crowdsourcing efforts as they want their work to be included in the knowledge graphs and benefit from applications built on top of them. To contribute to scientific knowledge graphs, researchers need simple and easy-to-use solutions to generate new knowledge graph elements and establish the practice of semantic representations in scientific communication.

In this thesis, I present SciKGTeX, a LaTeX package to semantically annotate scientific contributions at the time of document creation. The LaTeX package allows authors of scientific publications to mark the main contributions such as the background, research problem, method, results and conclusion of their work directly in LaTeX source files. The package then automatically embeds them as metadata into the generated PDF document. In addition to the package, I document a user evaluation with 26 participants which I conducted to assess the usability and feasibility of the solution.

The analysis of the evaluation results shows that SciKGTeX is highly usable with a score of 79 out of 100 on the System Usability Scale. Furthermore, the study showed that the functionalities of the package can be picked up very quickly by the study participants which only needed 7 minutes on average to annotate the main contributions on a sample abstract of a published paper. SciKGTeX demonstrates a new way to generate structured metadata for the key contributions of research publications and embed them into PDF files at the time of document creation.

Zusammenfassung

Die andauernde Steigerung der Publikationsrate von wissenschaftlichen Beiträgen erschwert es Forschenden weltweit den Überblick über die neusten Entwicklungen zu behalten und diese zu vergleichen. Um den Problemen, die mit dem exponentiellen Zuwachs von Publikationen einhergehen, entgegen zu wirken, wurden Wissensgraphen für Forschungsinhalte entwickelt. Solche Wissengraphen nutzen systematische Strukturierung und maschinenlesbare Formate, um effizientere, computergestützte Rechercheverfahren zu ermöglichen. Um Wissengraphen anzulegen und zu erweitern, wird die Hilfe von sachkundigen Experten benötigt, die neue Inhalte erstellen oder in vorhandenen Texten annotieren. Für Wissengraphen von Forschungsinhalten erstellen idealerweise die Forschenden selbst die Bestandteile des Graphen. Hierfür benötigen sie leicht zu benutzende Softwarelösungen, die sich einfach in ihren Arbeitsablauf integrieren lassen.

In dieser Masterarbeit stelle ich SciKGTeX vor, ein LaTeX Paket zur semantischen Annotation von wissenschaftlichen Beiträgen während der Dokumenterstellung. Das LaTeX Paket erlaubt es Autoren von wissenschaftlichen Arbeiten, die wesentlichen Beiträge ihrer Arbeit direkt in LaTeX Quelldateien zu markieren. Das Paket bettet die Annotationen dann als Metadaten in das resultierende PDF Dokument ein. Die Arbeit beinhaltet zudem eine Nutzerevaluation mit 26 Teilnehmern, die Benutzerfreundlichkeit und Umsetzbarkeit von SciKGTeX umfassend beurteilt.

Die Ergebnisse der Evaluation zeigen, dass SciKGTeX mit 79 von 100 Punkten auf der System Usability Scale eine hohe Nutzerfreundlichkeit aufweist. Außerdem sind die Teilnehmer ohne wesentliche Vorkenntnisse in der Lage die wichtigsten Beiträge in der Kurzfassung einer wissenschaftlichen Publikation in durchschnittlich 7 Minuten zu annotieren. SciKGTeX demonstriert eine neue Möglichkeit die wichtigsten Beiträge einer wissenschaftlichen Arbeit direkt bei der Erstellung der Publikation als maschinenlesbare Metadaten in das Dokument einzubetten.

Acknowledgements

I want to thank my supervisors Dr. Anna Fensel and Dr. Oliver Karras for their help and the frequent feedback on the thesis. Especially, I want to thank Dr. Karras for the time he took for the weekly meetings during several months and all the feedback and support he provided in the process.

Furthermore, I want to express my gratitude towards Sören Auer from the Technische Informationsbibliothek (TIB) in Hannover for allowing me to work together with his group in the context of the ORKG project. Many of the colleagues at TIB have given me great feedback on SciKGTeX and helped me to achieve my evaluation goals. I hope the outcome of this work will ultimately contribute to their vision of an open scientific knowledge graph.

Lastly, I thank all the anonymous participants in the user evaluation of this work for their time and feedback. A specific thanks goes to Dr. Umutcan Şimşek for letting me present my thesis in his Seminar and gaining valuable insights as well as a few more evaluation participants.

Contents

List of Figures

List of Tables

List of Acronyms

DEO	Discourse Elements Ontology
DoCO	Document Components Ontology
HTML	Hypertext Markup Language
ORKG	Open Research Knowledge Graph
OWL	Ontology Web Language
PDF	Portable Document Format
RDF	Resource Description Framework
SALT	Semantically Annotated LaTeX
SUS	System Usability Scale
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universal Unique Identifier
WYSIWYG	What You See Is What You Get
XMP	Extensible Metadata Format

1 Introduction

In this chapter I will give the basic motivation for the work being presented in this thesis. I explain some of the problems with scientific publishing as it is today and present solutions to overcome these problems. Further, I will clearly define the goals which I want to achieve and explain where each of the components of the solution will be presented in greater detail within the text. The chapter is structured as follows. Section 1.1 contains the main motivation behind the thesis work together with the problem statement and the research questions. Based on the motivation I present the goals of the thesis in section 1.2 and explain the structure of the thesis in section 1.3

1.1 Motivation

Scientific discoveries have long become a community effort with sometimes hundreds of researchers from different institutions collaborating on the solutions to increasingly complex research problems. While problems and approaches in research have evolved greatly over the years, scientific communication has still a lot of potential to improve in the age of information. The standard process in scientific communication is to publish scientific articles which are archived and distributed as PDF files [2]. This is a very basic approach for digitisation of research content and does not leverage modern technologies which could pave the way to a computer-assisted knowledge exchange. Instead, with the immense number of published articles, it gets increasingly harder to keep an overview of the state-of-the-art in certain fields while at the same time reproducibility of research [3, 4] and quality of peer reviews have been stagnating continuously [5].

A modern innovation process begins by establishing the state-of-the-art in the field and an awareness of the most recent advancements on the specific research problem. Researchers have to find relevant work to build their research on, so they can develop new solutions. Furthermore, they can compare or try to reproduce the findings in these publications to gain new knowledge [6]. For this process it is impossible to take into account all the relevant works which are published, instead it is necessary to select certain articles based on keyword search or by skimming through works of peer-researchers, conference proceedings or renowned journals.

The approach of manually selecting related works does not scale well with the explosion of published material and becomes increasingly ineffective. Keyword searches might be too coarse and give thousands of results, or they might be too restrictive and miss a big number of relevant results. For example due to variability of jargon

and naming conventions at different institutions around the world, keyword searches potentially fail to deliver the desired result [7, 8].

Another problem can arise in the form of so called *filter bubbles* [9, 10]. Through the use of personalised profiles, search engines nowadays filter relevant information that best fits the user's profile. While this filtering was developed as a way to handle an overabundance of irrelevant information, it is not guaranteed to work as intended. For example, users may be overly presented with information stemming from their geographic or cultural region based on their geolocation information. These filter bubbles might become increasingly more relevant as large academic social networks are emerging such as ResearchGate[1] and Academia.edu[2] [11]. Apart from algorithmic bias, bubbles can also form due to the search behaviour of individuals. If researchers mainly consider publications from other research groups they know well (for example from the same university or country) or journals of their choice to limit their search space, they might inadvertently create an information bubble similar to the algorithmic filter bubbles. In extreme cases this can lead to different research groups working on the same or similar problems, being unaware of each other's progress and possibly producing redundant work. Filter bubbles can also lead to biased results in meta-analyses where researchers overlook relevant articles outside of their bubble.

Scientific knowledge graphs like the Open Research Knowledge Graph (ORKG) (see section 2.3) can alleviate these problems by representing research content in a semantically rich representation which allows more sophisticated methods of information extraction [12]. A well curated knowledge graph allows to find publications not just based on the choice of keywords which might or might not occur in the text. In a knowledge graph, search programs can access networks of terms and concepts which can be found in the articles' text. Different than the raw text, these graph networks contain a semantic representation of the content which is more structured and consistent. Knowledge graphs lay a solid foundation for a plethora of applications which can exploit such semantically enriched graph structures. Among the possible applications are enhanced document retrieval techniques [13], automated content comparisons [14], reasoning engines, autonomous research systems [15], mathematical proof assistants [16] and paper recommendation systems [17, 18].

For all of the stated problems and applications the widespread adoption of knowledge graph technologies in the field of research publication is a desirable development. To construct large-scale knowledge graphs of research contributions, one can not exclusively rely on automated techniques like entity relation extraction [6, 12]. The creation of complex, high quality knowledge graphs requires domain and ontology experts to define concepts and relations of the graph. These concepts and relations must be identified in research texts and then annotated so they can be extracted into a knowledge graph.

A common strategy to achieve annotation on a large scale is through crowdsourc-

[1] `https://www.researchgate.net/`

[2] `https://www.academia.edu/`

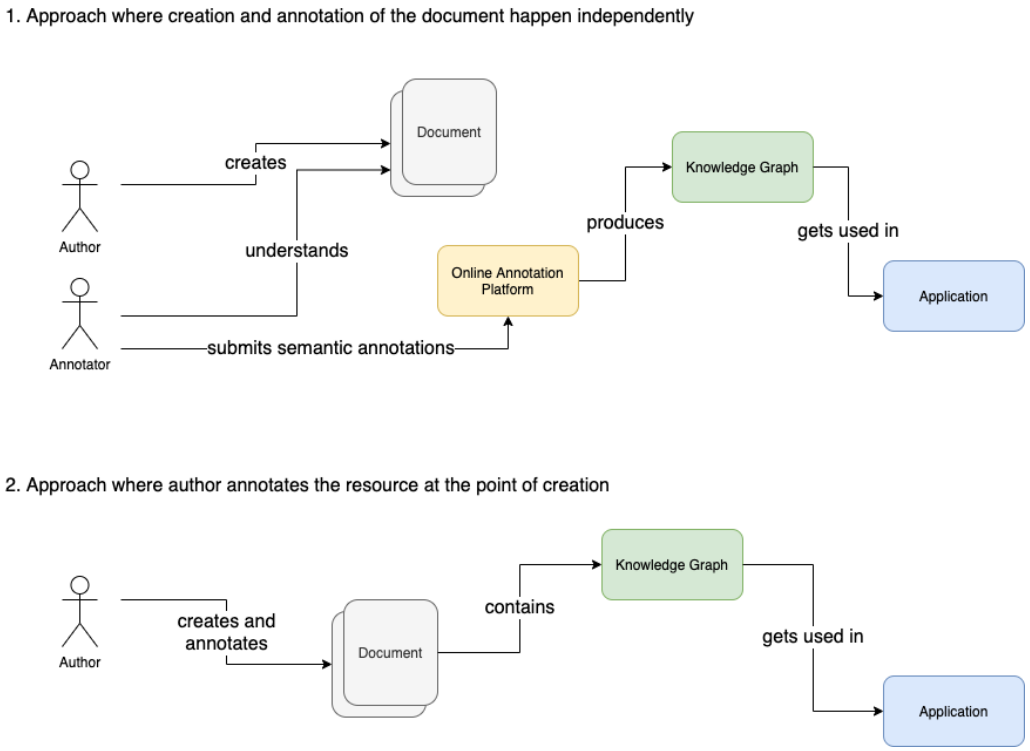

Figure 1.1: Different creation workflows of a knowledge graph of scholarly contributions

ing [19]. In the case of scientific knowledge graph creation crowdsourcing is realised through an annotation tool which is provided for example on a web platform. Then, the crowd i.e. anyone with access to the tool can submit annotations which are quality checked before they can be integrated into the resource. The first scenario in figure 1.1 shows the current process as it can be found on the crowd annotation platform of the ORKG. A paper gets created by the author and then an annotator adds the semantic annotations with the help of a web platform. The web platform then publishes the knowledge graph with an explorable and searchable interface. In the case of the ORKG the platform directly features an analysis application which accesses the resource, namely the paper comparison tool[3].

In this process it is possible that there is an arbitrary period of time between the creation of the document and the annotation of contributions. This makes it possible that documents which are older than the knowledge graph can be retroactively added to the graph either by the authors or any other person who read and understood the content. However, if the annotators are not the authors of the resources, they introduce some bias into the information they are annotating as they might not have the same understanding of the content as the authors. Additionally, the first approach is more complex and requires more distinct steps compared to

[3] `https://www.orkg.org/orkg/comparisons`

the second approach which is depicted in the lower part of figure 1.1. In this approach annotations are directly added during the production of the content at the time of document creation and the online annotation platform is omitted. Another advantage of the second approach is that the annotations and resulting knowledge graph can be directly attached to the metadata of the document when it is created and saved. In the first approach where creation and annotation are distinct events the knowledge graph is stored separately from the resources it is referring to. This makes the process dependent on the institution which hosts the knowledge graph data (i.e. the ORKG project in the ORKG example). In conclusion the first approach might be more flexible, but at the same time it is also more complex and harder to adopt than the second approach. Of course, both of these approaches can exist at the same time and give the users more choices for their preferred workflow.

In summary I motivate this work by the need for a new solution to annotate contributions in scientific publications and create semantic metadata. The following two research questions (RQs) will be underlying this work:

RQ 1: How can the process of manual semantic annotation of research contributions in scientific articles be simplified?

RQ 2: How do researchers use SciKGTeX to semantically annotate research contributions in scientific articles at the time of document creation?

The first research question is based at the projected simplifications of the process as described above and depicted in figure 1.1. The main outcome of this thesis will be a software solution to implement this simplification which is further specified in the next section. In RQ 2 'SciKGTeX' is the name of the software solution which is proposed to mitigate the problem sketched above. The question is concerned with the usability of said software solution and whether the basic task of identifying the main contributions of a scientific document for knowledge graphs is feasible during the production of the document. The answer to RQ 2 will be investigated in the usability evaluation in chapter 6 where the usability from the standpoint of the developed solution as well as the actual feasibility as a task is assessed. Both research questions are answered in chapter 7.

1.2 Goals

The overall goal of this thesis can be stated in a Goal Question Metric approach as defined by Basili et al. [20] which can be formulated as such: I **analyse** annotation of research contributions **for the purpose** of developing an approach to annotate research contributions at the time of document creation **with respect to** a convenient and easy solution to create semantically rich descriptions of research contributions **from the perspective of** this researcher **in the context of** researchers writing scientific publications.

To achieve this goal I partition the work into two sequential parts:

1. The technical implementation of a tool for semantic annotation of research contributions in scientific articles at the time of document creation (see approach 2 in figure 1.1).

2. An evaluation of the hypothesised usability and simplification benefits.

I further narrow the scope of my work to an environment where authors write their documents with the LaTeX type-setting system (see section 2.4). In the evaluation part, I will test the use case of a LaTeX package called SciKGTeX with real users and discuss the implications of the results. A user test evaluation will give insights into the usability of the package and assess the potential of the approach. In the end, I want to find out whether such a package is appreciated and convenient to use for authors of research publications which are the targeted user group.

1.3 Structure

The thesis is structured as follows. Chapter 2 establishes the minimal background knowledge to understand the presented work. It explains the connection to the topic of scientific knowledge graphs and gives the necessary concepts of the LaTeX language environment. Also the basic concepts of the Semantic Web and the Portable Document Format (PDF) are briefly touched upon. I discuss the related work which builds the base of this project in chapter 3 giving an overview on annotation software for LaTeX, ontologies and crowdsourcing for semantic annotation. In chapter 4, I explain the concepts and actual approach borrowed from the agile development method which was used to build the different parts of the project. These include user stories and the iterative development cycle. The implementation details of the LaTeX package and documentation of the functionalities are specified in chapter 5. Chapter 6 presents the overall approach of the user evaluation as well as the metrics used to measure the different variables. This chapter also details the results from the user test which are further discussed in chapter 7 with a comprehensive review of the strengths and weaknesses as revealed by the evaluation. This chapter also gives final answers to the research questions posed in section 1.1. The final conclusion and future work can be found in chapter 8.

2 Background

In this chapter I give a thorough introduction into the basic underlying concepts of the presented work. Beginning with section 2.1, I explain what is understood by FAIR data and data management. Then, in section 2.2, I introduce the concepts of knowledge graphs and the Semantic Web. Building on these concepts section 2.3 will introduce the scientific knowledge graphs such as the Open Research Knowledge Graph (ORKG) and other prominent attempts at the idea of semantically modeling research publications, which is also the goal of SciKGTeX. In section 2.4 I describe what LaTeX is and what it is used for. This will help to understand the use case which I introduce in chapter 4 and the implementation choices in chapter 5. In section 2.5 the PDF format will be further expanded on, as it is the format which finally holds the annotations in the XMP metadata field.

2.1 FAIR Principles

The FAIR principles are at the core of the design background for this work. FAIR is an acronym for **F**indability, **A**ccessibility, **I**nteroperability and **R**eusability in relation to scientific data and metadata. The principles were stated first in a 2016 paper by a number of scientists and organisations [21]. Their goal was to define a set of principles which can be followed to increase the quality of published data. While not suggesting any concrete implementation choices or standards the principles are intended to serve as a best practice checklist. They can be used to ensure that an implementation choice produces data which can be found, accessed, integrated and reused by human as well as machine actors. The data being machine actionable is a specific requirement of the FAIR principles, specifically the principle of interoperability. Wilkinson et al. [21] define this as follows.

Definition 2.1.1 (machine actionable)**.** The phrase *machine actionable* indicates a continuum of possible states wherein a digital object provides increasingly more detailed information to an autonomously-acting, computational data explorer. This information enables the agent /…/ to have the capacity, when faced with a digital object never encountered before, to: a) identify the type of object (with respect to both structure and intent), b) determine if it is useful within the context of the agent's current task by interrogating metadata and/or data elements, c) determine if it is usable, with respect to license, consent, or other accessibility or use constraints, and d) take appropriate action, in much the same manner that a human would.

The different FAIR principles may be adhered to in a best effort approach where a solution should strive to fulfil as many of the principles as possible but is not

Findability	
F1:	(meta)data are assigned a globally unique and persistent identifier.
F2:	data are described with rich metadata (defined by R1 below).
F3:	metadata clearly and explicitly include the identifier of the data it describes.
F4:	(meta)data are registered or indexed in a searchable resource.
Accessibility	
A1:	(meta)data are retrievable by their identifier using a standardized communications protocol.
A1.1:	the protocol is open, free, and universally implementable.
A1.2:	the protocol allows for an authentication and authorization procedure, where necessary.
A2:	metadata are accessible, even when the data are no longer available.
Interoperability	
I1:	(meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2:	(meta)data use vocabularies that follow FAIR principles.
I3:	(meta)data include qualified references to other (meta)data.
Reusability	
R1:	meta(data) are richly described with a plurality of accurate and relevant attributes.
R1.1:	(meta)data are released with a clear and accessible data usage license.
R1.2:	(meta)data are associated with detailed provenance.
R1.3:	(meta)data meet domain-relevant community standards.

Table 2.1: The FAIR principles

necessarily invalid if it does not accomplish all of them. FAIR principles are already widely accepted as a standard and respected by many data repository initiatives [22, 23, 24]. One technology which is very suitable for implementing the FAIR guidelines are knowledge graphs. They are specifically designed to support machine actionable data and can be used to realise the principles of interoperability and reusability. How knowledge graphs work is explained in detail in the following section.

2.2 Knowledge Graphs

The *Semantic Web* is a term coined by Sir Tim Berners Lee [25] to describe a version of the internet which has a meaningful, common and standardised vocabulary to denote real world objects or concepts and the relations between them. It should enable autonomous agents, like computer programs to understand the network of data much better by resolving ambiguities of natural language and providing a graph structure for information encoding.

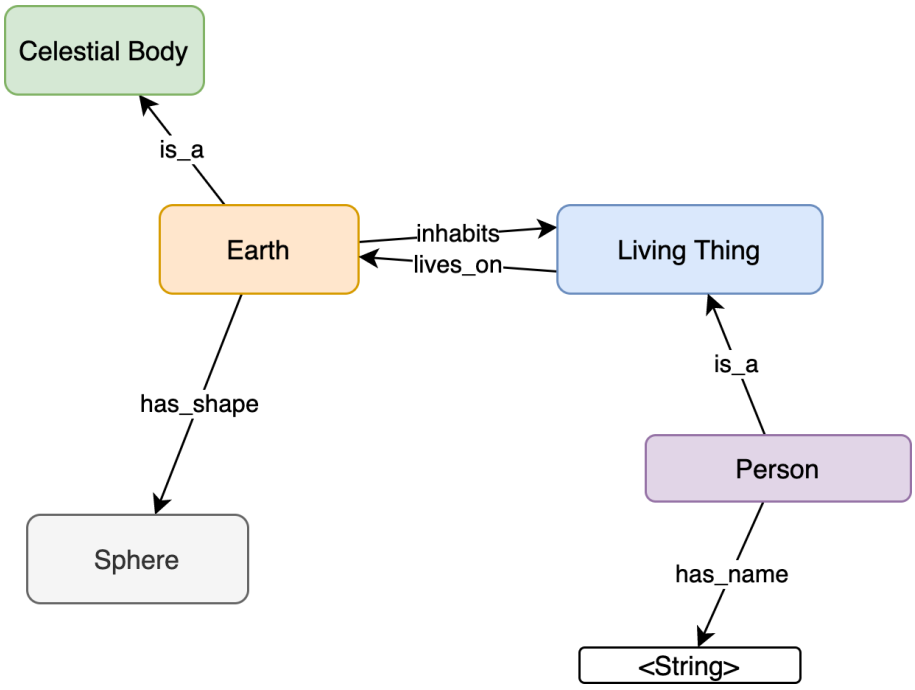

Figure 2.1: A simple knowledge graph, describing a small portion of world knowledge. The rounded rectangles contain concepts and labeled edges represent the relations between them.

2.2.1 Ontologies

Ontologies build the foundational model for the Semantic Web. Guarino et al. [26] define them as follows.

Definition 2.2.1 (Ontology)**.** An ontology is a formal, explicit specification of a shared conceptualization.

Ontologies are used to formally define the common vocabulary of objects, concepts and relations surrounding certain topics. The vocabulary is further turned into a conceptualisation by modeling the semantics of a real world domain through a network of terms and by specifying how the concepts denoted by these terms relate in different ways. This might include relations like two concepts being part of another, being similar to another, or any describable relation that the two concepts can have. For example, a machine can understand the concept of earth as a type of celestial body which has the shape of a sphere and further inhabits living things. What is needed is a specification of this sentence in terms of the central concepts and how they relate to each other (see figure 2.1). The network of concepts and relations gives a much deeper and more flexible understanding than the natural language sentence.

The connections between the concepts of earth, living thing, person, etc. are called the relations of these concepts to each other. Relations can have properties like transitivity or symmetry which allow inference operations on a knowledge graph.

In the figure, the type relation (`is_a`) implies through its transitivity property that the earth as an object inherits all the properties and relations from the *Celestial Body* concept. The `is_a` relation allows to infer new knowledge by exploiting this transitivity. In our example (figure 2.1) we can infer from the fact that a person is a type of *Living Thing* that therefore a person lives on earth because *Person* inherits the relations applied to *Living Thing*. Furthermore, the knowledge graph can be easily extended by adding more concepts (for example *Animal*) and relating them to the existing concepts.

The advantage of such interrelated conceptualisations over natural language is that we can greatly reduce ambiguities which arise for example from using different number strings for the same concept e.g.: *2011-08-01* or *first of August, 2011* or even just *tomorrow*. Apart from resolving ambiguities machines can get a limited portion of world knowledge through networks of concepts and relations between them, conceivably similar to the mental images in the human mind.

2.2.2 Data Specification

In the Semantic Web we usually conceive objects or concepts as nodes and relations as edges in a graph called a *knowledge graph* [27]. A knowledge graph contains an accumulation of knowledge about a certain topic or domain. The graph can be expressed in RDF (Resource Description Framework) [28] which is a language to specify directed graphs as triples of *subject*, *predicate* and *object* where subjects and objects can be arbitrary objects or concepts and the predicates denote the relation between them. For example the fact that the concept of earth and the concept of the sphere are connected by the relation of `has_shape` can be modeled by the RDF snippet in listing 2.1.

Listing 2.1: RDF snippet modeling a simple fact.

```
@prefix ex:http://example_uri.net
ex:Earth ex:has_shape ex:Sphere
```

RDF graphs may be specified in multiple different syntax formats such as RDF/XML, TURTLE[1] or JSON-LD[2]. An example of the TURTLE syntax is shown in listing 2.1.

Defining new concepts will inevitably introduce inconsistencies with other conceptualisations from different domains or different authors. For example, a newly created knowledge graph might give earth the shape of a geoid instead of a sphere. This might be a more accurate description depending on the context of the domain that the graph was produced for. This example shows how there can exist knowledge graphs with different, sometimes even conflicting world views. A knowledge graph only ever models a domain-specific representation of concepts and can never claim completeness as it models the real world which is always growing and can never be

[1] `https://www.w3.org/TR/2014/REC-turtle-20140225/`

[2] `https://www.w3.org/TR/json-ld/`

described exhaustively. Technically domains are implemented with the help of so called *namespaces* which ensure the uniqueness of a conceptualisation by attaching its domain to it. In listing 2.1 the first line specifies a namespace which is then used to mark concepts belonging to the domain.

2.3 Scientific Knowledge Graphs

Scientific knowledge graphs are large collections of facts representing scientific information such as metadata about people, documents, datasets, institutions, grants, and also semantic descriptions of research contributions [29]. Auer et al. [5] reviewed existing scientific knowledge graphs and found that many existing solutions were aiming at either organisation of bibliographic metadata or fully automated knowledge extraction. In this section I will introduce the most notable solutions which have components to represent research contributions in scientific knowledge graphs in chronological order.

ScholOnto One of the first notable efforts at the attempt to construct a scientific knowledge graph is described by Buckingham-Shum et al. [30] which 20 years ago developed an infrastructure to represent archived documents with a semantic network of concepts and discourse called *ScholOnto*, the Scholarly Ontology. The implementation came in the form of a java applet server to create and visualise such networks. Uren et al. [31] developed a variety of prototype tools in the *ScholOnto* project. The goal of these tools was to facilitate a digitalised scientific discourse through the use of *Claim Networks* which represent scientific contributions and allow computer assisted sense making.

ScienceWISE Aberer et al. [32] developed the platform ScienceWISE[3] which features a community-created encyclopedia of articles (mostly about physics) with an ontological structure growing alongside it. Additionally the platform displays connections from ontology entries to a collection of papers and provides an annotation platform where scientists can add papers, link them and update the ontology. While being similar in concept to the ORKG, ScienceWISE does not create a knowledge graph of specific content contributions of publications. Meanwhile, the web applications are fairly outdated already.

ClaimsKG Tchechmedjiev et al. [33] present the ClaimsKG, a knowledge graph of claims which were automatically crawled from fact-checking sites. The idea of this graph is similar to that of the ORKG with the claims being the equivalent to contributions. However, these claims comprise mainly political statements and they are not limited to scientific publishing. Also, the resource is built fully automatically with no manual curation.

[3] `http://ScienceWISE.info`

Publishing Organisations Scientific knowledge graphs are also adopted by some publishing organisations such as Elsevier's Research Knowledge Graph[4] or Springer with the Springer Nature SciGraph [34]. These graphs are however mainly concerned with the publishing activities of the companies and not open-source, collaborative platforms.

ORKG With all the presented solutions, an open-source knowledge base of research contributions was still not realised on a large scale. This led to the conception of the Open Research Knowledge Graph (ORKG) [35] which is a knowledge graph of research contributions which are found in scientific publications across many disciplines. The project aims to bring the most essential findings of research publications into a semantic format to construct a machine-actionable resource. The possible applications of such a resource are manifold as mentioned in chapter 1.

Additional to the knowledge graph data itself, the ORKG project hosts a web application[5] for users to add the contributions of research papers to the knowledge graph. The web application also features search and inspection tools as well as a user interface to create systematic literature comparisons (see figure 2.2) based on the contribution triples [36].

In the ORKG, contributions are stored in a knowledge graph using a multitude of custom classes and relations. However there exists a core ontology[6] with the most essential classes and relations to define the graph. Arguably the most important notions are those of the *ResearchContribution* and the *Paper*. An instance of *Paper* can have a number of *ResearchContributions* assigned to it. *ResearchContributions* consist of subject, predicate and object, where the subject might be an instance of class Paper, the predicate could then be a relation like *addresses* and the object an instance of class *ResearchProblem*. However, this triple just describes one property of the *ResearchContribution*. Many more predicates and objects can be defined and assigned to the *ResearchContribution*. These predicates and objects are not predefined and can be specified by the creator.

The creation of contributions for the ORKG is mainly done via crowd-sourcing. Knowledgeable individuals annotate the triples for the ORKG via a web interface. Automatic extraction techniques to populate the knowledge graph are also used to some extent but are still in development phase [37]. Since the release of the ORKG web interface in the beginning of 2020, over 300 contributors added research contributions from publications to the knowledge graph. As of September 2021 the ORKG counts 9713 contributions from 6926 papers.

The ORKG platform features a form based process where the users can manually add information about a paper, about the research field and the explicit contributions from the paper. Apart from data entry users also have the possibility to

[4] `https://www.elsevier.com/connect/how%2Dai%2Dand%2Dknowledge%2Dgraphs%2Dcan%2Dmake%2Dyour%2Dresearch%2Deasier` - May, 2022

[5] `https://www.orkg.org/orkg/`

[6] `https://github.com/ORKG/orkg-core-ontology`

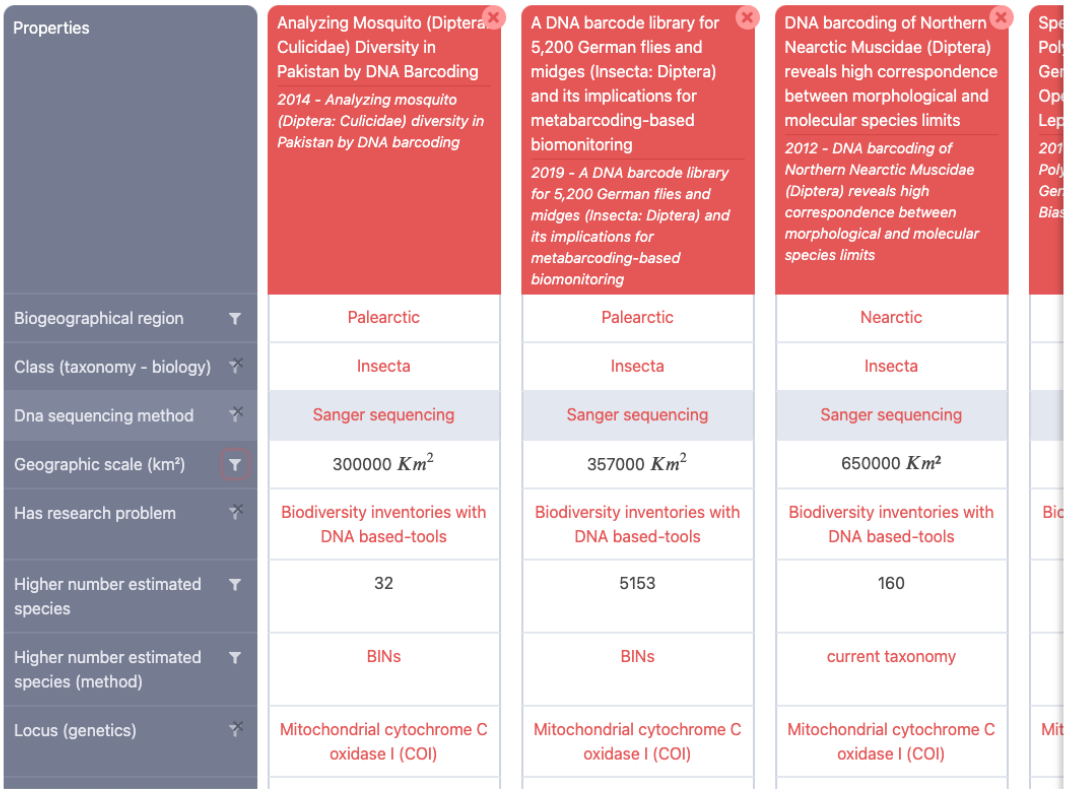

Figure 2.2: Example of a research contribution comparison on the ORKG website's user interface.

generate comparisons between research contributions on the ORKG web platform (see for example figure 2.2). The comparisons can be created by selecting contributions in the knowledge graph and choosing the properties on which they should be compared [36]. This allows to generate tables which review for example the results, methods or research material of certain publications and contrast them against each other. Paper comparisons are one example of the many possible applications of a scientific knowledge graph like the ORKG.

Nanopublications Another interesting approach are so-called *nanopublications* [38, 39]. Nanopublications are knowledge graph representations of elements of research communication such as articles or reviews. They are designed to be issued by researchers as an alternative to classical publications and do not act only as additional annotations to existing publications like most of the other approaches presented before. Similar to the contributions in the ORKG, nanopublications use certain properties to characterise scientific claims. Different from the ORKG approach, they use exactly these 5 properties to describe a scientific claim: context class, subject class, qualifier, relation type and object class. The objects of the qualifier and relation type properties must be chosen from a predefined list and the others must be classes from a linked open data knowledge graph. Nanopublications are very restrictive in

the patterns which they allow, which makes them less flexible but more consistent and exploitable for reasoning applications as demonstrated in [39].

2.4 LaTeX

LaTeX is an open-source type-setting system which can be used to create different kinds of documents in a consistent style. It is based on the TeX type-setting system developed by Donald Knuth [40]. LaTeX is essentially a collection of macros to simplify the usage of TeX to quickly build common document types. A typical LaTeX source file contains declarative statements (commands) alongside the raw text content to mark special elements of the text such as the titles, sections, tables, images, etc. On top of that there are predefined document types which are specified through the use of document classes in LaTeX code. The document classes contain the necessary TeX code to structure and style the documents with just a minimal set of commands (for example LaTeX code see figure 5.1). Commands can be very powerful and set up a complete bibliography of references for example with just one short expression. The 5 main document classes to choose from are *article, report, book, slides* and *letter* with the possibility to derive new custom classes from them.

Documents specified in LaTeX can be compiled to device-independent presentation formats such as DVI or PDF. By choosing predefined document classes the focus is shifted away from styling and formatting documents – which is handled by LaTeX – and more towards the actual content. Using LaTeX also eliminates the need for complicated graphical user interfaces for the production of documents. Once the basics are mastered, LaTeX allows to save lots of time which would otherwise be spent on repetitive tasks such as formatting bibliographies, positioning images or setting styles.

It is due to the time savings and online collaboration platforms such as Overleaf[7] that LaTeX has become one of the most used tools to create scientific publications. Overleaf is an online LaTeX editor which provides in-cloud compilation, collaborative editing and document management as its distinctive features. Overleaf massively facilitates the use of LaTeX by making the technically demanding installation obsolete and making it easier to share large documents. Overleaf reports over 8 million users (Oct. 2021[8]) of their web interface. This number gives an indication of the size of LaTeXs user base while still excluding all the users who write LaTeX in other editors than the Overleaf platform. LaTeX is one of the most used tools to produce scientific publications and the de-facto standard in STEM areas of academia.

2.4.1 LaTeX Packages

Working with LaTeX it is possible to define new commands and environments to facilitate the reuse of large code fragments. To reuse code it is helpful to separate

[7] `https://www.overleaf.com`

[8] `https://www.overleaf.com/about`

the reusable portions i.e. the command or environment definitions from the rest of the source material, so the definitions can be imported by other projects, without having to rewrite the code again. There are two different types of external LaTeX dependencies: document classes and packages. A document class applies only to a specific class of documents and can be an extension of the main document classes. A package on the other hand contains more general code which can be used for all kinds of documents [41]. Packages for LaTeX are distributed online through CTAN[9] (The Comprehensive TeX Archive Network) which contains thousands of packages for different purposes.

2.4.2 LuaTeX

As explained before, LaTeX defines a set of convenient macros for TeX. The TeX engine then compiles the LaTeX code to printable output. The original TeX engine only produced DVI output and other engines like pdfTeX, XeTeX or LuaTeX have been developed to compile PDF output directly. As a successor to pdfTeX, LuaTeX additionally features the embedded scripting language Lua and native Unicode support [42]. The multi-paradigm programming language Lua[10] incorporated into LuaTeX provides a very convenient bridge to parts of the TeX engine which were previously not accessible. LuaTeX offers helpful interfaces to the inner workings of TeX in the form of callback hooks. Since TeX itself is a language with many peculiar idiosyncrasies, which make it hard to understand and modify it, the Lua interface makes the engine much more flexible as developers can alter its behaviour without having to study TeX in great detail. For packages involved with low-level functionalities like PDF creation LuaTeX is a good way to implement such features evading the cumbersome task of partly rewriting the TeX engine.

2.5 Portable Document Format

The Portable Document Format (PDF) is a file format developed by the software company Adobe in 1993 [43]. It is designed to display documents including textual content and images consistently across all devices and operating systems. PDF is based on the PostScript language [44] which can be used to describe contents of a printed page on a higher level than a raster graphic representation. A PDF file contains a complete description of the document layout including text, fonts as well as positions of embedded images and vector graphics.

In 2008 PDF was first standardised in ISO 32000 [45] and since then it has become the de-facto standard for sharing documents on the web. The standard has been updated several times with the latest edition being released in December 2020. Alongside the ISO 32000 standard, more specialised standards such as the PDF/A for archiving or PDF/X as a special exchange format have been established. These

[9] `https://ctan.org`

[10] `https://www.lua.org/`

standards are subsets of the original PDF standard and impose restrictions on the PDF definition in order to make the documents compatible with their respective use case. The PDF/A standard is gaining traction in the world of publishing as it requires embedded fonts and embedded color profiles for consistent replication of the document as well as embedded metadata to provide information on the archived document.

Extensible Metadata Platform Metadata in PDF is data that is neither concerned with the content nor the structure of the document itself. It contains information like the title, author or modification dates of the document. Beginning with PDF 1.4, metadata can be stored either in metadata streams which contain XML data or in a *document information dictionary*. While the document information dictionary as a key-value store is rather restricted in terms of content flexibility, metadata streams can contain arbitrary XML constructs. For metadata streams Adobe recommends the eXtensible Metadata Platform (XMP) standard [46]. XMP has been developed as a general-purpose metadata specification standard. It can be used to add metadata to almost all of the common media data types such as PDF, JPEG, MOV, MP3 or SVG. XMP relies on serialised XML and specifically RDF to specify metadata properties of resources. Listing 2.2 shows how a typical metadata stream can be encoded in PDF. As an example the same information from the RDF turtle syntax (see listing 2.1) is embedded.

Listing 2.2: An uncompressed metadata stream as it can be found in a PDF file.

```
1152 0 obj
<< /Type /Metadata /Subtype /XML /Length 1706 >>
stream<?xpacket begin='' id='W5M0MpCehiHzreSzNTczkc9d'?>
  <x:xmpmeta xmlns:x="adobe:ns:meta/">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-...">
      <rdf:Description rdf:about="http://example_uri.net#Earth"
       xmlns:ex="http://example_uri.net">
        <ex:has_shape>
          <ex:Sphere/>
        </ex:has_shape>
      </rdf:Description>
    </rdf:RDF>
  </x:xmpmeta>
<?xpacket end='w'?>
endstream
endobj
```

3 Related Work

This work has several connection points with research that has been done in the past. In this chapter I list the major developments that have been brought up in related research problems. I also show where existing solutions differ from the one presented in this thesis and justify the approach compared to the state of the art. The first category of related works are the previous attempts at annotation in LaTeX which I give an overview on in section 3.1. The annotation scheme for the LaTeX package is largely based on the ORKG core ontology which relates the different concepts surrounding research contributions. I take a look at existing ontologies in the field of scientific documentation in section 3.2, compare the different vocabularies which are introduced and discuss the reuse of existing ontologies for the project. Finally, section 3.3 gives an overview of crowd sourcing concepts and lists some crowdsourced tasks in semantic annotation.

3.1 Annotation in LaTeX

Annotating document elements directly in a LaTeX writing environment has the advantage that annotation can happen concurrently while creating the document content. It eliminates the need for a posteriori editing of documents and encourages that authors themselves perform the necessary annotations. In 2007, Groza et al. [47] published a framework to semantically annotate structural and content-related text elements in LaTeX. The framework is called SALT (Semantically Annotated LaTeX) and comprises a LaTeX package with annotation commands and an annotation schema consisting of three ontologies. Similar to the approach chosen in this work, the annotations are stored in the PDF metadata field, whereas the use case is different. They concentrate primarily on generation of HTML content from the annotated PDF to support automatic creation of online proceedings but do not explore other use cases.

In SALT many annotation possibilities are provided through the LaTeX commands which facilitate the markup of different elements in the text like structural elements, rhetorical elements and general metadata. For the creation of a scientific knowledge graph of contributions like the ORKG many of the structural annotations are not relevant. Only the rhetorical elements such as contributions or claims and the metadata annotations are useful to form triples in the knowledge graph. Moreover, SALT as a software is not maintained anymore and can not be used at the time of writing this thesis. The SALT approach can be adapted in some parts for the thesis solution but overall the focus of SALT is quite different from the use case presented here,

where the focus is mainly on extracting research contributions for knowledge graph construction.

Semantic annotation through custom LaTeX commands has also been proposed by Krieg and Brückner in 2006 [48]. They present commands and an ontology to annotate semantic terms in documents, more specifically on the example of lecture slides. The idea is to semantically interrelate terms between documents through an ontology. Among the use cases they describe are a fine-grained version control system, ontology-based search and resolution of ambiguities. In contrast, the focus of this thesis' work is more on specific content contribution and not on document structure and term hierarchies.

Moreau et al. [49] released a LaTeX package which can be used to add provenance information to a document. As provenance they define a record that describes how entities, activities and agents have influenced a piece of data. For example it creates unique identifiers (URIs) for different components of the document and links the agents and organisations to websites or ORCIDs[1]. Their package generates RDF statements for different types of provenance and saves them in a TURTLE file. It also adds a link to the TURTLE file into the XMP metadata field of the PDF file. They claim that it was not possible for them to add the information directly into the XMP field. Contrary to this statement, the package developed in this thesis actually makes it possible to embed any RDF data into the XMP field, so future releases of the package might feature some mechanism to add provenance information along with the contribution annotations.

Another semantic annotation markup was developed by Michael Kohlhase [50] to turn LaTeX into a document format for mathematical knowledge management (MKM). He uses a collection of macros called sTeX[2] which define many new commands to specify mathematical statements and semantically describe whole theorems and proofs building on that. The range of possibilities of sTeX are impressive and while it does not feature mechanisms to explicitly state research contributions in a way that would be fitting for knowledge graph creation, its methodology is an inspiration for mathematical annotation templates which might be added to the package presented in this thesis in the future.

Furthermore, there exist several LaTeX packages to add XMP metadata to PDF files like minim-xmp[3], xmpincl[4] or hyperxmp[5]. These are rather generic packages which either only allow to embed bibliographic metadata or are too complicated to use for users without the knowledge of XMP technology. A solution to embed research contributions as RDF triples in XMP metadata does not exist to date.

[1] Open Researcher and Contributor IDs: `https://info.orcid.org/what-is-orcid/`

[2] `https://ctan.org/pkg/stex`

[3] `https://ctan.org/pkg/minim-xmp`

[4] `https://ctan.org/pkg/xmpincl`

[5] `https://ctan.org/pkg/hyperxmp`

3.2 Ontologies

Literature review articles like [51] and [52] from 2014 and 2020 respectively, review different ontologies for semantically describing scholarly and scientific documents. Ruiz-Iniesta et al. [51] classify the ontologies into three groups. Ontologies for describing the document structure (sections, paragraphs, etc.), ontologies for describing the rhetorical elements (introduction, results, etc.) and ontologies for describing bibliographies and citations. Since the ORKG deals with descriptions of concrete research contributions, I will focus on ontologies describing rhetorical elements, especially if they provide means to capture content-related semantics. Nguyen et al. [52] later compiled a list of 34 ontologies and investigated their role in the creation of a hypothetical research knowledge graph. They mapped the concepts and relations found in these ontologies to different areas of research information gathering which have the potential to be represented by a knowledge graph[6]. They found that in some areas there are overlaps between the topics that can be described by certain knowledge graphs (e.g. publications and organisations). Other areas stayed relatively untouched. For example there were no ontologies developed yet to model topics like *funding programs* or *start-ups*. Another comprehensive comparison of different scholarly ontologies can be found as a comparison on the ORKG page[7].

One of the most comprehensive ontologies to model documents and their content is the Document Components Ontology (DoCO) [53]. Despite featuring a number of different classes for structural and rhetorical elements of the text, DoCO does not have a class for a contribution per se. The subset of the ontology specifying rhetorical elements – the Discourse Elements Ontology (DEO) – does however have some general content-related classes like *background*, *methods* and *results* which can be used to describe some contributions for the ORKG.

Another ontology to describe rhetorical elements of scholarly documents is CISP (Core Information about Scientific Papers) [54] which uses classes from EXPO [55] which is a detailed ontology of experiments. EXPO has over 200 defined concepts and provides a very fine-grained vocabulary to describe scientific experiments.

The Argument Model Ontology (AMO)[8] provides classes and relations to model argument discourse. Similar to that, CoreSC [56] defines a set of core scientific concepts to automate access to the scientific discourse. CoreSC was developed in the area of biomedical publications where also the SWAN (Semantic Web Applications in Neuromedicine) [57] ontology originates from. SWAN comprises classes for people, discourse elements, bibliographic records, life science entities, vocabularies and versioning. The discourse elements are grouped into the classes *research statement*, *research question* and *structured comment* with a number of relations to link instances of these together. In table 3.1 I summarise the key classes and relations introduced by each of these ontologies.

[6] `https://github.com/nvbach91/iga-knerd/tree/master/coverage`

[7] `https://www.orkg.org/orkg/comparison/R8342`

[8] `https://sparontologies.github.io/amo/current/amo.html`

	Classes	Relations/Properties
DoCO/DEO	Background Contribution Methods ProblemStatement Results Conclusion Model Data Reference ...	-
CISP	Goal of investigation Motivation Object of investigation Research method Experiment Observation Result Conclusion	-
EXPO	ScientificExperiment ExperimentGoal ExperimentTechnology ExperiementResult ...	-
AMO	Argument Claim Evidence Warrant Backing Qualifier Rebuttal	backs forces proves involves leads_to supports hasEvidence hasClaim ...
SWAN	Research statement Research question Structured comment	alternativeTo discusses inconsistentWith consistentWith contains citesAsEvidence ...

CoreSC	Background Method_old Hypothesis Goal Object_new Motivation Model Experiment Method_new Conclusion Result Observation	-

Table 3.1: Ontologies to describe rhetorical elements in scholarly texts and some of their key classes and relations

For the design of the LaTeX package I used the findings of the literature research for the development of the annotation scheme. The annotation scheme mainly relied on the classes from the DEO which is further specified in section 5.2 on implementation.

3.3 Crowdsourced Semantic Annotation

Crowdsourcing is a problem-solving method where a group of human workers collaborates on task solution. Crowdsourcing has become increasingly popular in many fields of research due to its speed and relative cost-efficiency. Tasks usually fall into one of the following four categories established by Assis Neto et al. [58] in their review article on crowdsourcing: Object production, object for solution, object processing and object evaluation. A LaTeX package to semantically annotate contributions of research publications falls into the object production category, meaning that the crowdworker will produce new objects (e.g. annotations) by solving the task. Just like in the case of this thesis' work the majority of object production projects reviewed by Assis Neto et al. used a specific crowd as opposed to an anonymous crowd. This means that the annotation is made by a specific group of people (e.g. authors of research papers) which can also be identified as opposed to random people from the internet for example.

Another aspect of crowdsourcing are the incentives which convince a worker to invest time into a task. Hosseini et al. [59] differentiate between intrinsic (entertainment, altruism, etc.) and extrinsic (money, access, etc.) incentives. The ORKG project uses different types of both intrinsic and extrinsic incentives such as the personal incentive to increase the accessibility of one's work or also monetary compensation to maximise the user base of their crowdsourcing platform [60]. The LaTeX package has similar incentives, since the annotations can be uploaded to the ORKG, but may have a better potential for intrinsic motivation since the annotations enrich

the document itself which is owned by the author and annotator.

Apart from the ORKG, there are a number of initiatives which used crowd sourcing to produce semantic annotation for various applications. Svatek et al. [61] discuss the idea of community tagging for scientific knowledge graphs and experiment with relation annotation for 5 scholarly articles. Takis et al. [19] present an approach to specify semantic annotation for PDF documents. Their *SemAnn* approach lets users annotate PDF documents in a web application which provides annotation suggestions. To incentivise participation in the annotation effort the system recommends similar papers based on the annotations that have been done. The tool also comes with the possibility to extract tables from PDF which is an interesting feature.

A similar PDF annotation tool exists for the ORKG as well and is described by Oelen et al. [62]. While this approach seems viable for already published papers, I would argue that adding the annotations directly in LaTeX instead of first converting to PDF is a better approach since it is simpler and less error-prone. For example, tables are certainly easier extracted from LaTeX source than from a PDF file.

4 Approach

This chapter introduces the main ideas behind the developed solution to annotate contributions in scientific publications at the time of document creation. I define the requirements and functionalities of the sought-after solution and introduce the methods which are used to develop it. My approach borrows elements from the agile software development methodology [63], specifically the iterative cycles of development and user story specification. The chapter is structured into the following sections. Section 4.1 presents a possible scenario to demonstrate a use case and illustrate the various requirements to the solution. Based on the use case scenario specific user stories are listed and explained in section 4.2. Then, I derive the necessary functionalities for the solution in section 4.3 with references to the user stories. In section 4.4 I further illustrate the agile software development process employed to build the solution.

4.1 Use Case Scenario

The use case scenario is a fictional story which is fabricated to illustrate different requirements for the software solution which is presented in this work. Even though the scenario is fictional, it should be noted that it is based on the true problems and needs of the different user groups which have been established in conversational exchange. The story connects the requirements in a realistic way, so that the context and use case of the developed solution become clear. The following paragraph describes the use case scenario.

Scenario 1. A team of medical researchers conducts a study on the effectiveness of a certain drug treatment. After conducting the trial, the researchers want to publish their results to a wide audience. They create a document to submit to the publisher of a renowned journal. The researchers already know from past publications that individual papers can get lost under the enormous amounts of published material especially if it does not end up in literature survey papers or other sorts of meta-reviews. Consequently, they want to make sure that they prepare the paper optimally to maximise their impact. For them it is important that their findings reach the right audience and can be found easily by other researchers working on a related topic. They also want to ensure reproducibility and comparability of their work.

From colleagues they know that there exists a possibility to make their paper stand out by enriching the document metadata. In the metadata they can add useful information on their document which can be used by search engines to find

it or recommend it to other users. They find out that an easy way to add such information to the document metadata is using a certain annotation software. The researchers download the software and study the documentation.

After a short introduction, they start to annotate metadata in their project. The software allows them to mark the most important parts of their paper such as the background, research problem, methods and the results directly in the text of their publication document. Through the use of the annotation tool, they learn about the Open Research Knowledge Graph (ORKG). They find that their research problem already exists as an entity in the ORKG. So instead of just giving a natural language string as a research problem, they add the link to the research problem in the ORKG. They also realise that the antibiotic they are using can be found in a knowledge graph of drugs. The knowledge graph yields a unique identifier (URI) which they can embed into the document so that automatic systems can find it without relying on a specific name (which can change over time) to identify the mention of the drug.

All these steps make the document more complete and give the researchers confidence that they crafted their document in the best possible way. When they finally submit the paper, they make sure to upload it to the ORKG web platform as well to increase their reach. They upload it on the web platform via the paper upload feature which extracts the metadata from the PDF and saves it in the knowledge graph. By uploading them to the ORKG the authors allow other researchers to include their findings in paper comparisons created on the ORKG platform.

Months later the paper is incorporated into a comparison of antibiotic treatments which is part of a meta-review of many similar studies. As the researchers have clearly marked their results, it is easy for the review author to find them and compare them with other approaches. It turns out that their results are very helpful in the development of new treatment plans. Consequently, their study is widely cited and referenced in many more articles.

The use case scenario ends here on a very positive note which highlights the potential of the metadata annotation. In the best case the metadata contributes to the discovery of the research content and helps the readers to better understand and compare it. Even in the worst case, if the metadata is not used, it is still attached to the document for the duration of its existence which makes it future proof for downstream processing of the document content. Through the course of the story different aspects of the sought-after solution are illuminated. The following section lists the specific requirements which are addressed in the use case scenario in the form of user stories.

4.2 User Stories

To justify the functionalities of the developed solution I identify the requirements that users have for the annotation tool from the presented use case scenario. The requirements are illustrated with a number of user stories [64] following the *Connex-*

tra template [65]. User stories are a common practice of agile software development [63]. In the *Connextra* template a user story has 3 slots. A role, a requirement and a reason. The slots are connected into a sentence:

> As a ⟨*role*⟩ I want to ⟨*requirement*⟩, so that ⟨*reason*⟩.

The different user stories that I present are organised into three roles which were identified. The researcher, the publication provider and the literature review author.

4.2.1 Researcher

The notion of researcher specifically stands for an author of a scientific publication here. Of course writing is not necessarily a task of every researcher, but it is considered an essential part of scientific work since publications mostly form a researcher's record of achievement. Communicating research results is central to the research profession and the following user stories demonstrate the requirements a researcher might have for a tool to annotate scientific contributions.

User Story 1: As a researcher I want to find related work to my idea, so that I do not produce redundant work.

User Story 2: As a researcher I want to increase the findability of my work, so that my work gets noticed.

User Story 3: As a researcher I want to embed my research into a scientific ecosystem, so that my work can be discovered.

User Story 4: As a researcher I want to refer to existing research objects in the web like results, problems or definitions, so that my reasoning can be completely retraced.

User Story 5: As a researcher I want to highlight my contributions, so that my work stands out.

A scientific ecosystem in user story 3 would be the platforms and tooling which are used in a researcher's typical workflow such as paper repositories or comparison platforms. User story 4 aims at the frequently mentioned 'reproducibility crisis' [4] according to which published research results are often hard to reproduce by peer researchers. There are many possible reasons for this and a potential alleviation of the problem can be achieved through formal descriptions of methods and approaches in metadata of publications. Also, digital experiment artifacts such as code or videos can be linked in the metadata. Similar to the researchers also the distributors of publications profit from rich metadata in their documents. I will further specify their requirements in the next subsection.

4.2.2 Publication Provider

Here a publication provider denotes an entity or organisation which collects scientific publications and distributes them to the greater public. This includes publishers of academic journals, conference proceedings and books as well as library services and archive platforms such as arXiv.org. Even though these different types of providers might have vastly different goals or principles, here I focus on the requirements they would have in common for annotation of contributions in scientific documents.

User Story 6: As a publication provider I want to aggregate metadata on document content, so that I can build better search and recommendation tools for my users.

User Story 7: As a publication provider I want documents to conform to the PDF/A standard, so that the files are safe to store long-term.

User Story 8: As a publication provider I want to publish data according to the FAIR principles.

The distributors of publications have an interest in the metadata on the document content because they form a valuable addition to the information they extract automatically from the text such as topics or citations. Libraries for example build a structured catalogue of their inventory with any available metadata. Any information which is already provided by the authors in a structured, machine-readable way does not have to be extracted manually or through resource-intensive computations.

As the providers must work with the actual document files, they care about technical implementation standards which are followed. One of them is the PDF/A standard (see section 2.5) referred to in user story 7. While not many publishers require the standard nowadays, it might become important in the future and the metadata should be at least compatible with this standard. Similarly in user story 8 the FAIR principles (see section 2.1) should be incorporated into the metadata since many data repositories are already implementing them and are interested in data which adheres to these principles. Other potential beneficiaries of FAIR metadata are authors of literature reviews which are covered in the next section.

4.2.3 Literature Review Author

Semi-automatic generation of literature review articles is one of the possible downstream applications which result from a scaled adoption of content-related metadata annotation. If contributions are annotated in a document's metadata, they can be extracted automatically and compared with each other in little time. Literature review authors are authors of a review paper who intend to use automated comparison platforms to supplement the creation of their review.

User Story 9: As a review author I want the most important results of scientific publications available online in a structured way, so that I can efficiently build comparison tables between different papers.

Literature review authors can profit from authors' annotations when comparing the contributions of different relevant studies which deal with their research problem. For example when they set up a table of different papers to compare specific results.

4.3 Functionalities

From the use case scenario and the individual user stories, a number of functionalities or components can be derived which the sought-after solution should incorporate. For the most part, there are multiple different possibilities to implement these functionalities in real-world software. The actual implementation which is chosen for this project is explained in chapter 5. Here I list the functionalities and specify how they emerge from the user stories defined in section 4.2.

Functionality 1: Annotation of textual entities in documents

This functionality is the central element of the solution and lays the foundation for the user stories 1–4. Especially user story 5 calls for a functionality to mark text passages which describe the contributions of a publication.

Functionality 2: Assignment of properties to textual entities

For user story 3 and 5 it is helpful that textual entities can be given different labels. Instead of just marking contributions the users should be able to distinguish different kinds of contributions.

Functionality 3: Linking of URIs[1] to textual entities

User story 4 demands a functionality to link the annotated textual entities to corresponding objects on the Semantic Web. For example users should be able to mark a specific specific research problem by providing a unique link to an entity on the web which defines it. Also, real world objects like certain proteins, stars or plant species should be denoted with URI's which identify the object unambiguously.

Functionality 4: Permanently storing annotation in document metadata

The produced metadata and the document they refer to should be stored in the same entity i.e. the metadata should be attached to the document to ensure long-term portability. This is essential for the solution as a whole and specifically required for user story 7 where publishers distribute the documents maybe even unaware of the metadata which are attributed to them.

Functionality 5: Defining custom properties for annotation

This functionality is an extension of functionality 2 giving the users themselves the possibility to define the properties which can be assigned to the text elements. This gives the annotation tool a lot more flexibility and allows more sophisticated metadata creation for advanced users.

[1] see section 2.2

Functionality 6: Adding annotations exclusively to metadata and not to text

This functionality emerged during the development cycle (see section 4.4) and further extends the customisability of the metadata. Having in mind user story 9, it is important that the metadata specification is complete. For example, if certain information is left out of the document text for reasons of better text flow, it should still be possible to add them to the metadata. In the same way, it should be possible to add text to the metadata which is just slightly modified compared to the text that appears in the document content (e.g. remove capital letters or add more decimal places to a number).

Functionality 7: Upload to a scientific knowledge graph

This functionality is the main requirement for user stories 3 and 9. In order for downstream applications like document comparison to use the specified metadata in the documents, it must be uploaded to the web and stored there in a form which makes it accessible for the users. In large parts the upload functionality is out of the scope of the here presented software which mainly focuses on the creation of the metadata. However, the upload functionality should be straight forward to implement for any knowledge graph platform.

These are the central functionalities of the annotation tool which is outlined in this chapter. As is common in agile development, some of these functionalities were only devised during the development process and were not predefined from the start. The agile development process is described in the next section. More information on how the listed functionalities are implemented technically as a LaTeX package can be found in chapter 5.

4.4 Development Process

To develop the solution described in this chapter I made use of agile software development [63] practices. Agile is a software development methodology which has gained a lot of traction since its conception in the early 21st century. As a methodological approach it originated out of the necessity for more flexible software development in contrast to the earlier waterfall approaches. Gheorge et al. [66] define the agile methodology as follows.

Definition 4.4.1 (Agile)**.** Agile is a set of values and principles and involves the delivery of a good software product to the customer, using an adaptive, incremental and iterative way of working by cross-functional and self-organized teams.

The development process which was applied in this work mainly adopts the principles of iterative cycles, incremental improvements and the definition of user stories from the agile methodology. Figure 4.1 shows an agile development cycle. Step 1, the requirements phase, begins with the definition of user stories. From the user stories I derive specific functionalities in step 2, the design phase. These functionalities are then implemented by defining tasks in the development phase. All the

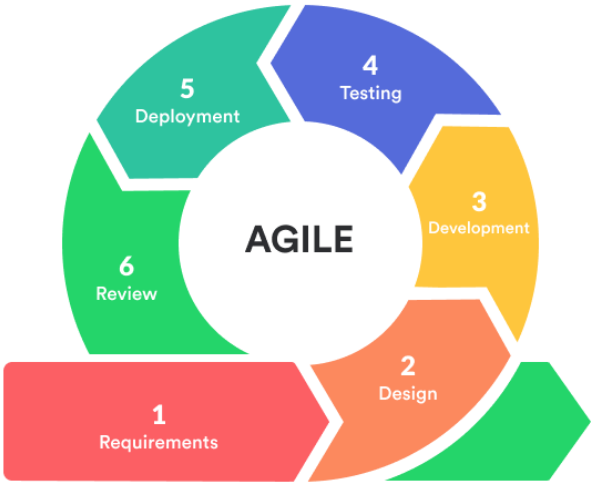

Figure 4.1: The agile development cycle. Source: `https://indevlab.com/blog/what-is-agile-development/`

functionalities have dedicated tests (test-driven development) which are run in step 4. The tests also comprise all the critical edge cases which could be conceived. The deployment phase (5 in figure 4.1) is not very relevant here because the software is still in early development stage where it does not have any users. Therefore, deployment mainly consists of releasing a new version in the version control system for backtracing purposes. During the first cycles with no user feedback, the review phase (6 in figure 4.1) is mainly introspective and consists of regular discussions with the supervisors of the work. As soon as the first user evaluations are performed they are analysed in the review phase and taken into account for the next cycle. After the review, the cycle either ends or starts again with new user stories.

Typically, cycles recur in short time intervals (around 2-3 weeks) and do not include writing excessive documentation for the developed features. The agile approach is designed to allow maximum flexibility. Features coming from user stories can be scratched again if they do not meet the requirements any more and new requirements arise at any iteration. This is why documentation is of secondary importance. Once a functionality is completely implemented and not probable to be removed again, it will be described in the documentation.

5 Implementation

This chapter introduces the technologies which I chose to implement the solution and illustrates the reasoning behind the choices (section 5.1). Furthermore, the implementation is explained in detail both in terms of how it was built as well as how it can be used to annotate contributions with LaTeX in section 5.2. Another important aspect of the implementation is the metadata format and how the metadata are embedded into the PDF.

5.1 Technology Choices

LaTeX is a very popular tool for creation of scientific publications as discussed in section 2.4. Compared to alternatives like Microsoft Word, LaTeX features a whole ecosystem of open source extensions which are built by an active community. Extension packages for the LaTeX type-setting system are freely distributed over the internet and can be built by anyone with the technical knowledge. Furthermore, LaTeX as a system relies on text markup to tag the source document with commands which determine the output. This means that it is not necessary to build graphical user interface components to implement new features like with WYSIWYG (What You See Is What You Get) word processors. Implementing the tool as a LaTeX extension is a logical first step while similar tools for other word processors are also conceivable but probably require a larger development overhead.

Another big advantage of LaTeX is that it is possible to influence the behaviour of the PDF production directly. This makes it possible to embed metadata at time of PDF compilation instead of after the PDF creation through a separate program. While possible, interfering with the pdfTeX engine is not trivial and an extensive task. Fortunately, with LuaTeX (see section 2.4.2) there exists an alternative PDF engine which features the embedded Lua scripting language and callback hooks to the most important events in the PDF generation process. LuaTeX is a modern LaTeX compiler and compatible with most LaTeX writing environments. Implementing the package with LuaTeX is therefore the best decision. Implementations for other TeX engines are of course still possible in the future if necessary.

To embed metadata into PDF, XMP is the recommended standard by Adobe[1] and the only viable option. Since XMP is based on a subset of RDF/XML it is already perfectly fitting to embed the knowledge graph in the metadata. The LaTeX package is developed as open source software and published on Github[2].

[1] `https://www.adobe.com/products/xmp.html`

[2] `https://github.com/Christof93/SciKGTeX`

5.2 LaTeX Package

As explained in the previous section the LaTeX package is implemented in LuaTeX. A LuaTeX package consists generally out of parts written in TeX/LaTeX and parts written in the programming language Lua. In the case of this package, the TeX part contains merely the custom LaTeX commands which serve as interfaces to the functions implemented in Lua. The functionality itself is implemented in Lua which is a far simpler language.

In the next subsections I will revisit most of the conceptualised functionalities described in section 4.3 and explain how they are implemented as parts of the LaTeX package. I will also illustrate on examples how the package is supposed to be used (see figure 5.1 for a comprehensive usage example).

5.2.1 Installation

To use the LaTeX package, it is necessary to relocate the two files `scikgtex.sty` and `scikgtex.lua` into the LaTeX project folder of the document. These files can be obtained from the github repository. Integrating the SciKGTeX functionality into the project can be achieved by putting `\usepackage{scikgtex}` into the document preamble. For the package to work it is necessary to compile the LaTeX source with LuaLaTeX. This can be achieved either by using the `lualatex` command from the command line or by setting the default compiler of the LaTeX development environment to LuaLaTeX. This option can typically be found in the settings of most modern LaTeX environments such as Overleaf, MikTeX or TeX Live.

5.2.2 Mark Up

The first two functionalities mentioned in section 4.3 (functionality 1 & 2) were the possibility to mark elements in text and assign certain properties to the marked elements. I implemented this by defining new LaTeX commands which can be used to mark specific parts in the document. Five commands were reserved for the most important properties describing a scientific contribution: research problem, background, method, result and conclusion. These command names were chosen from the DEO classes (see section 3.2) as suggested by the approach of Oelen et al. [62]. The DEO class of *research statement* was adapted to the *research problem* property which is a central concept of the ORKG vocabulary. For each of the 5 properties a corresponding LaTeX command exists in the following form:

(1) `\researchproblem{..}`

(2) `\background{..}`

(3) `\method{..}`

(4) `\result{..}`

(5) `\conclusion{..}`

```
\documentclass{article}
\usepackage{scikgtex}

\title{Effectiveness of Amoxicillin/Clavulanate Potassium in the Treatment of Acute Bacterial S
\author{Ellen R. Wald \and David Nash \and Jens Eickhoff}

\begin{document}
\maketitle

\begin{abstract}
\background{The role of antibiotic therapy in managing acute bacterial sinusitis (ABS) in child
The purpose of this study was to determine the \researchproblem{effectiveness of high-dose amox

This was a \method{randomized, double-blind, placebo-controlled study}.
Children 1 to 10 years of age with a clinical presentation compatible with ABS were eligible fo
\method{Patients were stratified according to age (<6 or ≥6 years) and clinical severity and ra
A symptom survey was performed on days 0, 1, 2, 3, 5, 7, 10, 20, and 30.
Patients were examined on day 14.
Children's conditions were rated as cured, improved, or failed according to scoring rules.

Two thousand one hundred thirty-five children with respiratory complaints were screened for enr
Fifty-eight patients were enrolled, and 56 were randomly assigned. The mean age was 6630 months
Fifty (89\%) patients presented with persistent symptoms, and 6 (11\%) presented with nonpersis
In 24 (43\%) children, the illness was classified as mild, whereas in the remaining 32 (57\%) c
Of the 28 children who received the antibiotic, 14 (50\%) were cured, 4 (14\%) were improved, 4
Of the 28children who received placebo, 4 (14\%) were cured, 5 (18\%) improved, and 19 (68\%) e
\result{Children receiving the antibiotic were more likely to be cured (50\% vs 14\%) and less
ABS is a common complication of viral upper respiratory infections. \conclusion{Amoxicillin/pot
\end{abstract}
\end{document}
```

(a) LaTeX source code

Effectiveness of Amoxicillin/Clavulanate Potassium in the Treatment of Acute Bacterial Sinusitis in Children.

Ellen R. Wald David Nash Jens Eickhoff

December 19, 2021

Abstract

The role of antibiotic therapy in managing acute bacterial sinusitis (ABS) in children is controversial. The purpose of this study was to determine the effectiveness of high-dose amoxicillin/potassium clavulanate in the treatment of children diagnosed with ABS.

This was a randomized, double-blind, placebo-controlled study. Children 1 to 10 years of age with a clinical presentation compatible with ABS were eligible for participation. Patients were stratified according to age (<6 or 6 years) and clinical severity and randomly assigned to receive either amoxicillin (90 mg/kg) with potassium clavulanate (6.4 mg/kg) or placebo. A symptom survey was performed on days 0, 1, 2, 3, 5, 7, 10, 20, and 30. Patients were examined on day 14. Children's conditions were rated as cured, improved, or failed according to scoring rules.

Two thousand one hundred thirty-five children with respiratory complaints were screened for enrollment; 139 (6.5%) had ABS. Fifty-eight patients were enrolled, and 56 were randomly assigned. The mean age was 6630 months. Fifty (89%) patients presented with persistent symptoms, and 6 (11%) presented with nonpersistent symptoms. In 24 (43%) children, the illness was classified as mild, whereas in the remaining 32

(b) PDF output

Figure 5.1: Source and output of a document tagged with the LaTeX package (paper source [1])

To add arbitrary types of properties, there is a command

(6) `\contribution{..}{..}`

which takes two arguments: a contribution property name and a value. These commands trigger functions in Lua code which add nodes to an internal metadata graph representation which can be serialised to RDF/XML to output the metadata.

A scientific paper typically has a small number of distinct contributions. In the case that there is more than one contribution in the same document, all the above commands accept an optional argument which allows to distinguish the contributions. The optional argument can be any identifier but is most intuitively understood as an enumeration. For example, these two commands add two contributions with the respective research problems:

(7) `\researchproblem[1]{..}`

(8) `\researchproblem[2]{..}`

The two contributions can have their own background, methods and results which must be numbered accordingly. If two problems have a property in common (e.g.: the same background, or methods), it is possible to assign the same property to two contributions using a comma between the arguments, for example as such:

(9) `\method[1,2]{..}`

Another discussed feature was linking of URI's to text elements (functionality 3). I realised this with a `\uri` command.

(11) `\uri{..}{..}`

The `\uri` command internally creates a structure which ends up in the RDF/XML metadata in such a form:

```
<rdf:Description rdf:about="https://www.orkg.org/resource/...">
  <rdfs:label>antibiotic therapy</rdfs:label>
</rdf:Description>
```

The URI in the first argument is converted to a node of the XMP knowledge graph and an optional second argument to the command is added as a label property.

A requirement which only came up after the second development cycle is a functionality to mark properties whose values should not appear in the document text (functionality 6). This can be achieved using the starred variant of the pre-defined or self-defined LaTeX commands. To illustrate this, we assume a research finding which did not make it into the final paper but is still interesting enough to be added to the metadata. I present a possible mark up to achieve such a metadata specification in listing 5.1. The information in the commands ends up in the metadata but is invisible in the PDF document for the common reader.

Listing 5.1: Usage of 'invisible markup'

```
We tested the same experiment with different parameters
and could not reproduce the result.
\conclusion*[2]{No improvements}
\contribution*[2]{number of iterations}{10000}
\result*[2]{52\% accuracy \contribution*[2]{accuracy}{0.52}}
\method*[2]{Support Vector Machine}
```

5.2.3 Custom Annotation Properties

Another requirement which was identified in chapter 4 is the functionality to define custom properties for contributions (functionality 5). For one part this is possible with the `\contribution` command which allows to specify a property. Apart from that there is a more sophisticated command which lets the user define new properties:

(12) `\addmetaproperty{..}`

The defined properties can then be used as the first argument in a `\contribution` command. The `\addmetaproperty` command accepts the name of the property and as an optional argument an alternative namespace. This command allows to influence how the RDF in the XMP output is structured and is therefore a more advanced capability.

5.2.4 Metadata Storage

After annotation of the metadata elements, they must be stored and embedded into the created PDF document (functionality 4). With LuaTEX it is possible to bind callback functions to the PDF creation process and append elements to the PDF file. In the Lua code the metadata graph is serialised into RDF/XML and appended to the PDF file in a metadata stream at the `finish_pdffile` callback. The actual format of the metadata is discussed in the next section.

5.3 Metadata Format

As a format for the embedded metadata which is produced by the LaTeX package, I follow the XMP standard (see section 2.5) which is recommended by Adobe to add metadata to PDF documents. XMP metadata are most commonly serialised to an RDF/XML format [46]. Since RDF is a very fitting format for storage of semantic information this format can be used to represent the annotations of contributions. To model the paper and contributions, I largely rely on the ORKG core ontology[3]. Specifically on the classes `Paper` and `ResearchContribution`, as well as the predicate `hasResearchContribution`. In code listing 5.2 the xmp output of the example

[3] `https://github.com/ORKG/orkg-core-ontology`

Listing 5.2: How the metadata of the example paper looks like in the PDF stream

```
1  <x:xmpmeta xmlns:x="adobe:ns:meta/">
2  <rdf:RDF
3    xmlns:orkg="http://orkg.org/core#"
4    xmlns:orkg_property="http://orkg.org/property/"
5    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
6    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
7    <rdf:Description
8       rdf:about="https://www.orkg.org/orkg/paper/
9       48fdc517-5814-4d0c-cd03-0c296941c6">
10     <rdf:type rdf:resource="http://orkg.org/core#Paper"/>
11     <orkg:hasResearchContribution>
12       <orkg:ResearchContribution
13         rdf:about="https://www.orkg.org/orkg/paper/
14         48fdc517-5814-4d0c-cd03-0c296941c6/contribution_1">
15           <orkg_property:background>
16             The role of ... is controversial
17           </orkg_property:background>
18           <orkg_property:researchproblem>
19             effectiveness of high-dose ... diagnosed with ABS
20           </orkg_property:researchproblem>
21           <orkg_property:method>
22             randomized, double-blind, placebo-controlled study
23           </orkg_property:method>
24           <orkg_property:method>
25            Patients were stratified ... placebo
26           </orkg_property:method>
27           <orkg_property:result>
28            ... less likely to have treatment failure ...
29           </orkg_property:result>
30           <orkg_property:conclusion>
31             ... results in significantly more cures...
32           </orkg_property:conclusion>
33       </orkg:ResearchContribution>
34     </orkg:hasResearchContribution>
35   </rdf:Description>
36 </rdf:RDF>
37 </x:xmpmeta>
```

from figure 5.1 is displayed. There are two ORKG namespaces: `orkg` for the ORKG core ontology and `orkg_property` as a common namespace of all the user-defined properties. The paper is the top level node in the graph (line 7) and assigned a URI with a unique UUID[4]. It is assigned to the `Paper` class on line 10. The paper node

[4]A universally unique identifier. `https://datatracker.ietf.org/doc/html/rfc4122`

can have a number of `hasResearchContribution` edges (line 11) which point to a `ResearchContribution` node (line 12). The `ResearchContribution` also has a URI and can have a number of property predicates which contain the actual markup.

A specific requirement on the metadata is raised by User Story 7, namely that of PDF/A compatibility. The inclusion of XMP metadata is required in PDF/A but also restricted by default to a defined set of properties. Custom metadata fields and properties can be added with XMP extension schemas. Full PDF/A compatibility can currently only be guaranteed by the LaTeX package if the five standard commands (1-5) are used. If user defined properties are added to the metadata, the XMP is not compatible with PDF/A at the moment but should become compatible in future work.

6 Evaluation

In this chapter I describe the user evaluation which I conducted over a period of three weeks between 09.03.2022 and 30.03.2022 after the first preliminary release of the SciKGTeX LaTeX package. In section 6.1 I describe the overall design of the evaluation experiment and give the hypotheses which were tested with the collected data. In section 6.2 the results of the evaluation are presented together with the statistical analysis which was performed to test the evaluation hypotheses.

6.1 Experiment Design

The goal of the evaluation was to test the usability of the approach with potential users who are given a small series of tasks (see section 6.1.3) to complete using the package. The test serves to collect a number of metrics which are analysed to reveal the usability, convenience and usefulness of the developed solution. These metrics will support the answers to the research questions. Before the test I established three specific hypotheses (Hypothesis 1 - Hypothesis 3) which I planned to test in the evaluation. Hypothesis 1 and 2 are connected to RQ 1 and hypothesis 2 and 3 give insights into RQ 2. The hypotheses are formulated as alternative hypotheses which I can verify by refuting their negated counterparts, the null hypotheses.

Hypothesis 1: The system is easy and convenient to use.

Hypothesis 2: Annotation of main contributions in a short text summary can be performed in less than 10 minutes.

Hypothesis 3: Different annotators produce similar annotations.

It must be noted that hypothesis 1 is the only statement which concerns just the developed LaTeX package itself and its functionality. Hypotheses 2 and 3 on the other hand are only partly indicators of the usability of the annotation tool. Testing these hypotheses is also a test of the feasibility of crowd-sourced annotation of contributions since it assesses the difficulty of the tasks which are executed (e.g. finding the main contributions and attributing different properties like research problem, method, etc. to parts of the text related to the main contribution). While hypothesis 1 and 2 mainly concern usability and convenience, hypothesis 3 is rather evaluating the usefulness of the solution.

Hypothesis 1 is a hypothesis which can be explored by assessing the perceived usability by the participants. This is achieved using the System Usability Scale (see section 6.1.5) and feedback by the participants after the evaluation.

Hypothesis 2 was formulated to emphasise the simplicity of the annotation tool. It will be tested by measuring the time delta variable while the participants annotate main contributions of the same sample text. If it holds true, it proves that with little prior training typical users can learn to achieve the most important objectives of the LaTeX package in little time.

If Hypothesis 3 can be verified it indicates that the tool is able to produce consistent metadata from different users on different documents. In the following subsections I will detail the design of the evaluation covering the topics of candidate selection, procedure, specific tasks and calculation of the metrics.

6.1.1 Candidate Selection

For the evaluation 26 volunteers were recruited from the University of Innsbruck (7), the Leibniz University Hannover (6), the Leibniz Information Centre for Science and Technology University Library[1] (5) and several other institutions (8). I considered only people with at least an undergraduate degree but beyond that I intentionally kept the group of participants as heterogeneous as possible in terms of academic experience. This makes it possible to compare groups of researchers with very little experience to groups of researchers with more experience and contrast their usage of SciKGTeX. Additional to that, it was important to include both male and female participants as well as various age groups to be aware of any possible correlations of age, gender or academic experience with the outcome of the measured variables.

6.1.2 Procedure

To test the hypotheses I designed an approximately 30 minutes long evaluation procedure which was executed in a live online meeting with individual participants. The procedure of the test consists of the following consecutive steps which were walked through with every participant.

1. Give the participant approximately 5 min to read the SciKGTeX documentation[2] and make sure that they understood the idea behind it.

2. Introduce the participant to the testing environment and give the first task.

3. Measure time until completion of task 1 by the participant.

4. Introduce tasks 2 and 3 and let the participant complete them.

[1]The Leibniz Information Centre for Science and Technology University Library which is also known as the Technische Informationsbibliothek (TIB) is the German National Library of Science and Technology as well as Architecture, Chemistry, Computer Science, Mathematics and Physics

[2]The documentation is provided on the Github repository: `https://github.com/Christof93/SciKGTeX/blob/main/README.md`

5. Let the participant fill in the survey questionnaire[3].

6. Let the participant give additional oral feedback.

The participants could choose from 12 possible slots per day (12 days total) to hold the live test session and were given the instruction to read the documentation already with the date confirmation email. This means the reading in step 1 was mostly done without supervision shortly before the test session and participants could then ask questions critical for understanding the tool at the beginning of the test. In step 2 I gave the participants a link to a LaTeX project on Overleaf with a file already in place containing a short abstract of a scientific paper.

For EH 2 it was necessary to measure the time it took participants to complete task 1 (step 3). I did not inform the participants that time was measured for this task to prevent a bias due to the participants trying to finish the assignment as fast as possible. Time pressure would not be the realistic environment for the first time usage of the tool which is simulated here. After participants affirmed that they had completed the first task, I took the time and introduced them to the second and third task in the same manner.

After step 4 the work on the tasks was finished and I provided the participants with a link to the online survey. The survey questionnaire was divided into three parts where the first part would assess general demographics, prior knowledge and affiliation of the participants. The second part consisted of the questions for the System Usability Scale described in section 6.1.5. In the end the participants were presented with a free text form to word their feedback and also invited to give oral feedback if preferred.

6.1.3 Tasks

During the course of the test procedure I gave three tasks to the participants which were the following:

1. Annotate the 5 properties of the main contribution (background, research problem, method, result and conclusion).

2. Find a unique resource identifier for the term 'Natural Language Processing' and link it to the expression in the text. Annotate the resource as a method.

3. Find a new optional property which you want to annotate in this text. Check if it exists on the ORKG website.

The participants were allowed to consult the documentation of SciKGTeX at all times during the test. With the help of the documentation most of them were able to understand the tasks easily. Additionally, if users had trouble understanding a task assignment I took notes to rewrite the concerning sections of the documentation as a result of the first evaluation.

[3]The exact questionnaire which was used can be found here: `https://doi.org/10.5281/zenodo.6544552`

6.1.4 Experiment Material

The text[4] which was used to perform the tasks on was a paper on the topic of Requirements Engineering [67]. I chose this topic and specific paper abstract based on the common backgrounds of the candidates which volunteered for the evaluation. While it was not possible to test each participant on a document that they authored themselves, they should be capable of understanding the text with relative ease to simulate the scenario of authorship as close as possible. The fact that the participants are not actually the authors of the text which they annotate in this test must be considered as a possible threat to validity of the experiment. It was however not possible to let the participants work on their own texts because the given-above evaluation hypotheses could only be evaluated by measuring comparable values in the independent variables. In order to be comparable all participants have to work on the same underlying text.

6.1.5 Metrics

To further investigate the evaluation hypotheses I defined several independent variables which I measure for each sample of the evaluation. In this subsection I explain the metrics which were used to measure the independent variables. All the calculations were performed with Python and are open to inspection[5].

System Usability Scale The System Usability Scale (SUS) score is widely used to measure usability of software systems [68]. The score as an indicator of usability gives a quantitative measure for hypothesis 1 in particular. The SUS is calculated from a collection of 10 statements about the user experience of the system. Of course this assumes that the user interacted in a meaningful way with the system before. The following statements are used to determine the SUS score.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.

[4]The actual text used in the test can be found here:`https://doi.org/10.5281/zenodo.6544552`

[5]All calculations of metrics can be found in the Python notebook here: `https://doi.org/10.5281/zenodo.6544552`

8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

For the answer possibilities a five-level Likert scale is used with the typical options ranging from 'Strongly agree' to 'Strongly disagree'. The statements are arranged such that the end of the scale which maximises the score switches for each statement. This is reflected in equation (6.1) by the modulo 2 operations. Each statement is assigned an integer value which is obtained by mapping the Likert scale to the range of 1-5. The usability score S_u for a user u is calculated as follows.

$$S_u = 2.5 * \sum_{i=1}^{|Q|} (i \mod 2) * (v(q_i) - 1) + (1 - (i \mod 2)) * (5 - v(q_i)) \tag{6.1}$$

In the above equation Q stands for the set of statements q from the usability scale and $v(q_i)$ is the value assigned to the ith statement of the SUS. Finally, the value is scaled by 2.5 to give a more familiar percentage score.

Gestalt Pattern Matching Furthermore, a measure to quantify hypothesis 3 was needed. To determine whether the participants of the test produced consistent annotations, I applied a measure to determine the similarity of the annotation. A possibility to measure similarity of annotations is using Gestalt Pattern Matching which is also known as Ratcliff/Obershelp Pattern Matching [69]. The idea of Gestalt Pattern Matching is to give the similarity of two sequences a number between 0 and 1, where 0 is different in all elements and 1 is identical. The algorithm of Gestalt Pattern Matching starts with finding the longest common subsequence (LCS) of the two sequences and then recursively finding the LCS of the remaining sequences on the left and right of the initial LCS. Next, the lengths of all the common subsequences are summed to the number of overlaps K. The similarity score D_{ro} between two sequences of annotations S_1 and S_2 is then calculated according to equation (6.2).

$$D_{ro} = \frac{2K}{|S_1| + |S_2|} \tag{6.2}$$

While Gestalt Pattern Matching is specifically formulated for similarities of strings (i.e. sequences of symbols), it can be generalised to other sequences such as sequences of words which is more fitting for the results of the user evaluation. The reason for this is that the measure should not take into account sub word overlaps between annotations because two different words are completely different annotations even if some parts of the words overlap. It must be remembered that, in order to derive the annotations, the participants are only selecting a sequence of words from the already present text and not adding their own text.

To compare the similarities among all the annotations I measure the similarity of each type of annotation for all possible pairs of participants p_i and $p_{i'}$. I denote

the similarity of two annotations by the expression $D_{ro}(A_{ann}(p_i), A_{ann}(p_{i'}))$ where $A_{ann}(p_i)$ is the sequence of words comprising the annotation of type *ann* made by the ith participant and the types of annotations are taken from the set of main properties of a contribution $ann \in \{$*background, researchproblem, method, result, conclusion*$\}$. Next, I average the similarities of all the pairs of the same annotation type to get the overall similarity score. Gestalt Pattern Matching is not a commutative operation meaning that $D_{ro}(S_1, S_2) \neq D_{ro}(S_2, S_1)$. This implies that $D_{ro}(A_{ann}(p_i), A_{ann}(p_{i'}))$ must be calculated for any permutation of two participants p_i and $p_{i'}$ to calculate the average similarity.

$$D_{ann} = \frac{\sum_{i \neq i'}^{|P|} D_{ro}(A_{ann}(p_i), A_{ann}(p_{i'}))}{|P|(|P| - 1)} \tag{6.3}$$

Equation (6.3) shows how the overall Gestalt Similarity score for an annotation type is calculated where P denotes the set of participants p.

Fleiss kappa The agreement between annotators also known as inter-annotator agreement or inter-rater reliability was measured by calculating the Fleiss Kappa [70]. The Fleiss kappa is a measure of agreement of m annotators which all assign a nominal value to each of N units. In the case of the evaluation data, every word of the abstract text corresponds to a unit of which the annotator decides whether it belongs to a specific category (property of contribution) or not. Thus, the number of units is equal to the number of words in the abstract and every unit u (word) gets a value v of 1 if it is part of the property annotation or 0 if it is not.

$$K = \frac{P_o - P_e}{1 - P_e} \tag{6.4}$$

The general form of the Fleiss kappa is given in equation (6.4) where P_o is understood as the observed agreement between annotators and P_e is the expected agreement between annotators. I calculated P_o and P_e by means of a coincidence matrix[6] which contains counts of coincidences for all the possible pairs of annotation values between any two annotators. The coincidence matrix is a symmetric V-by-V matrix where V is the number of possible values (for example V = 2 in the case of binary annotation). The frequencies contained in the matrix are calculated according to equation (6.5) where $I(\cdot)$ equals 1 if $\cdot$ is true and 0 if it is false.

$$o_{vv'} = \sum_{u=1}^{N} \frac{\sum_{i \neq i'}^{m} I(v_{iu} = v) \cdot I(v_{i'u} = v')}{m - 1} = o_{v'v} \tag{6.5}$$

The diagonal of the coincidence matrix, for example, contains the counts of cases where two annotators i and i' agree on the annotation value of any unit u i.e.

[6]More information on coincidence matrices can be found here: `https://en.wikipedia.org/wiki/Krippendorff%27s_alpha#Coincidence_matrices`

$v_{iu} = v_{i'u}$. I further define the sum of frequencies for a value v as follows.

$$n_v = \sum_{\ell=1}^{V} o_{v\ell} \tag{6.6}$$

From the coincidence matrix and the total frequencies we can derive the observed agreement and expected agreement as demonstrated in equation (6.7) and equation (6.8) respectively[7].

$$P_o = \sum_{c} \frac{o_{cc}}{mN} \tag{6.7}$$

$$P_e = \sum_{c} \frac{n_c^2}{(mN)^2} \tag{6.8}$$

6.2 Results

In the following section I will present the results[8] of the evaluation which was conducted to verify the hypotheses. The sample size of the test was 26 participants from different institutions with varying levels of prior knowledge of LaTeX and Semantic Web technologies. In section 6.2.1 I will give an overview on the demographic indicators of the participants and introduce different groups based on the information obtained in the first part of the evaluation survey. In section 6.2.2 I present the results of the usability analysis and put into perspective the average SUS score for the LaTeX package. Further, I report the mean time to completion of task 1 and explore possible correlations between prior knowledge of participants and the usability score. The question whether annotation was performed consistently by the participants is answered in section 6.2.3 where I report the Gestalt Pattern Matching similarity score and the Fleiss kappa as inter-annotator agreement measures.

6.2.1 Demographics

Starting with gender, 69% of the participants were male and 31% were female. The most prominent age group was 19-29 years (69%) and no participant was older than 52 years. A little more than 50% of the participants (14) were currently pursuing a PhD degree at the time of the test, while another 19% were master students and 11% worked as post-doctoral researchers. The rest of the participants were working in different research-related positions.

[7] The calculation was largely derived from this section in Wikipedia comparing the Krippendorff alpha to the Fleiss kappa: `https://en.wikipedia.org/wiki/Krippendorff%27s_alpha#Alpha's_embrace_of_other_statistics`

[8] All the results and the analysis which was done on them can be found as a dataset here: `https://doi.org/10.5281/zenodo.6544552`

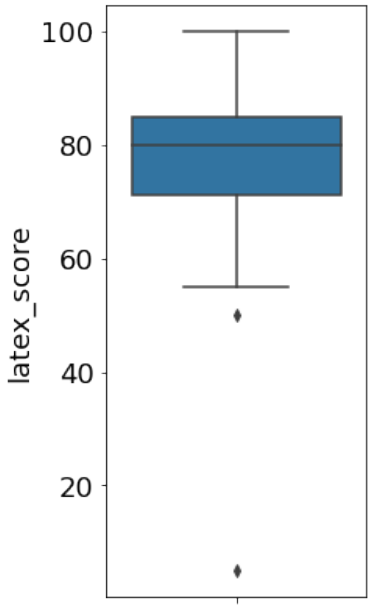

Figure 6.1: Distribution of prior knowledge on LaTeX among the participants

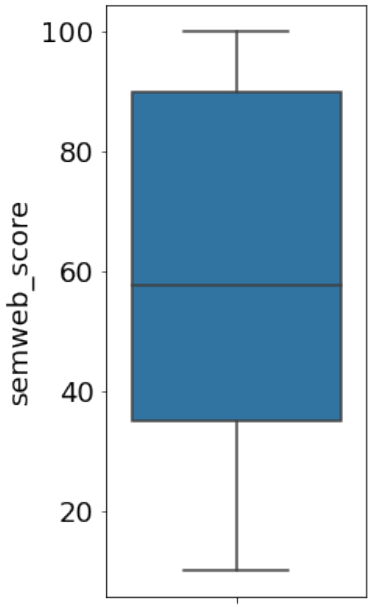

Figure 6.2: Distribution of prior knowledge on the Semantic Web among the participants

To get an idea on the level of prior knowledge of the participants in the fields of LaTeX and Semantic Web technologies I prepared sections of questions and statements in the questionnaire which were answered or rated by the participants. The section used to test LaTeX knowledge consisted of the following elements.

1. I am familiar with the LaTeX document system.
2. How many years of experience do you have with LaTeX?
3. Compared to other tools, how often do you use the LaTeX document preparation system to write publications, reports or similar documents?
4. I prefer LaTeX to produce scientific documents.

For the statements a 5 level Likert scale was used and the questions had five answer possibilities as well which were ordered from negative through neutral to positive. To test the background in Semantic Web technologies I asked the participants to rate the following statements.

1. I am familiar with the concept of the Semantic Web.
2. I have worked with Semantic Web technologies (Knowledge Graphs, ontologies, etc.).
3. I know the Resource Description Framework (RDF).

4. I know the difference between Uniform Resource Identifiers (URIs) and Uniform Resource Locators (URLs).

5. I am familiar with the FAIR (Findable, Accessible, Interoperable, Reusable) principles for data management.

I calculated an overall Likert scale from the items similar to the SUS calculation to get a representative number for each of the two fields of prior knowledge. For this I summed the scores of each question and scaled the result to 0-100 range to get a percentage value. As can be seen in figure 6.1 the familiarity with LaTeX was high among the participants which adds to the claim made in section 2.4 that LaTeX is widely used in the academic sector. The knowledge of Semantic Web concepts was far more varied at a standard deviation of 31 points around the mean. Notice the large extent of the second and third quartile in figure 6.2. While some of the researchers which were part of the test were working at least partly in the field of Semantic Web, others did not have a lot of knowledge of it. These scores paint a comprehensive picture of the background of the participants of this test which can be taken into account in the further analysis.

6.2.2 Usability

The usability of the LaTeX package was measured with the help of the System Usability Scale as described in section 6.1.5. For interpretation of the score I rely on the work of Bangor et al. [71] who mapped the usability scale to a 7 adjective scale comprising 'Worst Imaginable', 'Bad', 'Awful', 'Poor', 'OK', 'Good', 'Excellent', 'Best Imaginable'. They ran 212 SUS surveys and collected the adjective ratings by the participants along with them. The systems rated with 'Good' had a mean SUS score of 71.4 ($\sigma = 11.6$) while the 'Excellent' rating was assigned at a mean score of 85.5 ($\sigma = 10.4$). The overall mean SUS score of the SciKGTeX package turns out to be 79.8 ($\sigma = 11.6$). The second and third quartile are situated between 75.0 and 85.0. This ranks the package clearly closer to 'Excellent' than 'Good' in terms of matching adjective. When looking at the different groups of occupations in figure 6.3 among the participants, it can be observed that PhD students rated a slightly higher mean SUS score at 82.3 ($\sigma = 12.34$).

Another thing I looked at is the correlation between prior knowledge and the usability score outcome. A positive correlation between the LaTeX score and the SUS score for example would suggest that a higher level of prior knowledge of LaTeX leads the user to assign a higher usability score to the package. To explore the correlation I calculated Pearson's correlation coefficient between the variable of LaTeX score and SUS score which amounts to 0.2. This means that there is practically no correlation between these two variables further implying that prior LaTeX knowledge does not substantially influence the usability of the package. Also, the Semantic Web knowledge score is not correlated with the SUS score at a Pearson correlation of 0.13. The combination of near-excellent usability score and independence of prior knowledge make the package easy and convenient to use as was hypothesized in hypothesis 1.

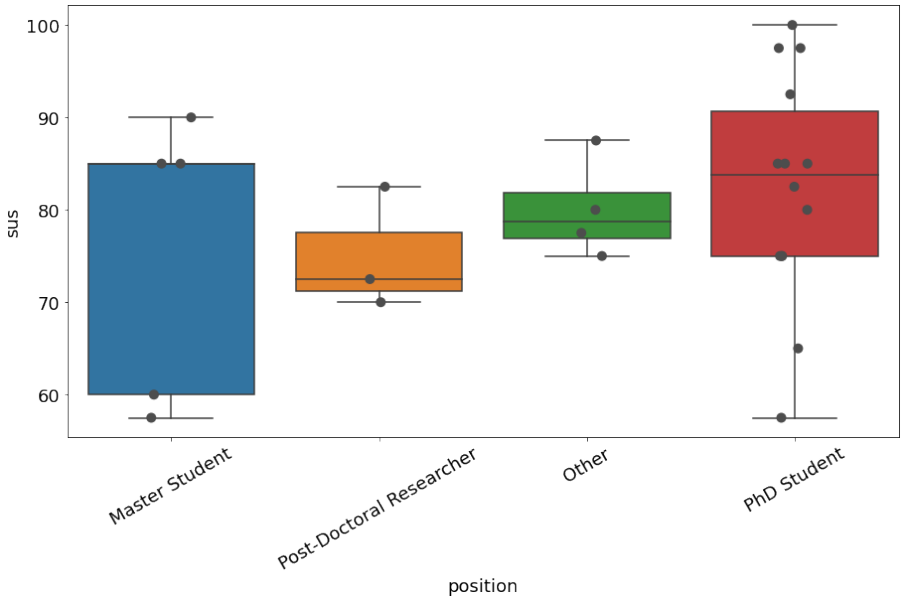

Figure 6.3: System Usability Scale distributions among the different groups of participants.

Furthermore, the variable of time spent on annotation of main contribution gives an indication on usability as it proves or disproves that the idea of the package can be picked up in little time by typical users. The mean duration of work on task 1 in the evaluation is 7 Minutes 34 seconds. The second and third quartiles lie between 3 and 10 minutes. I evaluated the time delta variables with a one-sample t-test. The test implies that the hypothesis 2 (participants take less than 10 minutes to annotate the properties of the main contribution) is verified based on these results as the null-hypothesis can be rejected ($p < 0.0006$).

6.2.3 Inter-annotator Agreement

Measuring the similarity of the annotations made by the participants of the evaluation gives an impression on the usefulness of the tested tool. Only if different annotators agree to some degree on the content of certain annotations the data generated by the annotations is truly comparable and valuable for sophisticated applications. Thus, hypothesis 3 is concerned with the similarity of annotations which was measured both through the Gestalt Pattern Matching and the Fleiss kappa inter-annotator agreement metric.

Table 6.1 contains the result for both metrics on each of the five property annotations of the main contribution. It is apparent that both metrics paint a similar picture where the first three properties *background*, *research problem* and *method* get far less consistent annotation than the *result* and *conclusion* annotations. According to Landis et al. [72] Fleiss kappa values over 0.81 can be considered 'almost perfect agreement' whereas 0.61-0.8 is 'substantial agreement'. Applying this to *result* and

Annotation Category	Gestalt Similarity	Fleiss kappa
Background	0.28	0.21
Research Problem	0.44	0.44
Method	0.31	0.24
Result	0.78	0.74
Conclusion	0.86	0.81

Table 6.1: Different measures of inter-annotator agreement for the different categories of annotations in the LaTeX package.

conclusion they can be considered fairly consistent, especially since the annotations also overlap to 78% and 86% on average respectively according to the Gestalt similarity score. For the other three categories there are big differences in the parts of text which are assigned to them by different annotators. Manual investigation reveals that there seem to be systematic disagreements on what is considered a *research problem* and a *background* which are often tagged in opposing order. The *method* annotation is often split into several annotations of sentence fragments which mention methods but sometimes also a whole paragraph is tagged as a *method*.

Based on the inter-annotator agreement result hypothesis 3 must be rejected at least partially. Contrary to the claim of the hypothesis, in this user evaluation different annotators produced substantially dissimilar annotations. The reasons for this result and possible solutions are discussed in the next chapter along with more discussion of the results of this evaluation with regards to the overall objective of this work.

7 Discussion

In this chapter I revisit the main objectives of this work in detail and discuss if they were achieved. Further, I expand on the results of the user evaluation and draw conclusions from them. I partition the discussion into strengths (section 7.1) and weaknesses (section 7.2) of the current solution, with reference to the user evaluation and feedback. Section 7.2 contains discussion on possible solutions to the presented problems as well. In the end of the chapter I will move on to answer the research questions declared in section 1.1.

7.1 Strengths

The following paragraphs detail the strengths of the LaTeX package approach which crystallised in the user evaluation.

Simplicity The evaluation has shown that SciKGTeX is light-weight and easy-to-use as both learning and using the commands does not take a long time and is rather intuitive. This finding is further backed by comments from the participants:

> Participant 11: "Overall, I think the codes are not difficult to handle."

> Participant 18: "The annotation package is easy to use."

On the practical side, the package could easily be setup on Overleaf by adding two files to the working directory and importing them in LaTeX which demonstrated how easy it is to install SciKGTeX. Further, the workflow does allow to make annotations at the same time as writing the document which is simpler than having to make the annotations later on a different platform.

Usefulness SciKGTeX was received as very useful by the majority of the participants. Many participants left written or oral positive feedbacks such as:

> Participant 6: "Great job. Keep it up!"

> Participant 13: "Good work! :)"

> Participant 25: "Very nice LaTeX package."

They liked the idea and were interested to learn how this project would develop in the future.

Usability During the user evaluation it became apparent that SciKGTeX was highly usable for most of the participants. All the tasks (see section 6.1.3) of the user evaluation were accomplished by all of the participants even though the second and third task were already at the limit of complexity for the short instruction period. This is further supported by the high usability score achieved in the survey. However, there is still room for improvement. Further iterations of the design cycle can alleviate the SUS score from 'near-excellent' to 'excellent' by addressing the feedback of the first test users.

Embedded Metadata SciKGTeX uses a novel approach to embed metadata directly in PDF files. This has various benefits such as the persistence of the data and the decentralised storage of information. Notably the metadata are saved in a machine-readable XMP format which relies on the Semantic Web technologies such as RDF/XML.

Clear provenance of information Another positive aspect of the LaTeX package approach compared to other metadata specification approaches like the ORKG web platform is that it ensures that only the authors of a paper can add contribution metadata. The metadata is added at the time of document creation which implies that the creator must be the origin of the information. This can only be guaranteed provided that the specified metadata is not changed by any other entity unintentionally or even maliciously. With modern read-only PDF-A formats this will however become less of a problem in the future. While other entities may still add further contribution information to the knowledge graphs, the provenance should be stated specifically since any other entities than the authors have a different view point and bring a bias into the data.

7.2 Weaknesses

Aside the positive feedback the user evaluation still revealed some problems which were either voiced by the participants themselves or became visible to me during the live testing or analysis of results. In the following, I will address the biggest problems of the implementation which came up during the user evaluation and present possible solutions to them.

Complex and confusing functionality Due to the short time frame of the user evaluation not all functionalities of the LaTeX package were actually tested. I concentrated on the most basic operations that should be the most interesting for the general user. For example the functionality of contribution numbering (see section 5.2.2 items 7 & 8) was intentionally not included in a task since it does not fulfil the requirement of simplicity in the current form and is still under development. Some participants confirmed that they had a hard time understanding this

concept in the documentation. It is both in the design and in the documentation that some functionalities still need improvement in the coming iterations.

Unclear scope of annotation Many participants stated that they felt the documentation was not specific enough in defining what exactly should be the content of the different annotations (research problem, methods, etc.).

> Participant 25: "I guess the main problem is to determine which parts of the sentence should be selected. For example, the whole sentence including the research problem or only the relevant parts."

Many asked during the evaluation if they were supposed to annotate words, sentences or paragraphs, and some even put custom text instead of encapsulating some of the available text. Of course, the documentation should have accurate definitions of what the different properties of a contribution mean exactly. However, this is far from trivial and different definitions of the same concepts will always exist. Still, the documentation must become clearer on the purpose of the property annotations and suggest some indication on the number of words in an annotation. This will also be a possible solution to combat the bad inter-annotator agreement on the properties *background*, *research problem* and *method*. One easily implementable solution to the problem could be to define that annotations can be at most the length of a sentence.

Unsuitable properties of annotation For some of the property annotation commands it can be argued that they are inherently too ambiguous as expressions. This was voiced by one participant (translated from German):

> Participant 15: "There is potentially a fluid transition between terms like background, research problem and method. If authors use differing definitions the comparisons [...] break."

Instead of trying to define the terms of property commands rigorously in the documentation, it should be considered to replace them with different expressions whose semantics are agreed upon by more people. For example the properties *background* and *research problem* were hard to differentiate for many of the evaluation participants. *Background* can be used to denote – among other things – either the related work which led to the publication or the higher level context of the work. *Research problem* was also often applied to mark higher level context but sometimes also to point out the specific objective of the paper. A possible solution to this would be to remove the *background* property and clearly define *research problem* in the meaning of contextual background. A new property called *Objective* or *Solution* can be introduced to fill a gap in the description capabilities which was noted by several participants.

Only shallow annotation In it's current form the LaTeX package only allows annotation in a key-value style where some property (key) of the contribution is annotated

with a part of the text (value). Some participants in the evaluation were missing more sophisticated methods of annotation like nesting properties in the metadata or completely custom triples. While it is already possible to define custom properties, they are always relations where a contribution acts as the subject and only the type of property relation and the object can be defined. This way, any custom property essentially stays a key-value specification with a custom key. The reason for this is mainly that I tried to keep the functionalities of the package simple and easy-to-understand. The end goal would be that the package is based on simple building blocks but can be used to produce arbitrarily complex metadata depending on the requirements of the user.

Low inter annotator agreement The low inter annotator agreement is an indicator of consistency problems which can arise in crowd-sourced knowledge graph construction. It is very likely that different people will have conflicting views on what the main contributions of a research publication are. There are several solutions suggested above to reduce this problem to an acceptable level, mainly coming down to more rigorous definition of annotation guidelines and choice of unambiguous property keywords. In this experiment a main driver of error is the fact that the participants of the evaluation were not actually the authors and had to form their own understanding of the short text which makes the task even more difficult.

Another take-away from the low agreement is that automatic extraction methods such as Machine Learning approaches will have a hard time achieving accurate results, since even human annotators are not agreeing on the different annotation labels. Conversely, this is another argument for the usefulness of crowd-sourced semantic annotations for metadata of scientific publications as a whole. A human annotator relies on world knowledge and expertise to judge the main contribution of a scientific publication and this will be hard to automatise at the current state of artificial intelligence. SciKGTeX acts as a tool for authors of publications to semantically describe their work by themselves.

7.3 Answers to Research Questions

In this section I revisit the research question RQ 1 and RQ 2 from section 1.1. I provide comprehensive answers to the questions based on the findings presented in this work.

RQ 1 *How can the process of manual semantic annotation of research contributions in scientific articles be simplified?*

With the development of the LaTeX package I have shown that basic semantic information can be directly embedded into the document metadata at the time of document creation (i.e. at the same time as writing the text of the document itself). Through the usability evaluations I have shown that the system is understandable, intuitive and easy-to-use. The process of annotation is simple enough for a typical

researcher to achieve it in little time.

SciKGTeX is not reliant on any other systems than the LuaTEX document typing system and does not require a connection to the internet to produce the metadata. This is a simplification compared to the approach where document creation and metadata specification are separated systems.

With the novel embedding approach, metadata are directly saved into the PDF files which saves them from perishing or getting detached from their source material. Furthermore, the authors themselves dispose of their semantic contribution metadata and do not have to rely on any third-party applications to publish and manage them.

RQ 2 *How do researchers use SciKGTeX to semantically annotate research contributions in scientific articles at the time of document creation?*

The user evaluation has shown that a representative group of researchers from different universities was able to use the LaTeX package to produce valuable contribution metadata. Annotating the main contribution in a short text was achieved in well under 10 minutes by the majority of the participants with only little prior exposure to the package documentation. The participants had no problem using SciKGTeX with and without extensive prior knowledge of LaTeX or Semantic Web concepts. The resulting metadata is machine-readable and can be used to build large knowledge bases which facilitate various applications from which the researchers can profit in turn.

The low inter-annotator agreement in some of the property annotations poses a threat to the comparability of the produced metadata which can be a problem for various applications. This problem will be addressed in future releases of the package by the solutions suggested in this chapter.

8 Summary of the Thesis

In this chapter I will summarise the main contribution of this work and give a conclusion in section 8.1. I will then sketch the plans for future development of the package in section 8.2.

8.1 Conclusion

In this master thesis I suggest a solution for authors of scientific publications to annotate machine-readable metadata about the contributions of their publications. The semantic annotations serve to build scientific knowledge graphs which can alleviate problems arising from an overabundance of scientific publications such as inefficient keyword searches [7] or filter bubbles [9]. Different from previous solutions I present a LaTeX package called SciKGTeX which allows to directly specify the metadata at the time of document creation and embed them into the finished PDF file. This is a simplification compared to an approach where the metadata specification is handled through a separate web interface.

Further, I conduct a user evaluation with 26 participants to measure the usability, the mean time to complete basic annotations and the inter-annotator agreement of annotations created with the package. From the results I conclude that the LaTeX package is convenient and easy-to-use with a mean score of 79.8 on the System Usability Scale and an average time consumption of less than 10 minutes to learn and accomplish the main functionality of the package. The inter-annotator agreement of the property annotations is still lacking for the property keywords of *research problem*, *method* and *background* which will be addressed in future work.

Finally, SciKGTeX is a successful implementation of an annotation framework for metadata of scientific contributions. It is arguably simpler than other approaches at the same objective such as [62] or [19] and allows the author to specify metadata directly at the time of document creation. The user evaluation has confirmed that SciKGTeX can be used by the research community to transform scientific content into machine-actionable metadata. Further benefits of the approach are the compliance with Semantic Web standards, decentralised information storage and the ability to produce enriched PDF documents. SciKGTeX has the potential to act as an important building tool for large-scale scientific knowledge graphs which facilitate the development of supportive applications to elevate the modern research workflow to the next level.

8.2 Future Work

As described in section 4.4 the implementation presented here is the first iteration in an agile software design cycle. The development will continue on as an open source project to bring it to it's full potential. In this section I will first detail the immediate plans for the next development cycle and then sketch the vision for the more distant future.

Short-term Plans The short-term plans include the changes which have become obvious after the first user evaluation and incorporate the feedback from the participants. The plan is to start a new development cycle and execute the following updates in the next months.

1. Refactor the annotation commands for *background* and *research problem* and add a command for *objective*.
2. Ensure full PDF/A compatibility.
3. Implement ways to generate more complex metadata for example by enabling the users to inject custom RDF snippets.
4. Rework the documentation of the package to be more specific and incorporate links to tutorials on Overleaf.

Long-term Vision More farsighted plans are to transform the software into a more flexible tool with which the users can define arbitrarily complex facts to add to the document metadata in RDF format while relying on simple building blocks. Meanwhile, the user experience should stay as simple and elegant as possible. Ideally the package can be used as a framework to define custom metadata templates which can be used for example for journals or conference proceedings. These custom templates allow aggregation of consistent metadata on papers from the same research area.

Furthermore, on online LaTeX writing environments like Overleaf it could be possible to give the users recommendations for the commands and templates which they can use through requests to API's of scientific knowledge graphs. The vision is that SciKGTeX becomes a integral part of every researcher's toolbox which makes semantic representations of contributions ubiquitous.

Bibliography

[1] Jaime Perales-Puchalt, Ryan Townley, Michelle Niedens, Eric D. Vidoni, Allen K. Greiner, Tahira Zufer, Tiffany Schwasinger-Schmidt, Jerrihlyn L. McGee, Hector Arreaza, and Jeffrey M. Burns. Acceptability and Preliminary Effectiveness of a Remote Dementia Educational Training among Healthcare Professionals. *medRxiv*, 2022.

[2] Rob Johnson, Michael Watkinson, Anthony Mabe, IJsbrand Jan Aalbersberg, Bev Acreman, Rick Anderson, Eric Archambault, Gaelle Bequet, Peter Berkery, et al. STM: International Association of Scientific, Technical and Medical Publishers Fifth Edition published. 2018.

[3] Alexander A. Aarts, Joanna E. Anderson, Christopher J. Anderson, Peter R. Attridge, Angela Attwood, Jordan Axt, Molly Babel, Štěpán Bahník, Erica Baranski, et al. Estimating the Reproducibility of Psychological Science. *Science*, 349, 8 2015.

[4] Monya Baker. 1,500 Scientists Lift the Lid on Reproducibility. *Nature*, 533:452–454, 2016.

[5] Sören Auer, Muhammad Harris, Markus Stocker, and Allard Oelen. https://www.orkg.org/orkg/smart-review/R135360, 6 2021.

[6] Arthur Brack, Anett Hoppe, Markus Stocker, Sören Auer, and Ralph Ewerth. Requirements Analysis for an Open Research Knowledge Graph. volume 12246 LNCS, pages 3–18. Springer, 2020.

[7] Bas B.L. Penning de Vries, Maarten van Smeden, Frits R. Rosendaal, and Rolf H.H. Groenwold. Title, Abstract, and Keyword Searching Resulted in Poor Recovery of Articles in Systematic Reviews of Epidemiologic Practice. *Journal of Clinical Epidemiology*, 121:55–61, 5 2020.

[8] Suzanne K. Linder, Geetanjali R. Kamath, Gregory F. Pratt, Smita S. Saraykar, and Robert J. Volk. Citation Searches Are More Sensitive Than Keyword Searches to Identify Studies Using Specific Measurement Instruments. *Journal of clinical epidemiology*, 68:412–417, 11 2015.

[9] Antrea Chrysanthou, Pinar Barlas, Kyriakos Kyriakou, Styliani Kleanthous, and Jahna Otterbacher. Bursting the Bubble: Tool for Awareness and Research about Overpersonalization in Information Access Systems. pages 112–113. Association for Computing Machinery, 3 2020.

[10] Engin Bozdag. Bias in Algorithmic Filtering and Personalization. *Ethics and Information Technology*, 15:209–227, 9 2013.

[11] Richard Van Noorden. Online Collaboration: Scientists and the Social Network. *Nature News*, 512:126–130, 8 2014.

[12] Sören Auer, Allard Oelen, Muhammad Haris, Markus Stocker, Jennifer D'Souza, Kheir Eddine Farfar, Lars Vogt, Manuel Prinz, Vitalis Wiens, and Mohamad Yaser Jaradeh. Improving Access to Scientific Literature with Knowledge Graphs. *Bibliothek Forschung und Praxis*, 44:516–529, 11 2020.

[13] Anna Fensel. Towards semantic APIs for research data services. *Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen & Bibliothekare*, 70:157–169, 2017.

[14] Allard Oelen, Mohamad Yaser Jaradeh, Kheir Eddine Farfar, Markus Stocker, and Sören Auer. Comparing Research Contributions in a Scholarly Knowledge Graph, 2019.

[15] Pavel Nikolaev, Daylond Hooper, Frederick Webber, Rahul Rao, Kevin Decker, Michael Krein, Jason Poleski, Rick Barto, and Benji Maruyama. Autonomy in Materials Research: A Case Study in Carbon Nanotube Growth. *npj Computational Materials*, 2, 10 2016.

[16] Christoph Lange. Enabling Collaboration on Semiformal Mathematical Knowledge by Semantic Web Integration, 1 2011.

[17] Dinh V. Cuong, Dac H. Nguyen, Son Huynh, Phong Huynh, C Gurrin, Minh-Son Dao, Duc-Tien Dang-Nguyen, and Binh T. Nguyen. A Framework for Paper Submission Recommendation System. *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 2020.

[18] Eric Medvet, Alberto Bartoli, and Giulio Piccinin. Publication Venue Recommendation Based on Paper Abstract. *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, pages 1004–1010, 2014.

[19] Jaana Takis, A. Q. M. Saiful Islam, Christoph Lange, and Sören Auer. Crowdsourced Semantic Annotation of Scientific Publications and Tabular Data in PDF. volume 16-17-September-2015, pages 1–8. Association for Computing Machinery, 9 2015.

[20] Victor R. Basili, Gianluigi Caldiera, and Dieter H. Rombach. The Goal Question Metric Approach, 1994.

[21] Mark D. Wilkinson. Comment: The FAIR Guiding Principles for Scientific Data Management and Stewardship. 2016.

[22] Ali Hasnain and Dietrich Rebholz-Schuhmann. Assessing FAIR data principles against the 5-star open data principles. pages 469–477, 2018.

[23] Clément Labadie, Christine Legner, Markus Eurich, and Martin Fadler. Fair enough? Enhancing the usage of enterprise data with data catalogs. volume 1, pages 201–210, 2020.

[24] Anna-Lena Lamprecht, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, Stephanie Van De Sandt, Jon Ison, et al. Towards FAIR principles for research software. *Data Science*, 3:37–59, 2020.

[25] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities, 2001.

[26] Nicola Guarino, Daniel Oberle, and Steffen Staab. What Is an Ontology? 2009.

[27] Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. *Knowledge Graphs*. Springer International Publishing, 2020.

[28] Graham Klyne, Jeremy Carrol, and Brian McBride. https://www.w3.org/TR/rdf11-concepts/, 11 2021.

[29] Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria Esther Vidal. Towards a Knowledge Graph for Science. Association for Computing Machinery, 2018.

[30] Simon Buckingham Shum, Enrico Motta, and John Domingue. ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse. *International Journal on Digital Libraries*, 3:237–248, 2000.

[31] Victoria Uren, Simon Buckingham Shum, Michelle Bachler, and Gangmin Li. Sensemaking Tools for Understanding Research Literatures: Design, Implementation and User Evaluation. *Int. J. Hum. Comput. Stud.*, 64:420–445, 2006.

[32] Karl Aberer, Alexey Boyarsky, Philippe Cudré-Mauroux, Gianluca Demartini, and Oleg Ruchayskiy. ScienceWISE: A Web-based Interactive Semantic Platform for Scientific Collaboration, 2011.

[33] Pavlos, Boland Katarina, Gasquet Malo, Zloch Matthäus, Zapilko Benjamin, Dietze Stefan, Todorov Konstantin Tchechmedjiev Andon, and Fafalios. ClaimsKG: A Knowledge Graph of Fact-Checked Claims. pages 309–324. Springer International Publishing, 2019.

[34] Tony Hammond, Michele Pasin, and Evangelos Theodoridis. Data integration and disintegration: Managing Springer Nature SciGraph with SHACL and OWL. 2017.

[35] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. pages 243–246. Association for Computing Machinery, Inc, 9 2019.

[36] Allard Oelen, Mohamad Yaser Jaradeh, Markus Stocker, and Sören Auer. Generate FAIR Literature Surveys with Scholarly Knowledge Graphs. pages 97–106. ACM, 8 2020.

[37] Jennifer D'souza and Sören Auer. Sentence, Phrase, and Triple Annotations to Build a Knowledge Graph of Natural Language Processing Contributions-A Trial Dataset. *Journal of Data and Information Science*, 0, 2021.

[38] Cristina-Iulia Bucur, Tobias Kuhn, and Davide Ceolin. A Unified Nanopublication Model for Effective and User-Friendly Access to the Elements of Scientific Publishing. 6 2020.

[39] Cristina-Iulia Bucur, Tobias Kuhn, Davide Ceolin, and Jacco van Ossenbruggen. Expressing High-Level Scientific Claims with Formal Semantics. pages 233–240. Association for Computing Machinery, 2021.

[40] Donald E. Knuth. *The TeXbook: a complete user's guide to computer typesetting with TEX*. Addison-Wesley, 1984.

[41] The LATEX Project. LATEX 2 for Class and Package Writers, 2006.

[42] Manuel Pégourié-Gonnard. A Guide to LuaLaTeX, 2013.

[43] Adobe Systems Incorporated. PDF Reference Fourth Edition Adobe® Portable Document Format, 2001.

[44] T S Perry. 'PostScript' Prints Anything: A Case History. *IEEE Spectrum*, 25:42–46, 1988.

[45] Adobe Systems Incorporated. *ISO 32000-2:2020(en) Document Management — Portable Document Format — Part 2: PDF 2.0*. 2 edition, 12 2020.

[46] Adobe Systems Incorporated. XMP Specification Part 3: Storage in Files, 2016.

[47] Tudor Groza, Knud Möller, Siegfried Handschuh, Diana Trif, and Stefan Decker. SALT: Weaving the Claim Web. volume 4825 LNCS, pages 197–210, 2007.

[48] Bernd Krieg-Brückner, Arne Lindow, Christoph Lüth, Achim Mahnke, and George Russell. Semantic Interrelation of Documents Via an Ontology. pages 271–282. Gesellschaft für Informatik e.V., 2004.

[49] Luc Moreau and Paul Groth. Provenance of Publications: A PROV Style for LaTeX. USENIX Association, 7 2015.

[50] Michael Kohlhase. Using LATEX As a Semantic Markup Format. *Mathematics in Computer Science*, 2:279–304, 12 2008.

[51] Almudena Ruiz-Iniesta and Oscar Corcho. A Review of Ontologies for Describing Scholarly and Scientific Document. volume 1155, 9 2014.

[52] Viet Bach Nguyen, Gollam Rabby, Vojtěch Svátek, and Oscar Corcho. Ontologies Supporting Research-related Information Foraging Using Knowledge Graphs: Literature Survey and Holistic Model Mapping. 9 2020.

[53] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. The Document Components Ontology (DoCO). *Semantic Web*, 7:167–181, 9 2016.

[54] Larisa Soldatova and Maria Liakata. An Ontology Methodology and CISP-the Proposed Core Information about Scientific Papers. 2007.

[55] Larisa N. Soldatova and Ross D. King. An Ontology of Scientific Experiments. *Journal of the Royal Society Interface*, 3:795–803, 12 2006.

[56] Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. Automatic Recognition of Conceptualization Zones in Scientific Articles and Two Life Science Applications. *Bioinformatics (Oxford, England)*, 28:991–1000, 9 2012.

[57] Paolo Ciccarese, Elizabeth Wu, Gwen Wong, Marco Ocana, June Kinoshita, Alan Ruttenberg, and Tim Clark. The SWAN Biomedical Discourse Ontology. *Journal of Biomedical Informatics*, 41:739–751, 10 2008.

[58] Fábio R. Assis Neto and Celso A. S. Santos. Understanding Crowdsourcing Projects: A Systematic Review of Tendencies, Workflow, and Quality Management. *Information Processing & Management*, 54:490–506, 2018.

[59] Mahmood Hosseini, Alimohammad Shahri, Keith Phalp, Jacqui Taylor, and Raian Ali. Crowdsourcing: A Taxonomy and Systematic Mapping Study. *Computer Science Review*, 17:43–69, 2015.

[60] Oliver Karras, Eduard C. Groen, Javed Ali Khan, and Sören Auer. Researcher or Crowd Member? Why not both! The Open Research Knowledge Graph for Applying and Communicating CrowdRE Research. 8 2021.

[61] Vojtěch Svátek and Ondře Sváb. Towards Retrieving Scholarly Literature via Ontological Relationships. 9 2012.

[62] Allard Oelen, Markus Stocker, and Sören Auer. Crowdsourcing Scholarly Discourse Annotations. pages 464–474. Association for Computing Machinery, 4 2021.

[63] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, et al. Manifesto for Agile Software Development, 2001.

[64] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison Wesley Longman Publishing Co., Inc., 2004.

[65] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E. M. Van der Werf, and Sjaak Brinkkemper. The Use and Effectiveness of User Stories in Practice. pages 205–222. Springer International Publishing, 2016.

[66] Alina-Madalina Gheorghe, Ileana Gheorghe, and Ioana Iatan. Agile Software Development. *Informatica Economica*, 24:90–100, 1 2020.

[67] Francesco Casillo, Vincenzo Deufemia, and Carmine Gravino. Detecting Privacy Requirements From User Stories with NLP Transfer Learning Models. *Information and Software Technology*, 146:106853, 2022.

[68] John Brooke. SUS: A Quick and Dirty Usability Scale. *Usability Eval. Ind.*, 189, 4 1995.

[69] John W. Ratcliff and David E. Metzener. Pattern Matching: The Gestalt Approach. *Dr Dobbs Journal*, 13:46, 1988.

[70] Joseph L. Fleiss. Measuring Nominal Scale Agreement among Many Raters. *Psychological bulletin*, 76:378, 1971.

[71] Aaron Bangor, Philip T. Kortum, and James T. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies archive*, 4:114–123, 2009.

[72] Richard J. Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33:159–174, 1977.

A Appendix

In the appendix I provide all the materials used and produced in this master thesis. These include the open source software repository of SciKGTeX on Github and the materials from the user evaluation.

A.1 Github Repository

The github repository can be found here: `https://github.com/Christof93/SciKGTeX`. Further information is provided in the README.md file in the project folder.

A.2 Evaluation Materials & Results

This persistent Zenodo DOI leads to the materials and results from the user evaluation: `https://doi.org/10.5281/zenodo.6518995`. I will introduce all the different files which can be found through this link in the following.

Survey Questionnaire The file `SciKGTeX_Evaluation_Questionnaire.pdf` contains the PDF version of the Google Form which was used as a Questionnaire for the user evaluation.

Analysis The file `analysis.ipynb` contains an interactive Python notebook where all the calculations for metrics (see: section 6.1.5) and statistical tests were performed. The notebook also features some commentary on the calculations.

Task Material The file `raw_text_eval_task.txt` contains the raw text of the abstract presented to the evaluation participants for the evaluation tasks (see section 6.1.3).

Responses The file `responses.csv` contains all the responses to the questions of the survey questionnaire. The questions are columns while rows correspond to samples.

Results The file `results.tsv` contains the responses of the questionnaire plus all the other results such as the time delta values for task 1, the calculated SUS scores, the LaTeX score, Semantic Web score, the Fleiss kappa inter annotator agreement

values (column named Krippendorf alpha), the Gestalt similarity score and the actual text values of the annotations per property and participant.