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

Anonymous Author(s)

ABSTRACT

Scientific knowledge graphs have been proposed as a solution to structure the content of research publications in a machine-actionable way and enable more efficient, computer-assisted workflows for many research activities. Crowd-sourcing approaches are used frequently to build and maintain such scientific knowledge graphs. 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 paper, we present a workflow for authors of scientific documents to specify their contributions with a LaTeX package, called SciKGTeX, and upload them to a scientific knowledge graph. The SciKGTeX package allows authors of scientific publications to mark the main contributions of their work directly in LaTeX source files. The package embeds marked contributions as metadata into the generated PDF document, from where they can be extracted automatically and imported into a scientific knowledge graph, such as the ORKG. This workflow is simpler and faster than current approaches, which make use of external web interfaces for data entry. Our user evaluation shows that SciKGTeX is easy to use, with a score of 79 out of 100 on the System Usability Scale, as participants of the study needed only 7 minutes on average to annotate the main contributions on a sample abstract of a published paper. Further testing shows that the embedded contributions can be successfully uploaded to ORKG within ten seconds. SciKGTeX simplifies the process of manual semantic annotation of research contributions in scientific articles. Our workflow demonstrates how a scientific knowledge graph can automatically ingest research contributions from document metadata.

CCS CONCEPTS

• **Applied computing** → **Markup languages**; **Document metadata**; **Annotation**.

KEYWORDS

Semantic Annotation, LaTeX, FAIR data, Scientific Knowledge Graphs.

ACM Reference Format:

Anonymous Author(s). 2023. SciKGTeX - A LaTeX Package to Semantically Annotate Contributions in Scientific Publications. In *Proceedings of ACM/IEEE Joint Conference On Digital Libraries (JCDL 2023).* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/xx.xxx/xxx_x

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

JCDL 2023, June 26 - 30, 2023, Santa Fe, New Mexico, USA

© 2023 Association for Computing Machinery.

ACM ISBN 978-1-4503-8713-2/22/04…$15.00
https://doi.org/xx.xxx/xxx_x

REMARK FOR REVIEWERS

The authors of this paper have already made publicly available the developed components and other supplementary materials, such as the LaTeX package or the evaluation questionnaire and data. For this reason, we cannot guarantee that reviewers will not determine the identities of the authors if they **actively** search for SciKGTeX themselves. We have made all necessary materials available for reviewers in anonymous GitHub repositories: https://anonymous.4open.science/r/SciKGTeX-5B74 and https://anonymous.4open.science/r/PDF2ORKG-5072. In case of acceptance of the paper, we will add all links and references to the already published components and materials.

1 INTRODUCTION

Scientific discoveries have long become a community effort, with sometimes hundreds of researchers from different institutions collaborating on 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. Nowadays, the standard process in scientific communication is to publish scientific articles which are archived and distributed as PDF files [17]. This is a basic approach to digitising research content and does not leverage modern technologies which could pave the way to computer-assisted knowledge exchange. 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 [1, 4] and quality of peer reviews have been stagnating [2].

As a possible solution to this problem, scientific knowledge graphs, such as the Open Research Knowledge Graph[1] (ORKG) [3, 30], have been developed [29]. Scientific knowledge graphs represent research content in a graph network of relations and concepts which allows more sophisticated methods of information extraction. Unlike raw text, 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 [12], automatic literature reviews [27], reasoning engines, autonomous research systems [26], mathematical proof assistants [21] and paper recommendation systems [11, 24].

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 extracted into the knowledge graph. A common strategy to achieve annotation on a large scale is through crowd-sourcing [31]. In the case of the ORKG for example crowd-sourcing is realised through an annotation tool provided on a web platform [18]. The problem with this approach is that the annotation

[1] https://orkg.org/

of concepts and relations is a laborious task which discourages potential users. Scientific contributions are already often fragmented across a multitude of platforms such as dataset repositories, preprint websites, postprint discussion threads and video platforms. Adding yet another platform to this mix distracts the researchers from their main objectives and complicates their workflow. To incentivise researchers contributing to scientific knowledge graphs, the annotation of metadata needs to be integrated seamlessly into the scientific process. Figure 1 illustrates the complexity problem of the current workflow (red). Furthermore, the metadata which the researchers generate should not end up in a data silo and possibly vanish with time.

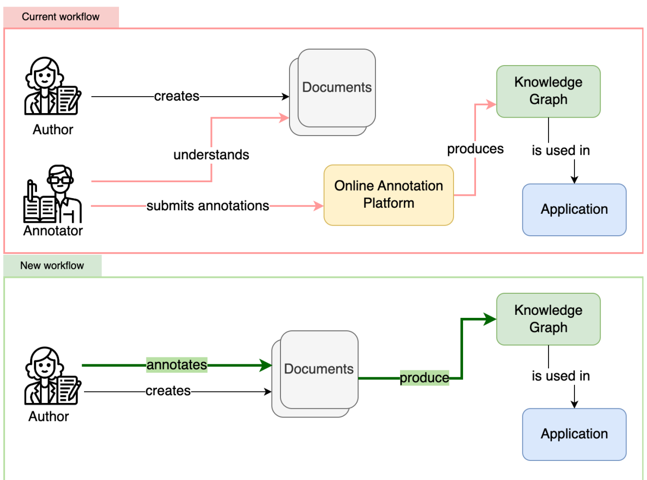

Figure 1: Our proposed solution simplifies the crowd-sourcing approach of scientific knowledge graphs.

We propose a new solution which utilises commonly used systems and technologies in a researcher's toolbox. The gain in simplicity which can be achieved by our approach can be seen in Figure 1 in the workflow highlighted in green. A part of the complexity of the current process is simplified by treating the annotation of metadata and the creation of the document as a common step. Our idea is to find a solution which does not rely on an online annotation platform or even a third-party annotator to submit the contributions. To develop the solution, we orient ourselves on the following two research questions (RQs):

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

RQ 2: How can a scientific knowledge graph automatically ingest research contributions from document metadata?

To clarify RQ 1, we develop the SciKGTeX LaTeX package and conduct a comprehensive user evaluation which highlights our solution's simplicity and usability. RQ 2 is about the import of PDF metadata to a scientific knowledge graph. We investigate the practicability of ingesting XMP metadata from PDFs into the ORKG.

In this paper, we present a complete workflow of solutions for annotating, embedding, extracting, and importing structured research contributions from scientific documents typed in LaTeX. The workflow includes three components: i) the SciKGTeX LaTeX package with markup capabilities for contributions, ii) the LuaLaTeX PDF compiler, and iii) the PDF2ORKG import module, where SciKGTeX, and PDF2ORKG are the original contributions of this work.

The outline of the paper is the following: in Section 2 we overview the related work on semantic annotation in LaTeX. In Section 3, we describe our approach through user stories and functionalities. Section 4 provides details on the implementation of the SciKGTeX package, and PDF2ORKG import module. Section 5 describes the user evaluation which we conducted on the system and reports the results of the user evaluation. Finally, in Sections 6 and 7 we discuss the research questions and further implications of our work and draw the final conclusion.

2 RELATED WORK

Some of the first documented attempts to solve the problem of semantic metadata specification in LaTeX were made in 2007 by Groza et al. [14] who 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, albeit the use case is slightly different. They concentrate primarily on the generation of HTML content from the annotated PDF to support the automatic creation of online proceedings but do not explore other use cases. Moreover, SALT is not maintained anymore and can not be used at the time of writing.

Moreau et al. [25] 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. 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 but fails to embed the data itself into the PDF document. Another semantic annotation markup was developed by Michael Kohlhase [19] to turn LaTeX into a document format for mathematical knowledge management (MKM).

Most recently, Martin & Henrich [23] worked on a similar objective of linking LaTeX publications with scientific knowledge graphs. They implemented the RDFTeX framework which enables importing existing contributions from scientific knowledge graphs to LaTeX and exporting new contributions in RDF format. However, the authors do not consider embedding the contribution into the PDF file and rather store them in an additional RDF document which is detrimental to the persistence of the metadata. On top of that, they do not show the exchange with an existing scientific knowledge graph and only describe it in theory. We extend their work by providing a refined LaTeX package which strives to be more intuitive for regular scientific writers as it does not require the additional step of preprocessing the LaTeX files with a Python script. Furthermore, we contribute a first user evaluation to back our usability claims and set a benchmark for other similar tools.

Another related solution crowd-sourcing metadata annotation is the PDF annotation tool for the ORKG which is described by Oelen et al. [28]. This tool can be used through a graphical user interface on the ORKG. While this approach seems viable for published papers, we 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. Moreover, with our proposed workflow, it is not necessary for the authors to navigate to an annotation website and leave the usual LaTeX working environment. This resonates with the results of our evaluation which shows that SciKGTeX scores higher on a standardised usability test (see Section 5.3).

In general, it should be noted that most metadata initiatives concentrate on bibliographic metadata and do not provide ways to encode machine-actionable representations of the actual scientific content of documents. Our solution serves as a practical implementation of content-related metadata specification and storage which is easier to use and adopt than previous approaches.

3 APPROACH

For the development of the scientific knowledge graph annotation workflow described in Section 1, we followed an agile development approach [6]. The first step in our approach consisted of assessing the requirements that users have for the tool from a variety of possible use cases which were determined from conversations with researchers. The requirements are expressed as user stories [9] following the *Connextra* template [22]. In the *Connextra* template, a user story has 3 slots – a role, a requirement and an optional reason. The slots are connected into a sentence:

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

The different user stories are organised into three roles which were identified – the researcher, the publication provider, and the scientific knowledge graph user. The notion of researcher specifically stands for an author of a scientific publication here. A publication provider denotes an entity or organization which collects and curates 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*[2]. A scientific knowledge graph user operates with the structured and machine-actionable representation of scientific contributions provided by platforms such as the ORKG. For example, it can be a literature review author who intends to use automated comparison platforms to supplement the creation of their review.

Afterward, we came up with specific functionalities which address the user stories. These functionalities were then implemented in the first prototype. The lists of stories, functionalities, and their relationships are represented in Figure 2.

We also adopted the principle of iterative development cycles from the agile approach. This means that after implementing the functionalities, we evaluated the resulting system either through a review process or the user evaluations described in Section 5. Then, the cycle restarts with the definition of new user stories gained from the review process.

4 IMPLEMENTATION

The proposed pipeline consists of two separate solutions: i) the SciKGTeX package for LaTeX, and ii) the PDF2ORKG import module to showcase ingestion of the metadata into a centralised scientific knowledge graph.

4.1 SciKGTeX Package

To implement the specified functionalities (Fig. 2: Functionalities 1–7), we developed the SciKGTeX package for LaTeX. LaTeX is a popular tool for the creation of scientific publications. Compared to alternatives like Microsoft Word, LaTeX features a whole ecosystem of open-source extensions which are built by an active community. Many publishers recommend writing scientific papers with LaTeX due to the possibility to supply extensive templates for journals or conference proceedings and get consistent output.

Extension packages for the LaTeX type-setting system are freely distributed over the internet and can be built by anyone with the technical knowledge to do so. 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 require a larger development overhead.

Interfering with the standard PDF generation engine (pdfTeX) is not trivial and an extensive task as it is implemented in the TeX typesetting system, which has many peculiar idiosyncrasies making it very time-consuming to develop new features. There exists an alternative compiler for LaTeX called LuaTeX[3] which features the embedded Lua scripting language and callback hooks to the most important events in the PDF generation process. Writing the package in LuaTeX allowed to keep the development effort comparatively low and implement the desired functionality quickly. Implementations for other TeX to PDF compilers are of course still possible in the future if compatibility problems arise.

The developed package SciKGTeX is available on the TeX package archive CTAN[4] and Github[5]. Integrating the SciKGTeX functionality into a project can be achieved by downloading the package and putting `\usepackage{scikgtex}` into the document preamble. To illustrate the configuration of the SciKGTeX package, there is a demo project in the Overleaf service[6]. For the package to work, it is necessary to compile the LaTeX source with LuaLaTeX. In the following, we expand on how we implemented the main functionalities depicted in Figure 2 as a LaTeX package.

Functionality 1. Assigning properties to textual entities is implemented by defining new LaTeX commands which can be used to mark expressions in the document. Five commands were reserved for the most important properties describing a scientific contribution: research problem, objective, method, result, and conclusion. These command names were chosen from the DEO classes (see [10]) as suggested by the approach of Oelen et al. [28]. The DEO class of *research statement* was adapted to the *research problem* property, which is a central concept of the ORKG vocabulary. A

[2]https://arxiv.org/

[3]https://www.luatex.org/
[4]Link will be added upon publication.
[5]https://anonymous.4open.science/r/SciKGTeX-5B74
[6]This link will be added on publication.

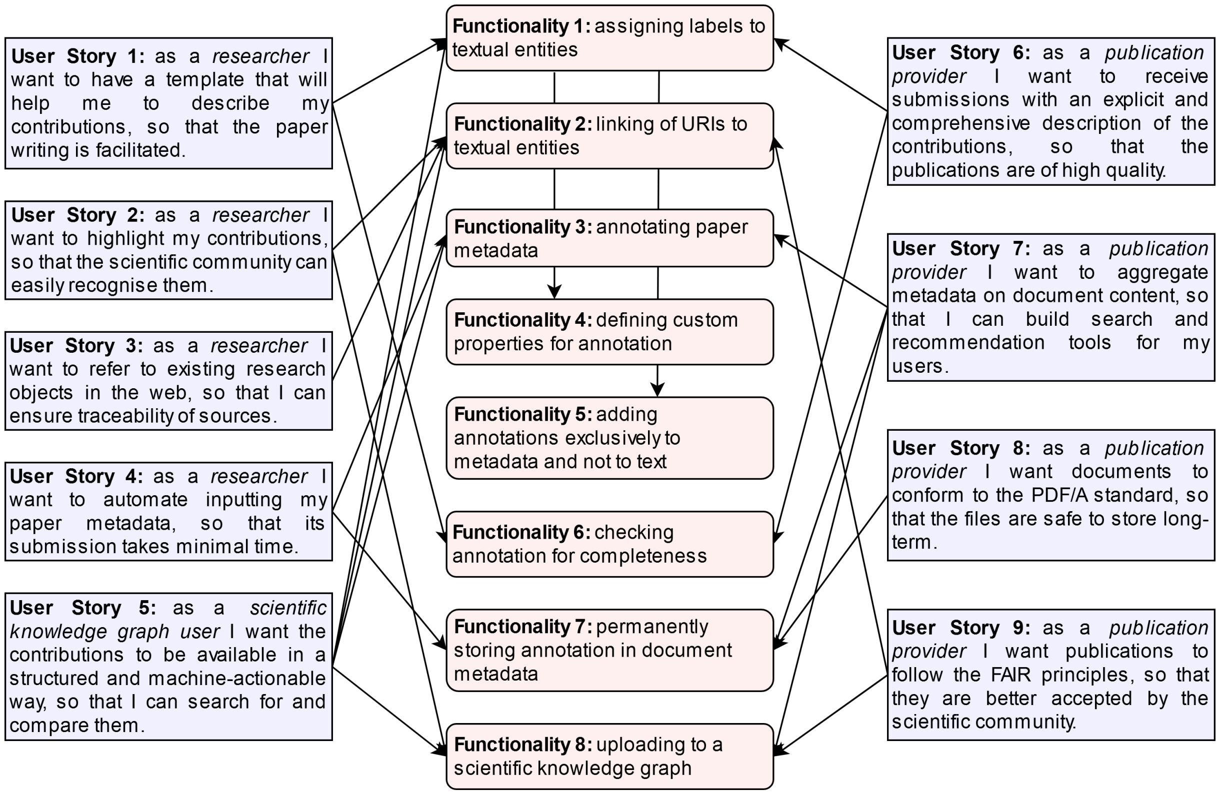

Figure 2: User stories and functionalities

new property *objective* was introduced to fulfil a requirement identified in the first round of user evaluations. From this we derive the five predefined commands in SciKGTeX: `\researchproblem{}`, `\objective{}`, `\method{}`, `\result{}`, and `\conclusion{}`.

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 distinguishing the contributions. The optional argument can be any identifier, but is most intuitively understood as an enumeration. For example, annotations `\researchproblem[1]{..}` and `\researchproblem[2]{..}` add two separate contributions with respective research problems. If two contributions have a property in common, the property can be assigned to the two contributions using a comma between the arguments, for example, `\method[1,2]{..}`. This would mean that there are two distinct contributions with the respective research problems which share the same method.

Additional to the 5 mandatory ones, it is also possible to specify other properties with the `contribution` command. These other properties can be any arbitrary string in theory but are especially valuable if common interesting properties of scientific subdomains are used. For example, properties of p-value or accuracy are useful for studies that include statistical examinations and can be attached to a contribution with `\contribution{p-value}{0.05}` and `\contribution{accuracy}{0.876}`. In the metadata these properties will be created as extensions to the ORKG ontology. If an author wants to reuse a property from another specific ontology this can be achieved with the commands detailed in functionality 4.

Functionality 2. Linking of URIs to textual entities is implemented with the `\uri{}` command placed inside an annotation. This takes the URI of an entity defined in the web as the first argument and an optional label as the second, see Listing 1. If a label is given, it is rendered as a hyperlink to the URI, see Figure 3.

Listing 1: Entity linking

```
The role of \researchproblem{\uri{https://www.orkg
    .org/orkg/resource/R12259}{antibiotic therapy
    }} in managing acute bacterial sinusitis (ABS)
     in children is controversial...
```

Figure 3: Entity linking rendering

Functionality 3. Annotation of bibliographic metadata is implemented with the commands `\metatitle`, `\metaauthor` and

`\researchfield` for the title, authors, and research field respectively, see Listing 2. The `meta` prefix is added to the `title` and `author` commands to not overwrite the existing commands in LaTeX.

Listing 2: Annotated bibliographic metadata

```
\title{\metatitle{Effectiveness of Amoxicillin/
    Clavulanate Potassium in the Treatment of
    Acute Bacterial Sinusitis in Children.}}
\author{\metaauthor{Ellen R. Wald} \and \
    metaauthor{David Nash} \metaauthor{Jens
    Eickhoff}}
\researchfield{pharmacology}
```

Functionality 4. The definition of custom properties for annotation is achieved by SciKGTeX with the `\addmetaproperty` command. This command registers a new namespace for the metadata, which can be given as a first parameter to the command. An abbreviation of this namespace can also be specified and used as a prefix for the annotation. This is useful if ambiguous properties from different ontologies are used. Listing 3 shows an example.

Listing 3: Custom properties

```
\addmetaproperty[amo, http://purl.org/spar/amo#]{
    has_claim}
\addmetaproperty[patent, https://other.type/of/
    ontology]{has_claim}
...
\contribution{amo:has_claim}{The earth is round}.
Our patent has the following claim:
\contribution{patent:has_claim}{An apparatus to
    achieve something new.}...
```

Functionality 5. The annotation of contributions without rendering them into the document text is implemented with the starred variant of the property commands. For example, having the sentence 'the p-value was 0.01% higher than in the earlier experiment', it may be desirable to report the actual p-value in the metadata instead of the relative change. In such a case, the command can be simply marked with a star (see Listing 4). In the rendered sentence, the content of the starred property (0.06) will be invisible.

Listing 4: Invisible markup

```
... the p-value was 0.01\% higher \contribution*{p
    -value}{0.06} than in the earlier experiment
    ...
```

Functionality 6. The completeness check of the mandatory metadata properties is implemented through compiler warnings. If any of the five mandatory commands are missing, the user gets a warning in the console. Figure 4 provides an example of the warning in the Overleaf interface.

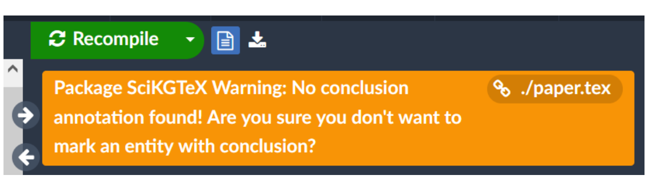

Figure 4: Example of SciKGTeX warning in Overleaf

Functionality 7. The permanent storage of the annotation is implemented by adding it to the PDF XMP metadata. In this way, the metadata is merged with the document and can be retrieved by anyone who obtains the PDF. We used SciKGTeX to encode the main contribution of this paper. However, EasyChair removes all metadata from uploaded PDFs presumably to ensure the double-blind review process. For this reason, we also provide a corresponding XMP metadata snippet[7] for inspection of the created metadata. While it is trivial to extract the metadata programmatically from the PDF, manual inspection is not as straight forward, as most PDF viewers are not capable of displaying an arbitrary metadata stream embedded in the document.

4.2 PDF2ORKG Import Module

Functionality 8. Having a paper annotated with SciKGTeX, it is possible to upload the annotation to the ORKG automatically. This is implemented with the PDF2ORKG import module[8]. The module is written in Python and utilises the Python ORKG API[9]. This module can be integrated into the ORKG interface or into an interface of the paper submission system to automate the ingestion process.

To create a paper instance in the ORKG you typically need to specify properties such as DOI, title, authors, publication date, publisher, research field, and contributions. At the moment of annotating the paper in LaTeX, DOI, publication date, and venue are unknown, while other properties (title, authors, research field, and contributions) are annotated with SciKGTeX. Table 1 demonstrates the mapping between the ORKG properties and the SciKGTeX annotation.

Table 1: Mapping of ORKG properties to SciKGTeX annotations

ORKG property	SciKGTeX command
DOI	—
Title	`\metatitle`
Authors	`\metaauthor`
Publication date	—
Published in	—
Research field	`\researchfield`
Contributions	`\contribution`

The dataflow of the module for importing the SciKGTeX annotations from PDF files to ORKG is represented in Figure 5. First, the PDF file is read and its metadata are extracted in XML format. Then, the module performs HTTP requests to the ORKG API to find the appropriate URIs for the annotated entities and properties. After obtaining all relevant data, a JSON string is formed and passed to the ORKG API method that creates or updates an instance of the paper. An example of the JSON data is available on the Github repository[10].

[7] https://anonymous.4open.science/r/SciKGTeX-5B74/publication/output.xmp_metadata.xml
[8] https://anonymous.4open.science/r/PDF2ORKG-5072
[9] https://orkg.readthedocs.io/en/latest/index.html
[10] https://anonymous.4open.science/r/PDF2ORKG-5072/metadata.json

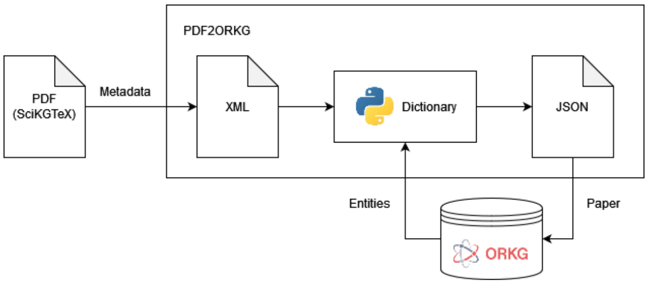

Figure 5: PDF2ORKG dataflow

5 EVALUATION

The goal of the evaluation for SciKGTeX was to test the usability of the approach with potential users who are given a small series of tasks to complete using the package. The evaluation serves to collect a number of metrics which indicate the current usability, and convenience of the developed solution and provide a baseline for further development of the package in the future. The metrics are designed to reveal more about research question 1. We transformed the research question into three verifiable hypotheses (H1 - H3) which we planned to test in the evaluation.

H1: The system is easy and convenient to use.
H2: Annotation of main contributions in a short text summary can be performed in less than 10 minutes.
H3: Different annotators produce similar annotations.

We explored H1 by assessing the perceived usability by the participants of the evaluation. We specifically took into consideration the System Usability Scale (see section 5.2) and feedback from the participants after the evaluation. H2 sheds light on the simplicity of the annotation tool. 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 H3 can be verified, it indicates that the tool is able to produce consistent metadata from different users on different documents. This is a desirable outcome of the tool since consistent metadata brings many advantages for downstream applications of the data.

5.1 Evaluation Setup

For the evaluation, 26 volunteers were recruited from a range of different universities and institutes mainly in the broader domain of computer science. 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 remaining 8 participants were working in different research-related positions.

We determined the participants' prior knowledge in LaTeX and semantic web technologies before running the evaluation with a specific set of questions. The exact questionnaire can be found in the experiment artefacts[11]. The familiarity with LaTeX was high among the participants which adds to the claim 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 on a scale from 0 to 100.

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

Evaluation Procedure

(1) Give the participant approximately 5 min to read the SciKGTeX documentation[12] 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 the participant complete them.
(5) Let the participant fill in the survey questionnaire.
(6) Let the participant give additional oral feedback.

The three tasks we gave to the participants were the following:

Evaluation Tasks

(1) Annotate the 5 properties of the main contribution (background[13], 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.

5.2 Evaluation Metrics

To investigate the hypotheses we measured three independent variables for each participant in the evaluation setup: (i) the System Usability Scale (SUS) score for H1, (ii) the time to finish the first evaluation task in seconds for H2 and (iii) the Fleiss kappa [13] as a measure of agreement between annotators. The SUS is widely used to measure the usability of software systems [7]. It is calculated from the user rating of a predefined collection of 10 statements about the user experience of the system. The Fleiss kappa is a well-known measure to determine the agreement between annotators also known as inter-annotator agreement or inter-rater reliability. All calculations of metrics can be found in the Python notebook in the experiment artefacts[14]. Research question 2 is answered by a technical testing of the PDF2ORKG import module. The goal of this testing was to check that a paper and its annotation are successfully uploaded to the ORKG, and to measure the time required for executing the upload scripts. For the test, a scientific paper [15] was annotated with SciKGTeX. Two uploading modes were considered: adding a new paper and updating an existing one.

[11] https://anonymous.4open.science/r/SciKGTeX_experiment_artifacts-0E93/SciKGTeX_Evaluation_Questionnaire.pdf

[12] The documentation is provided on the Github repository.
[13] Background is an old property from the development of the first version of the LaTeX package used for the evaluation. Based on the results of the evaluation, we have revised the 5 predefined properties by removing the background property and adding the new property objective (see Section 4.1).
[14] https://anonymous.4open.science/r/SciKGTeX_experiment_artifacts-0E93/analysis.ipynb

5.3 Results

In the following section, we present the results of the evaluation. We provide all the resulting data including the measured values and user annotations in the experiment artefacts[15].

System Usability Scale. For interpretation of the SUS score, we rely on the work of Bangor et al. [5] who mapped the percentage-based usability scale to a 7-level adjective scale comprised of 'Worst Imaginable', 'Bad', 'Awful', 'Poor', 'OK', 'Good', 'Excellent' and 'Best Imaginable'. For the mapping they ran 212 SUS surveys and asked the adjective ratings alongside the user test. They found that 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 amounts to 79.8 ($\sigma = 11.6$). The second and third quartiles 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 among the participants (see Figure 6), it can be observed that PhD students rated a slightly higher mean SUS score than the other groups at 82.3 ($\sigma = 12.34$). Compared to the PDF annotation tool by Oelen et al. [28] (see Section 2), SciKGTeX scores 3 points higher with a similar sample size (23 vs. 26) and standard deviation (11.3 vs. 11.6).

Furthermore, we analysed the correlation between prior knowledge and the usability score outcome. To explore the correlation, we calculated Pearson's correlation coefficient between the variables 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 makes the package easy and convenient to use as was hypothesised in hypothesis 1.

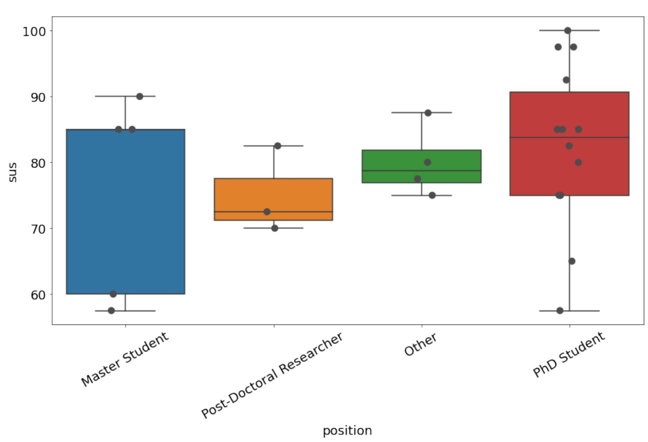

Figure 6: System Usability Scale distributions among the different groups of participants.

Annotation Time. The variable of time spent on the annotation of the main contribution gives an indication of usability as it proves or disproves that the idea of the package can be grasped in little time by typical users. The mean duration of work on task 1 in the evaluation is 7 Minutes 34 seconds at a standard deviation of 3 minutes 21 seconds. This is under our previously defined threshold of 10 minutes and shows that the package can be applied quickly without extensive prior knowledge.

Inter Annotator Agreement. Table 2 contains the results of the Fleiss kappa inter-annotator agreement values on each of the five property annotations of the main contribution. It becomes apparent that the first three properties *background*[16], *research problem* and *method* get far less consistent annotations than the *result* and *conclusion* annotations. According to Landis et al. [20] 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 *conclusion* they can be considered fairly consistent. For the other three categories, there are big differences in the text passages 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. This led us to redefine the *Background* command to *Objective* in the subsequent development cycle of the package as a consequence of the evaluation and user feedback. The *method* annotation is often split into several annotations of sentence fragments which mention methods but sometimes the *method* is tagged as a whole block of text.

Table 2: Inter-annotator agreement for the different categories of annotations in the LaTeX package.

Annotation Type	Fleiss kappa
Background	0.21
Research Problem	0.44
Method	0.24
Result	0.74
Conclusion	0.81

Upload Time. When testing the PDF2ORKG module, a paper instance has been successfully uploaded to the ORKG[17] (see Figure 7). The measurements are provided in Table 3. It contains the time consumption for the two scenarios, and for the three main steps of the PDF2ORKG workflow. The measurements show that the running time is within tens of seconds, the main amount of time is spent on uploading data to ORKG. Expectedly, creating a new paper takes longer than updating an existing one.

5.4 Threats to Validity

While the user evaluation is designed to model an actual use case as closely as possible, there are some abstractions which differentiate it from real-world usage. These must be considered threats to the validity of the experiment. One such limitation is that the

[15] https://anonymous.4open.science/r/SciKGTeX_experiment_artifacts-0E93/results.tsv

[16] Background is an old property from the development of the first version of the LaTeX package used for the evaluation. Based on the results of the evaluation, we have revised the 5 predefined properties by removing the background property and adding the new property objective (see Section 4.1).

[17] https://sandbox.orkg.org/paper/R258002

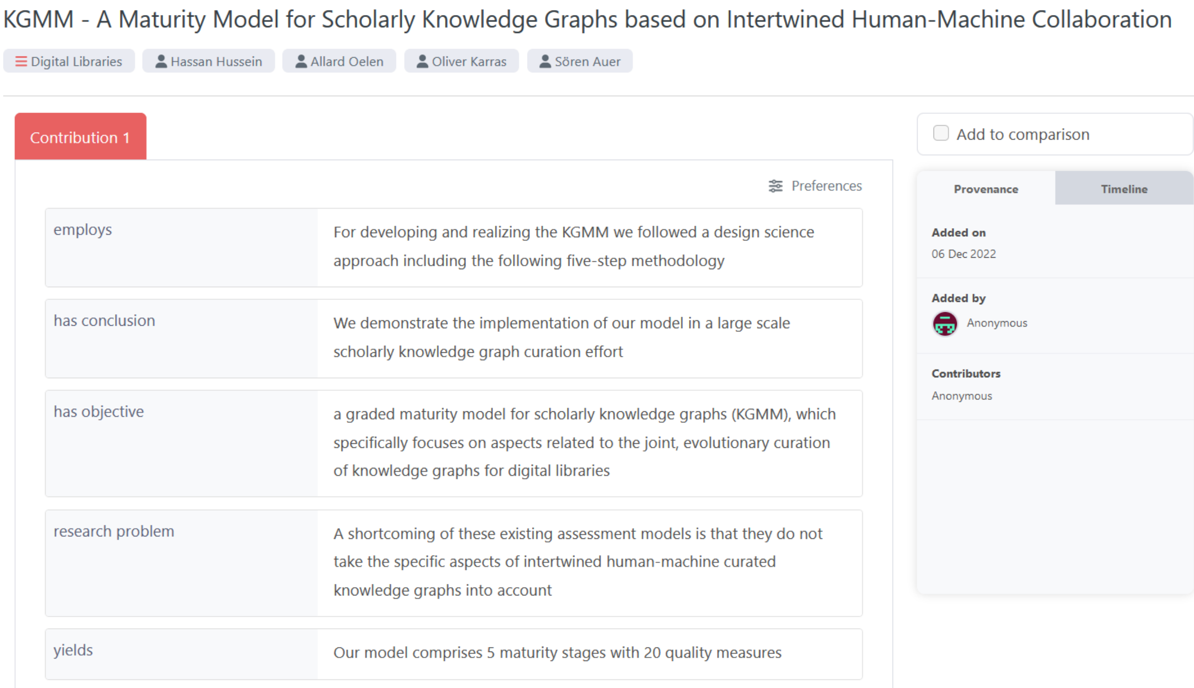

Figure 7: A paper imported to ORKG

Table 3: Time measurements for PDF2ORKG

Step	Adding paper, s	Updating paper, s
Reading and extracting PDF metadata	0.01	0.01
Requesting IDs from ORKG	0.07	0.07
Uploading data to ORKG	31.3	17.07
Total	***31.38***	***17.15***

text[18] which was used as a base for the evaluation tasks was not a document authored by the participants themselves. Specifically, we chose a randomly selected paper on the topic of Requirements Engineering [8]. This topic was selected as most of the candidates which volunteered for the evaluation had at least some background in it. 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 self-authorship as closely as possible. Nonetheless, the fact that the participants are not actually the authors of the text that they annotate in this test compromises the validity of the experiment. It was not possible to let the participants work on their own texts because the given-above hypotheses could only be evaluated by measuring comparable values in the independent variables. In order to be comparable, all participants had to work on the same underlying text.

On the choice of hypotheses, it must be noted that H1 is the only statement which concerns just the technical implementation and functionality of the LaTeX package itself rather than the broader task of metadata annotation. Accordingly, hypotheses 2 and 3 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).

6 DISCUSSION

In this section, we revisit the research questions RQ 1 and RQ 2 from section 1 and discuss the implications of our research.

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 we 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, we 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.

Different from comparable metadata annotation solutions [28], [16], SciKGTeX is not reliant on any systems other than the LuaTeX

[18] The actual text used in the test can be found in the experiment artefacts: https://anonymous.4open.science/r/SciKGTeX_experiment_artifacts-0E93/raw_text_eval_task.txt

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.

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 resulting metadata is machine-actionable and can be used to build large knowledge bases which facilitate various applications from which the researchers can benefit 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 (i) extending the package documentation with clearer examples and (ii) introducing new properties which leave less space for differing interpretations.

RQ 2. How can a scientific knowledge graph automatically ingest research contributions from document metadata?

With the PDF2ORKG import module, we demonstrate a possible implementation of the metadata upload to a centralised knowledge graph. The PDF2ORKG import module is currently rather a proof of concept. However, in the future, it could be integrated into Overleaf, the ORKG interface, conference submission pages, or paper submission systems.

Another issue with using PDF2ORKG is that at the moment of annotating a paper with SciKGTeX, there is no DOI, publication date and publisher specified. The most straightforward solution to this is to add the missing data manually after the paper is published.

Another direction for future work can be integrating ORKG API calls directly into SciKGTeX. It would provide the possibility for feedback from the ORKG to an author via the warnings inside the LaTeX editor being used. This feedback can be related to, for example, the ORKG entities, recognised in the annotated text.

7 CONCLUSION

In this paper, we propose a solution for authors of scientific publications to annotate machine-actionable metadata about the contributions of their publications at the time of writing the manuscript. The semantic annotations serve to build scientific knowledge graphs, which constitute the future digital record of scholarly publications. Different from previous solutions, we present a LaTeX package called SciKGTeX which allows directly specifying the metadata at the time of document creation and embedding them into the resulting PDF file. The metadata can be automatically extracted from the PDF file and uploaded to a scientific knowledge graph, such as the ORKG. This is a simplification compared to an approach where the metadata specification is handled through a separate web interface.

In essence, SciKGTeX is a successful implementation of an annotation framework for metadata of scientific contributions. It is arguably simpler than other approaches with the same objective such as [28] or [23] 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. Additionally, we have presented an implementation of extracting the metadata from the PDF file and importing it to the ORKG. Further benefits of the approach are 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 and to facilitate the development of supportive applications which elevate the modern research workflow to better standards.

Future 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 by established publishers to define custom metadata templates which can be used for journals, conference proceedings or other use cases. These templates then allow the aggregation of consistent metadata on papers from the same research area.

REFERENCES

[1] Alexander A. Aarts, Joanna E. Anderson, Christopher J. Anderson, Peter R. Attridge, Angela Attwood, et al. 2015. Estimating the Reproducibility of Psychological Science. *Science* 349 (8 2015). Issue 6251. https://doi.org/10.1126/science.aac4716

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

[3] Sören Auer, Allard Oelen, Muhammad Haris, Markus Stocker, Jennifer D'Souza, Kheir Eddine Farfar, Lars Vogt, Manuel Prinz, Vitalis Wiens, and Mohamad Yaser Jaradeh. 2020. Improving Access to Scientific Literature with Knowledge Graphs. *Bibliothek Forschung und Praxis* 44 (11 2020), 516–529. Issue 3. https://doi.org/10.1515/bfp-2020-2042

[4] Monya Baker. 2016. 1,500 Scientists Lift the Lid on Reproducibility. *Nature* 533 (2016), 452–454. Issue 7604. https://doi.org/10.1038/533452a

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

[6] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. 2001. Manifesto for Agile Software Development. http://www.agilemanifesto.org/

[7] John Brooke. 1995. SUS: A Quick and Dirty Usability Scale. *Usability Eval. Ind.* 189 (4 1995).

[8] Francesco Casillo, Vincenzo Deufemia, and Carmine Gravino. 2022. Detecting Privacy Requirements From User Stories with NLP Transfer Learning Models. *Information and Software Technology* 146 (2022), 106853. https://doi.org/10.1016/j.infsof.2022.106853

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

[10] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. 2016. The Document Components Ontology (DoCO). *Semantic Web* 7 (9 2016), 167–181. https://doi.org/10.3233/SW-150177

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

[12] Anna Fensel. 2017. Towards semantic APIs for research data services. *Mitteilungen der Vereinigung Österreichischer Bibliothekarinnen & Bibliothekare* 70 (2017), 157–169. Issue 2.

[13] Joseph L. Fleiss. 1971. Measuring Nominal Scale Agreement among Many Raters. *Psychological bulletin* 76 (1971), 378. Issue 5.

[14] Tudor Groza, Knud Möller, Siegfried Handschuh, Diana Trif, and Stefan Decker. 2007. SALT: Weaving the Claim Web. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4825 LNCS, 197–210. https://doi.org/10.1007/978-3-540-76298-0_15

[15] Hassan Hussein, Allard Oelen, Oliver Karras, and Sören Auer. 2022. KGMM – A Maturity Model for Scholarly Knowledge Graphs based on Intertwined Human-Machine Collaboration. https://doi.org/10.48550/ARXIV.2211.12223

[16] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. *K-CAP 2019 - Proceedings of the 10th International Conference on Knowledge Capture*, 243–246. https://doi.org/10.1145/3360901.3364435

[17] Rob Johnson, Michael Watkinson, Anthony Mabe, IJsbrand Jan Aalbersberg, Bev Acreman, et al. 2018. STM: International Association of Scientific, Technical and Medical Publishers Fifth Edition published. (2018). www.ciber-research.eu.

[18] Oliver Karras, Eduard C. Groen, Javed Ali Khan, and Sören Auer. 2021. Researcher or Crowd Member? Why not both! The Open Research Knowledge Graph for Applying and Communicating CrowdRE Research. (8 2021). http://arxiv.org/abs/2108.05085

[19] Michael Kohlhase. 2008. Using LATEX As a Semantic Markup Format. *Mathematics in Computer Science* 2 (12 2008), 279–304. Issue 2. https://doi.org/10.1007/s11786-008-0055-5

[20] Richard J. Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33 (1977), 159–174. Issue 1. http://www.jstor.org/stable/2529310

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

[22] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn E. M. Van der Werf, and Sjaak Brinkkemper. 2016. The Use and Effectiveness of User Stories in Practice, Oscar Daneva Maya and Pastor (Eds.). *Requirements Engineering: Foundation for Software Quality*, 205–222.

[23] Leon Martin and Andreas Henrich. 2022. RDFtex: Knowledge Exchange Between -Based Research Publications and Scientific Knowledge Graphs. In *Linking Theory and Practice of Digital Libraries*, Gianmaria Silvello, Oscar Corcho, Paolo Manghi, Giorgio Maria Di Nunzio, Koraljka Golub, Nicola Ferro, and Antonella Poggi (Eds.). Springer International Publishing, Cham, 26–38.

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

[25] Luc Moreau and Paul Groth. 2015. Provenance of Publications: A PROV Style for LaTeX. *7th USENIX Workshop on the Theory and Practice of Provenance (TaPP 15)*. https://www.usenix.org/conference/tapp15/workshop-program/presentation/moreau

[26] Pavel Nikolaev, Daylond Hooper, Frederick Webber, Rahul Rao, Kevin Decker, Michael Krein, Jason Poleski, Rick Barto, and Benji Maruyama. 2016. Autonomy in Materials Research: A Case Study in Carbon Nanotube Growth. *npj Computational Materials* 2 (10 2016). https://doi.org/10.1038/npjcompumats.2016.31

[27] Allard Oelen, Mohamad Yaser Jaradeh, Kheir Eddine Farfar, Markus Stocker, and Sören Auer. 2019. Comparing Research Contributions in a Scholarly Knowledge Graph. http://purl.org/spar/c4o

[28] Allard Oelen, Markus Stocker, and Sören Auer. 2021. Crowdsourcing Scholarly Discourse Annotations. *International Conference on Intelligent User Interfaces, Proceedings IUI*, 464–474. https://doi.org/10.1145/3397481.3450685

[29] Markus Stocker, Tina Heger, Artur M. Schweidtmann, Hanna Ćwiek Kupczyńska, Lyubomir Penev, et al. 2022. SKG4EOSC - Scholarly Knowledge Graphs for EOSC: Establishing a Backbone of Knowledge Graphs for FAIR Scholarly Information in EOSC. *Research Ideas and Outcomes* 8 (2022). https://doi.org/10.3897/rio.8.e83789

[30] Markus Stocker, Allard Oelen, Mohamad Yaser Jaradeh, Muhammad Haris, Omar Arab Oghli, et al. 2023. FAIR Scientific Information with the Open Research Knowledge Graph. *FAIR Connect* 1, 1 (2023). https://doi.org/10.3233/FC-221513

[31] Jaana Takis, A. Q. M. Saiful Islam, Christoph Lange, and Sören Auer. 2015. Crowdsourced Semantic Annotation of Scientific Publications and Tabular Data in PDF. *ACM International Conference Proceeding Series* 16-17-September-2015, 1–8. https://doi.org/10.1145/2814864.2814887