Crowdsourcing Scholarly Discourse Annotations

Allard Oelen
oelen@l3s.de
L3S Research Center, Leibniz University of Hannover
Hannover, Germany

Markus Stocker
markus.stocker@tib.eu
TIB Leibniz Information Centre for Science and Technology
Hannover, Germany

Sören Auer
soeren.auer@tib.eu
TIB Leibniz Information Centre for Science and Technology
Hannover, Germany

ABSTRACT

The number of scholarly publications grows steadily every year and it becomes harder to find, assess and compare scholarly knowledge effectively. Scholarly knowledge graphs have the potential to address these challenges. However, creating such graphs remains a complex task. We propose a method to crowdsource structured scholarly knowledge from paper authors with a web-based user interface supported by artificial intelligence. The interface enables authors to select key sentences for annotation. It integrates multiple machine learning algorithms to assist authors during the annotation, including class recommendation and key sentence highlighting. We envision that the interface is integrated in paper submission processes for which we define three main task requirements: The task has to be (1) straightforward (2) time efficient (3) well-defined. We evaluated the interface with a user study in which participants were assigned the task to annotate one of their own articles. With the resulting data, we determined whether the participants were successfully able to perform the task. Furthermore, we evaluated the interface's usability and the participant's attitude towards the interface with a survey. The results suggest that sentence annotation is a feasible task for researchers and that they do not object to annotate their articles during the submission process.

CCS CONCEPTS

• **Human-centered computing** → **Web-based interaction**; • **Information systems** → **Web interfaces**; *Crowdsourcing*.

KEYWORDS

Crowdsourcing Text Annotations, Intelligent User Interface, Knowledge Graph Construction, Structured Scholarly Knowledge, Web-based Annotation Interface

ACM Reference Format:

Allard Oelen, Markus Stocker, and Sören Auer. 2021. Crowdsourcing Scholarly Discourse Annotations. In *26th International Conference on Intelligent User Interfaces (IUI '21), April 14–17, 2021, College Station, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3397481.3450685

1 INTRODUCTION

The number of published scholarly articles continues to grow every year [27]. However, scholarly communication remains largely document-based. Scholarly articles are mostly published in PDF format, which is specifically designed for human readability [38] and portability across systems. With this form of publishing, scholarly knowledge is not machine actionable [9, 41]. Knowledge graphs can be employed to represent scientific contributions semantically, render scholarly knowledge more machine actionable, and thus making it easier to find, compare and process knowledge. Knowledge graphs are defined as semantic networks describing entities and their interrelations [42]. Prominent examples of openly available knowledge graphs include DBpedia [4], YAGO [51] and Wikidata [56]. With projects such as Semantic Scholar [3], Microsoft Academic Graph [47] and Open Research Knowledge Graph (ORKG) [26], knowledge graphs are gaining popularity in the scholarly domain to structure scholarly knowledge. Except for ORKG, these graphs only capture metadata about research articles and do not describe the content of reported research work, including research contributions [44].

Populating knowledge graphs with scholarly metadata is a relatively straightforward task due to the low task complexity and high accuracy of automated parsing tools (such as GROBID [33]). In contrast, generating graphs of the contents of research articles (i.e. research contributions) is a considerably more complex task which can currently hardly be performed by Natural Language Processing (NLP) tools alone. Crowdsourcing can be a solution: By including paper authors in the process of creating structured knowledge, it is possible to leverage human intelligence. However, crowdsourcing also comes with its challenges. Firstly, crowdworkers have to decide *what* to model, which requires a thorough understanding of the research topic. Secondly, crowdworkers have to decide *how* to model the knowledge, which is a cognitively demanding task that also relies on skill in conceptual modeling and possibly relevant technologies.

We present a methodology and web-based graphical user interface that serves as a first step towards intertwining human intelligence (via crowdsourcing) and machine intelligence (via machine learning) for the creation of a scholarly knowledge graph. The interface is designed to perform the task of annotating key sentences within scholarly PDF articles. This task focuses specifically on the aforementioned challenge of *what* to model. The user has to select a sentence and afterwards annotate this sentence with an appropriate class. The set of classes consists of a predefined set of 25 discourse elements (e.g., background, contribution and methods). During the annotation process, the user is supported by Artificial Intelligence (AI) tools. With this machine-in-the-loop approach [23] synergy is achieved between crowdsourcing and autonomous NLP extraction. The AI components are available for the tasks that require human judgement and provide support during the decision process. For example, selecting important sentences is supported by automated

IUI '21, April 14–17, 2021, College Station, TX, USA

ACM ISBN 978-1-4503-8017-1/21/04.
https://doi.org/10.1145/3397481.3450685

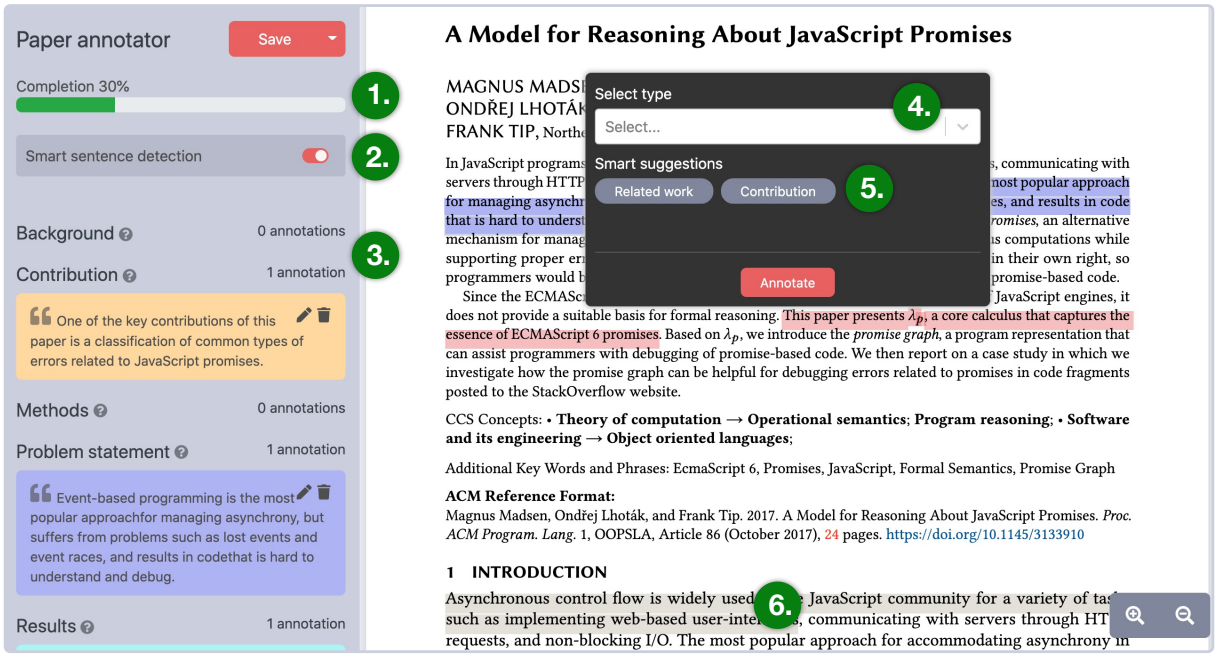

Figure 1: Screenshot of the annotation interface. Numbers indicate system components that are explained in detail in Section 4.2. Legend: 1. Completion indicator 2. Automatic sentence highlighting 3. User annotations 4. Annotation class selector 5. Automatic class suggestions 6. Automatically highlighted sentence.

sentence highlighting. Additionally, selecting suitable classes for sentences is supported by a class recommendation tool. We envision that this interface is integrated in paper submission systems to produce a more structured description of the paper's content. Having annotated sentences is a crucial step towards generating truly structured semantic scholarly knowledge. Among other things, it is possible to further process the annotated sentence to create better structured and more semantic data.

We address the following research questions: 1) How to design an intelligent user interface to populate a scholarly knowledge graph using crowdsourcing? 2) How to employ a machine-in-the-loop approach to assist users in this process? These questions are addressed by devising use cases which are used to determine the requirements. Based on the requirements, system components are designed that address those requirements. Finally, to evaluate whether the requirements are met, a user evaluation is conducted.

2 BACKGROUND AND RELATED WORK

Text annotation tools are widely used in the Natural Language Processing (NLP) community to visualize automatically generated annotations by NLP tools, such as [50]. Additionally, some of these tools focus on corpus annotation and support the generation of complex corpora [7]. Such annotation tools have proven to be valuable for the collaborative creation of datasets. Due to the popularity of the PDF format, PDF annotation has received considerable attention (e.g., [18, 46, 52]). PDF documents are widely used among various domains, for example, in government data [13], legal documents [30], patents [5] and product datasheets [54]. However, the PDF format hinders access and reuse of the data presented within the documents [14]. Eriksson [18] presents a tool to directly generate semantic descriptions from PDF documents. This tool requires annotators to have data modeling knowledge since the annotator is responsible for the modeling aspect. Shindo et al. [46] integrates multiple linguistic technologies in the annotation tool. More similar to our approach, Takis et al. [52] presents a crowdsourcing approach for creating semantic annotations in scientific publications. In contrast to our approach, Takis et al. focus on entity annotation rather than full sentence annotation. Furthermore, the integration of ontologies in their approach is prominently present. In our work, we focus on task separation and design an interface that can be used without requiring any modeling knowledge. In addition, we aim for task simplicity to make the interface suitable for paper submission integration. Capadisli et al. created a tool Dokieli that enables authors to create semantic annotations within the authoring tool itself [10]. This differs from our approach, where we present a tool to create annotations retrospectively (i.e., after writing an article). Related to our approach, Snow et al. [48] has demonstrated that crowdsourcing can be successfully employed to generate labeled datasets. Such crowdsourcing approaches rely on comprehensive task descriptions and guidelines to ensure high-quality results [11]. With respect to our interface, this means that we have to make a clear task description and leave no room for ambiguity.

3 USE CASES

We now discuss multiple use cases supported by illustrative examples from the literature. We begin with two use cases in which the annotation interface is used to generate structured data (*data*

entry). The four use cases that follow outline the usefulness of the generated annotations (*data consumption*).

3.1 Data Entry

Paper submission. The annotation interface is mainly designed to be used as part of paper submission processes. More specifically, when the camera-ready article is uploaded. This prevents additional workload when uploading the paper for review. The interface can be integrated in open-access platforms such as arXiv [21] or CEUR Workshop Proceedings (CEUR-WS)[1]. A similar approach has been taken by arXiv, which integrated the ScienceWISE [1] platform where authors can add automatically generated entity annotations to their uploaded articles. Additionally, CEUR-WS has been frequently used as data source for semantic publishing approaches (e.g., [29, 32, 44]).

Literature review. Sentence annotations can also be generated while reading articles. In this case, not only authors but also other researchers can create annotations. Compared to the *paper submission* approach, this will most likely produce less complete and possibly lower quality results. Less complete results are expected because readers will presumably only annotate what is of interest to them at the time of reading the article. Lower quality results are likely because readers are less familiar with the article's content than the authors. However, due to the scalability of this approach the generated annotations combined are still valuable. Although our interface is not designed to support this use case directly, it could be adopted easily.

3.2 Data Consumption

Further semantification. The result of the annotation task is a set of sentences annotated with a relevant discourse class. These sentences must be transformed into more machine-readable descriptions. This can be done automatically using Named Entity Recognition and Classification (NERC) [39]. The resulting recognized entities can leverage the already determined discourse class. For example, if a method is recognized in a discourse element with the *Background* class, this means that the method is discussed within the paper. However, it does not necessarily mean that this method has been employed, since it is discussed as background information. Furthermore, the sentence can be modeled using existing ontologies. However, this task relies on domain experts with knowledge of data modeling.

Structured abstract. Based on the annotated sentences a structured abstract can be generated automatically. Structured abstracts have a long history [19] and are commonly used in certain domains, most prominently in life sciences. Research shows that structured abstracts make it easier for researchers to select appropriate articles more quickly [40]. Within our user interface, the annotator is urged to only annotate the most important sentences. This results in an abstract that provides a relevant summary of the article.

Effective search. With annotated sentences, search can be improved in two ways, by more effectively finding papers and by enhanced navigation within the paper. It is possible to more effectively look for concepts that are related to specific discourse elements. This can be further enhanced with additional semantification. Based on an experiment, de Ribaupierre and Falquet [16] reported that the participants found more useful results using faceted search compared to keyword based search. The facets were generated by extracting discourse elements and using annotations. Additionally, annotations can help in navigating the paper, displaying the highlighted sentences and their classes to readers. Highlighting sentences within a text has proven to increase information comprehension and retention [20, 34].

NLP training data. Finally, annotations can serve as gold standard for NLP related tasks. A frequently recurring task in dataset generation is human annotation of the data. After the data is annotated (or labeled) it can be used to train and test machine learning algorithms. Labeling of datasets is oftentimes done manually by expert users (cf. [53]). This is an expensive and time-consuming task and therefore other methods have been proposed, for example, leveraging crowdsourcing for dataset labeling [11]. The resulting data from our annotation system can be used to train NLP systems in multiple ways. Among other tasks, this includes the task of recognizing various discourse elements within a scholarly article.

4 SYSTEM DESIGN

In this section we discuss requirements, present the system architecture and its components and outline the technical implementation. The annotation interface is integrated in the Open Research Knowledge Graph (ORKG) and is available online.[2]

4.1 System Requirements

Based on the use cases described in Section 3, we determined the system requirements from which the most essential ones are listed below. Additionally, we used the findings of a previously conducted user study where we asked seminar students (n = 14) to generate structured descriptions from papers they read. For this task, they had to use a tool that was designed to populate a scholarly knowledge graph by creating entities and the relations between them. In contrast to the annotation tool presented in this work, the tool did not rely on text annotation but on manual structured data creation. It was designed in such a way that it did not require any technical skills to perform the task.

The interface must adhere to the following functional requirements:

- FR1 **Sentence annotation.** The interface should provide a method that enables users to select sentences within scholarly articles in PDF format. The selected sentences are annotated with an appropriate discourse class.
- FR2 **Task separation.** The task should focus on *what* to model and not *how* to model it. According to the seminar user study, 71% of the students indicated that the data modeling aspect is the most time-consuming aspect. By separating the task of data selection and data modeling, we provide a task that is more feasible for crowdsourcing during a paper submission process.

[1] http://ceur-ws.org

[2] https://www.orkg.org/orkg/pdf-text-annotation

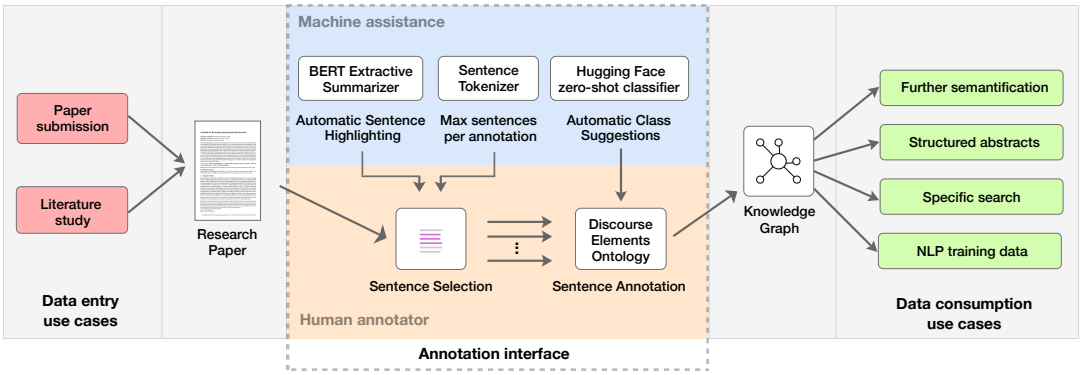

Figure 2: Overview on the system design illustrating the intertwining of human and machine intelligence for the scholarly article annotation.

FR3 **Machine assistance.** Users should be supported by machine assistance during the annotation process. The machine assistance should provide guidance during the user's decision process. This includes guidance for which sentences to annotate and help deciding which class to annotate the sentence with.

Furthermore, we defined the following non-functional requirements:

NFR1 **Straightforward.** The task should be easy to perform. This means that the task is not cognitively demanding, has a low complexity and takes little time to complete, which are typical characteristics used in crowdsourcing tasks [28]. The task easiness should not be confused with the usability of the system. In the seminar user study, the tool was evaluated with a System Usability Scale (SUS) [8] score of 67, which is average. Still, the task of modeling scholarly data in a structured way was a complicated endeavor.

NFR2 **Time efficient.** To convince paper authors to annotate their papers during the submission process, the task should not be time consuming. We consider less than 10 minutes as time efficient.

NFR3 **Well defined.** The task definition has to be unambiguous. This contributes positively to the quality and consistency of the generated data [2]. If the resulting annotations are according the task description, it means the task is well understood and we consider the interface well defined.

4.2 Architecture and Components

The overall system architecture is shown in Figure 2. A key concept is to intertwine human and machine intelligence. A core component is the human user-driving sentence selection, which is facilitated by the two machine intelligence components, Extractive Summarizer and Sentence Tokenizer. Similarly, the second step Sentence Annotation using the Discourse Elements Ontology is facilitated by automatic class suggestions of the zero-shot classifier. We now discuss the individual system components. For each component we explain how its design ensures that the system requirements are met.

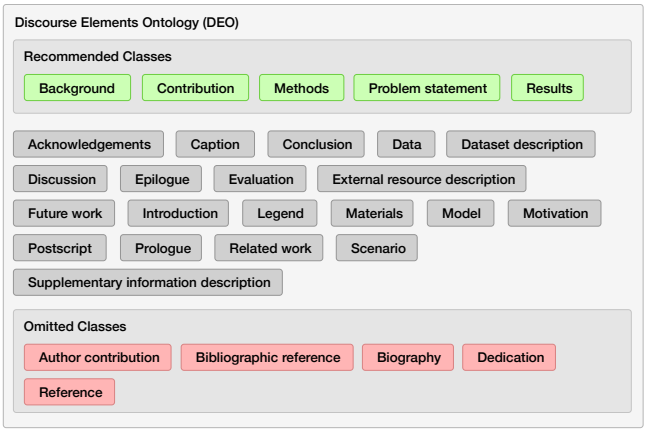

Figure 3: Discourse annotation classes. Green boxes indicate the recommended classes, red the omitted classes and grey the remaining classes. In total, our model uses 25 classes.

4.2.1 Discourse Knowledge Representation. Users can choose between a predefined set of discourse classes to annotate a selected sentence (Figure 1, node 4). To support interoperability with other systems, we build on the existing Discourse Elements Ontology (DEO) [12] to model the data. This ontology is part of the Semantic Publishing and Referencing (SPAR) ontologies which are designed to describe the scholarly publishing domain [43]. Our discourse knowledge representation model is illustrated in Figure 4. We omitted five classes as they are irrelevant for this annotation task (either because it is straightforward to extract this data automatically or because the data is not useful for the data consumption use cases). The omitted classes are: (1) Author contribution, (2) Bibliographic reference, (3) Biography, (4) Dedication and (5) Reference. The resulting set of discourse elements consists of 25 classes. These classes are listed in Figure 3. This component is part of FR1. Additionally, by limiting the number of classes it also contributes to NFR1 and NFR2.

4.2.2 Automatic Sentence Highlighting. To guide users during sentence selection, automatic sentence highlighting is applied. This

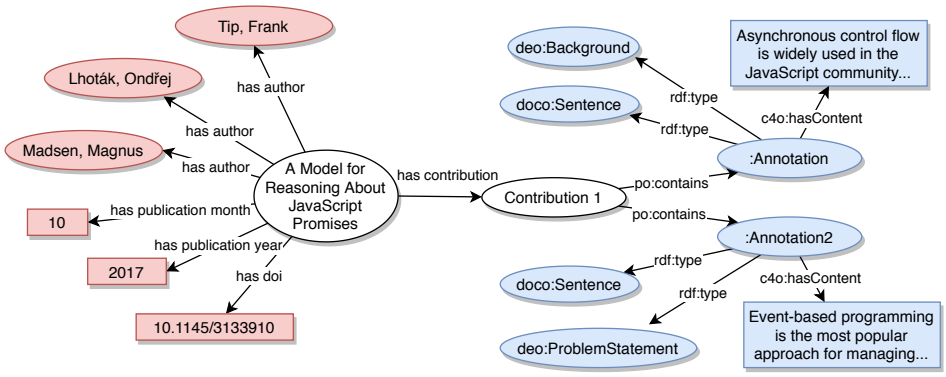

Figure 4: Example of the resulting knowledge graph obtained from the annotation task. The red nodes on the left depict the automatically fetched metadata (metadata types are omitted for simplicity). The white nodes are system concepts related to our internal data model. The blue nodes on the right depict two annotated sentences.[3]

is displayed in Figure 1, where node 2 and 6 respectively refer to activation and visualisation of highlights. The highlights aim to ease the annotation task and are implemented for FR3 and NFR1. The highlights are generated by applying automatic sentence summarization to the full text of the article. The resulting summary sentences are highlighted within the text. For sentence summarization, we adopted BERT embeddings [17] for extractive text summarization inspired by the approach in [37]. Compared to abstractive summarization, where vocabulary is used beyond the specified text, extractive summarization uses the exact structures and sentences from the original text [36]. Extractive summarization is thus more suitable as it allows for tracing back and highlighting the original sentence.

Since summarization tools specifically focus on extracting key sentences from a text, we leverage this technique to highlight key sentences within the original text. Automatic text summarization techniques are not always accurate and therefore not commonly used. This is not an issue in our use case, since the highlights appear in context and can be ignored when not relevant, which contrasts to a self-contained summary where the quality of the summary plays a crucial role [49]. Furthermore, the user has the possibility to hide all automatically generated highlights (Figure 1, node 2).

4.2.3 Automatic Class Suggestions. The class suggestions help users to choose from the 25 discourse classes (Figure 1, node 5), thus addressing FR3 and NFR2. The class recommendations can save time during the annotation and are generated using a zero-shot classifier from Hugging Face [15]. A zero-shot text classifier is able to predict classes for text without requiring training data [58]. This makes such a classifier suitable for our task, since the selected sentences can be classified according to the DEO ontology. The accuracy of the recommendations depends on the text structure. When certain key phrases are present in the text (e.g., "In the future..." for future work or "In the presented use case..." for a scenario) the classifier is able to make accurate suggestions. However, the accuracy drops when such key phrases are not present. A maximum of five suggestions ranked above an empirically determined threshold are displayed to the user.

4.2.4 Completion and Recommended Classes. The task completion bar indicates how complete the annotations are (Figure 1, node 1). It helps defining the task by providing guidance on the progress, which relates to NFR3. The completion rate only provides an indication, users do not have to reach 100% in order to finish the task. Completion is based on recommended classes, namely: (1) Background (2) Contribution (3) Methods (4) Problem statement (5) Results. The classes are selected based on the literature and the importance of these classes is argued as follows. Firstly, the classes are closely related to the elements from the IMRAD (Introduction, Methods, Results, Discussion and Conclusion) structured abstract style which are considered important features of articles [35]. Furthermore, findings from de Ribaupierre and Falquet show that researchers are mainly looking for findings, hypothesis, methods and definitions when reading scholarly literature [16]. These concepts are largely covered by the five recommended classes we selected. The completion rate indicator determines whether at least two annotations per recommended annotation class are created, which results in a completion rate of 100%.

4.2.5 Miscellaneous Guidance Functions. The following components further guide users during the annotation task. (1) **Annotation limit.** A maximum of three annotations per class can be created, thus maintaining the scope of the annotations (NFR3). It forces users to distribute the annotations across multiple classes which consequently contributes to higher data quality. The annotation limit (indicated by a warning) is not strictly enforced; hence, it is possible to deliberately cross the limit. (2) **Maximum sentences per annotation.** An annotation can only contain a maximum of two sentences. The selected text is tokenized by sentences. This also contributes to NFR3. It prevents users from annotating full paragraphs and forces them to select only key sentences within the article. As with the annotation limit, a warning is displayed as the limit is a suggestion and not enforced. (3) **Tooltips and guidance.** Tooltips are displayed throughout the interface. This contributes

[3]Used ontologies: DEO (Discourse Elements Ontology) - http://purl.org/spar/deo, C4O (Citation Counting and Context Characterization Ontology) - http://purl.org/spar/c4o, PO (Pattern Ontology) - http://purl.org/spar/po, DoCO (Document Components Ontology) - http://purl.org/spar/doco

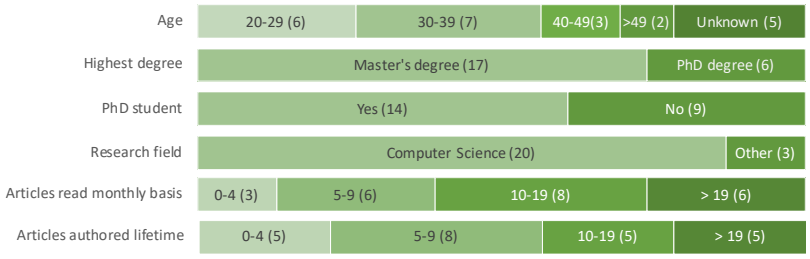

Figure 5: Participants' demographics (n = 23).

to NFR1. The tooltips explain system functionalities and the DEO classes. For each class, a description explains the purpose of the class. Furthermore, a guided help tour automatically appears when using the interface. The tour explains the goal of the annotation task and provides an overview of the main functionalities.

4.3 Technical Implementation

The interface has been implemented in JavaScript using React[4], the source code and its documentation are available online[5]. For displaying PDF files, we used an extended version of PDF.js[6], which is a JavaScript library for parsing and rendering PDF files developed by Mozilla. Since PDF.js is used as default PDF viewer within the Firefox web browser it is able to correctly display PDF files within a browser environment. Additionally, PDF.js has been used successfully in other PDF annotation tools (e.g., [46, 52]). The default PDF search functionality is leveraged and extended to support multiple search queries at once. The endpoints for the machine learning components are implemented in Python. The data is stored in a Neo4j[7] graph database which is using the Neosemantics[8] plugin for improved ontology support.

For saving the annotations, users are requested to provide a paper title or a Digital Object Identifier (DOI) to save the data. In case a DOI is provided, additional metadata related to the article is automatically fetched via Crossref [31]. Among others, this includes the article's title, authors and publication date. Users do not have to provide this data manually, which makes the annotation task more time efficient (NFR2). Figure 4 visualizes an example of the data structure for a saved paper. Various external ontologies are used to improve data interoperability.

5 EVALUATION

The interface is evaluated to determine whether the paper annotation task is indeed a feasible task to be performed by academics. Additionally, we want to obtain insights in the attitudes towards machine-assisted paper annotation in general. We focus specifically on evaluating the individual components discussed in Section 4.2. The evaluation also provides insights into whether or not the functional requirements are met and thus if the functionalities were indeed designed as envisioned, as well as non-functional requirements and thus if the quality aspects are met. We evaluated the interface by means of a user study. Firstly, we evaluated the participants' opinions about the usability and their attitudes towards our approach in general. Secondly, we analysed the data produced by the participants during the annotation task.

5.1 Evaluation Setup and Data Collection

An online task description was circulated among academic communities. This task description provided a brief explanation of how to participate in the evaluation. Participants were asked to annotate a paper with the paper annotation interface described here. This could either be an article they authored themselves or an article they (recently) read. Afterwards, participants were asked to complete an online questionnaire. The task description did not provide any instructions regarding the functionalities of the interface nor did we instruct the participants regarding the annotation task. This ensures that the interface can be used without external assistance and matches the real-world setting in which authors are asked to annotate their articles during submission without further help. We communicated that the evaluation takes approximately 20 to 30 minutes in total. A total of 23 researchers participated in the user study. Figure 5 displays the demographics of the participants, including data for the number of articles each participant reads and publishes, as a proxy for the level of expertise. Participants with more experience on reading and writing articles are presumably able to annotate more quickly and with a higher quality. As the demographics data shows, participants with varying levels of expertise participated in the study.

To determine the usability of the interface, we incorporated the System Usability Scale (SUS) [8] in the questionnaire. Furthermore, to determine the workload of the task we included questions from the NASA Task Load Index (TLX) [25]. This provides insights into the perceived workload by participants for the annotation task. To reduce the length of the questionnaire, we conducted the Raw TLX, which eliminates weighting the questions. Finally, we included additional questions to determine the participants' attitude towards the interface and the overall task. This included a question asking for general feedback about the interface.

5.2 Evaluation Results

The System Usability Scale evaluation resulted in a score of 76.09 (out of 100) which is considered "good". The individual questions and answers are displayed in Figure 6. Because of the format, text

[4]https://reactjs.org
[5]https://gitlab.com/TIBHannover/orkg/orkg-frontend/-/tree/e0a6a7a8d022119d9fb5cc7b749052f0f1c194d0/src/components/PdfTextAnnotation
[6]https://mozilla.github.io/pdf.js
[7]https://neo4j.com
[8]https://neo4j.com/labs/neosemantics

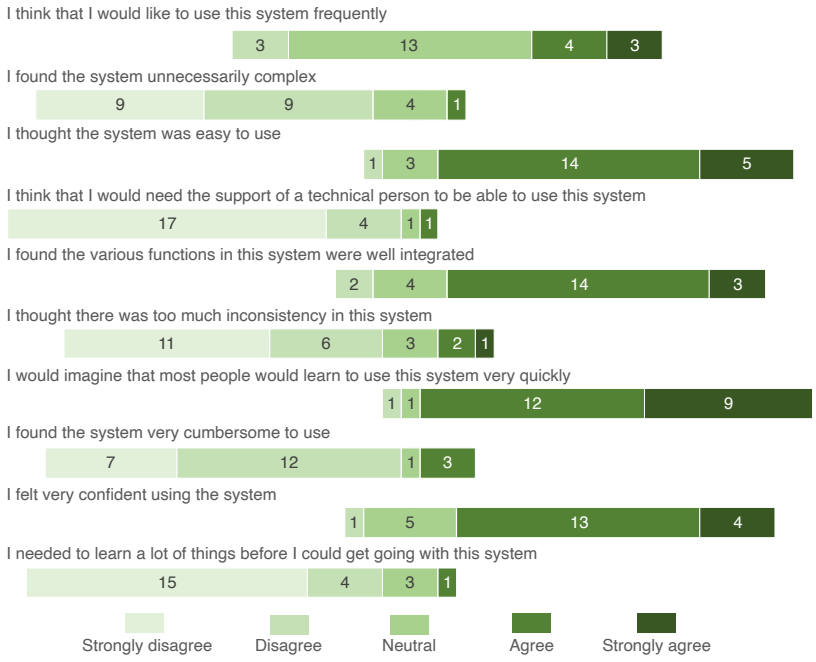

Figure 6: Individual System Usability Scale questions and answers, resulting in a mean score of 76.09 (SD = 14.38).

selection and extraction in PDF files remains a challenging task (see also [52]). Various participants explained that the text selection should be improved. The question "I think that I would like to use this system frequently" is rated lowest. An explanation could be that the participants are not (yet) performing article annotation on a regular basis in daily work. Therefore, the question is answered based on their own situation rather than on the general usability of the system. A similar conclusion was suggested by Weber et al. [57].

The results of the TLX evaluation are shown in Figure 7. The mean TLX score of 35.87 is considered low compared to the mean of 45.29 found in the meta-analysis from Grier [24]. This indicates a low perceived workload by the participants. On average, the three highest scored questions are related to mental demand (52%), performance (45%) and effort required (46%). This means that the annotation task in general does require some mental effort. However, this is expected due to the various task constraints (e.g., annotate only the most important sentences or a maximum of three annotations per class) and will possibly be partially mitigated by increasing familiarization of users during regular use of the system. The frustration level was relatively low (28%), this is in line with the positive SUS score.

The participants' attitudes towards the interface are visualized in Figure 8. Participants are split on the question whether the task is time consuming, most participants rate this as neutral. Most participants spent between five and 10 minutes to annotate their paper (52%). Of the remaining participants, 18% spent less than five minutes and 30% more than 10 minutes. No clear time difference could be observed between more experienced participants (i.e., participants with a PhD degree) and other participants. The majority of participants would be willing to annotate their paper in the

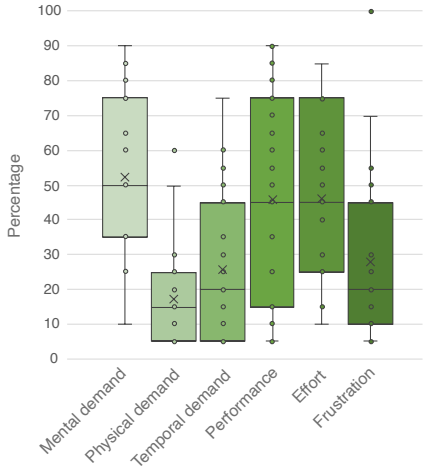

Figure 7: Raw NASA Task Load Index results (lower is better). The mean TLX score is 35.87 (SD = 26.17). The middle line represents the median and a cross the mean. Vertical lines represent the minimum and maximum values and circles the individual points and outliers.

submission process, given that the paper has been accepted already. The remaining questions in Figure 8 relate to the machine-assisted aspects of the system. The vast majority of the participants has a positive attitude towards leveraging machine-assisted technologies during the annotation task as they would like to see more artificial intelligence technologies being integrated. Participants are also split on the quality of the automatic class suggestions (called smart

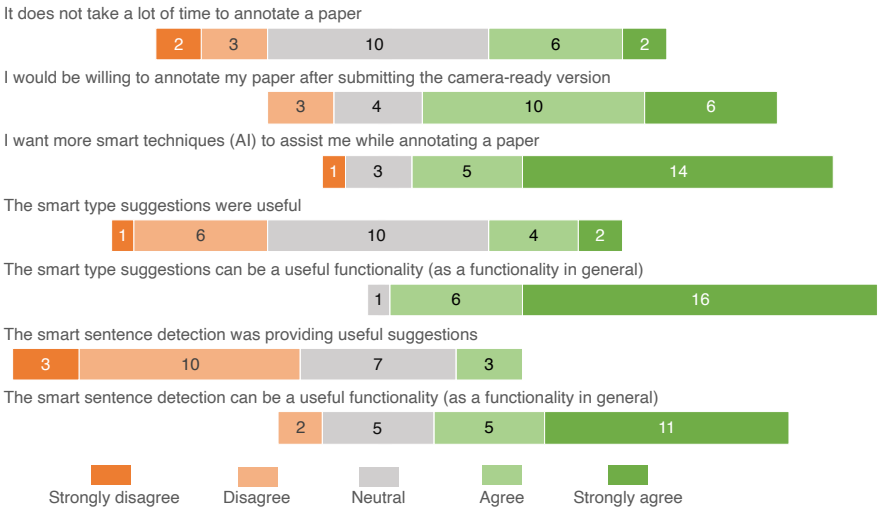

Figure 8: Participants attitudes towards the annotation task, specifically focused on the machine assistance perspective. Higher values in green represent more positive attitudes.

type suggestions in the interface). However, the majority agrees that the functionality is useful, given that the suggested classes are more relevant. The results of the automatic sentence highlighting (called smart sentence detection in the interface) are not always helpful according to the participants. But also here, most of the participants agree that the functionality is useful in general. These relatively positive results indicate that the participants appreciate the integration of AI in a user interface, even though the individual performance of the machine learning components leaves room for improvement according to some participants. This is expected given that we did not focus on a particular scholarly domain. Overall, this confirms that our approach for sentence annotation interface is a promising direction.

Furthermore, we determined whether the preselected recommended classes are indeed of interest to researchers. In the questionnaire, participants were asked to select five discourse classes they deem most important when reading scholarly literature. The 16 most selected classes are listed in Figure 9. As this figure indicates, four of the recommended classes are indeed considered most important. Ranked 10th, the background class is the only exception. Since the background class was included in the recommended classes, it was more prominently positioned in the interface. Therefore, it has a relatively high annotation frequency compared to the perceived class importance. Although not considered important by the participants, background information is valuable especially when creating structured abstracts. Therefore, we suggest to keep the background class in set of recommended classes. Furthermore, this figure displays the number of annotations per the listed discourse classes. As expected, the recommended classes are used most frequently, as they are prominently present in the interface. Interestingly, the related work class is used relatively frequently as well. This could be explained by the assumption that is it straightforward to recognize related work within an article.

Table 1 reports statistics for the generated annotations during the evaluation. An average completion rate of 73% has been reached.

Table 1: Statistics of the generated annotations per article from the user evaluation.

	Mean	SD	Max	Min
General				
Annotations per article	13.18	6.52	24	3
Completion ratio	72.72	24.72	100	10
Extra recommended classes	1.90	1.94	7	0
Non-recommended classes	3.91	4.42	14	0
Machine-assisted components				
Selected class in suggestions ratio	56.55	29.40	94.44	33.33
Selected class as first suggestion ratio	17.24	14.65	50	0
Annotations over two sentence limit	0.95	1.56	4	0
Annotations over three class limit	0.82	1.97	9	0

The completion rate only provided guidance and it was not mandatory for the participants to reach 100%. The relatively high completion rate indicates that participants were indeed guided by the completion bar, but did not feel obligated to reach the full completion. Per article, on average 3.9 non-recommended annotation classes were used. Indicating that the interface was successful in guiding users towards the recommended classes but also allowing other classes. With respect to the machine-assisted components, 57% of the suggested classes were indeed selected by the user. In 17% of the cases, this was the first suggestion in the list (i.e., the class with the highest certainty, as determined by the classifier). This leaves room for improvement which is in line with the results from the questionnaire (Figure 8). Finally, on average one annotation per article contained more than two sentences. In this case, a warning was shown to the user, which did however not hinder saving the annotation. The same applies to the maximum number of annotations, which was set to three. On average, an annotated article had one annotation class with more than three annotations. These relatively low numbers of crossing the limits indicate that warning participants about violations, but not enforcing them, is indeed effective.

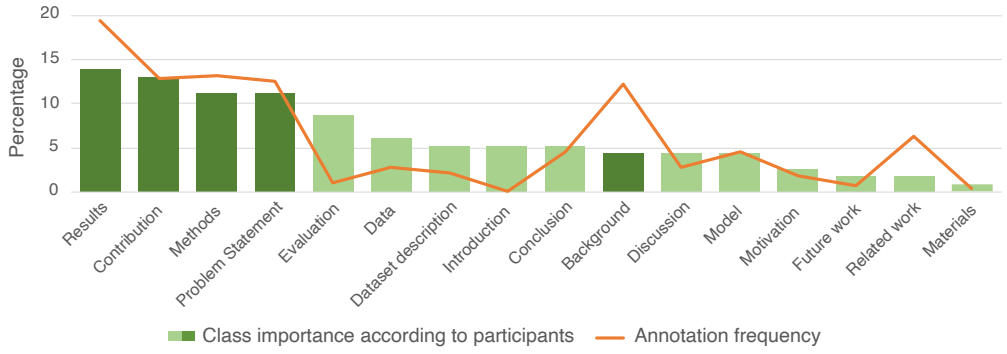

Figure 9: Top 16 discourse classes ranked by importance according to the participants (in dark green the recommended classes). The orange line shows the annotation frequency per class.

6 DISCUSSION

In order to answer our first research question, "How to design an intelligent user interface to populate a scholarly knowledge graph using crowdsourcing", we determined the system requirements based on several use cases. Our user evaluation focused on determining whether the requirements are met. Based on the functional requirements, we implemented a PDF sentence annotation component. The annotation task was focused towards what to model and not on how to model data. For example, users do not have to decide what ontologies to use or how to structure the data. Related to the second research question, "How to employ a machine-in-the-loop approach to assist users in this process", we integrated multiple machine-assisted technologies. This includes machine learning based components, such as the automatic sentence highlighting and the automatic class suggestions. With respect to the non-functional requirements, we conclude that the task was indeed straightforward as suggested by the NASA Task Load Index (TLX) results. Furthermore, the evaluation shows that most participants spent less than 10 minutes for the task. We consider that as time efficient, although the results were divided for the question "It takes a lot of time to annotate a paper". Despite that, most authors are willing to annotate their paper during the camera-ready submission. This suggests that a crowdsourcing approach, in which authors are included to generate structured paper data, is viable in practice. Finally, participants were able to perform the task without requiring additional help. They reported high levels of confidence and low frustration levels while using the system. This indicates that the task was well defined.

Once the scholarly knowledge graph contains a sizeable number of articles it can potentially revolutionize scholarly communication. For example, by providing more effective search or as a tool to analyze scholarly knowledge more efficiently. Our annotation interface serves as a step towards more structured scholarly communication. Generally, the more structured the data in the graph is, the better machines can read and process this data. Specifically, the annotated sentences can be complemented with structured data to further improve the data's machine readability. This can be done in an automated fashion by leveraging techniques such a Named Entity Recognition (NER) [39] to automatically detect concepts in a sentence. This results in additional structured data which in turn can be further enhanced by linking these concepts to other knowledge graphs, by means of Entity Linking [45]. These technologies can be effectively employed by leveraging the annotated sentences, thus applying them targeted on a specific sentence rather than on the full text of an article. Future work will focus on applying these technologies on the annotated sentences.

The evaluation results indicate that the usability of the system is "good" and that the workload is acceptable. With respect to the machine assistance, specifically the automatic sentence highlighting and automatic class suggestions, participants suggested that the quality of the recommendation could be improved. Improving these specific machine learning algorithms is out-of-scope here. More interestingly, participants indicated that they appreciate the overall integration of Artificially Intelligence (AI) within the user interface. Despite the quality of recommendations not being optimal, they would prefer more AI-powered support during the annotation. We conclude that the quality of the assistance does not have considerable negative impact on the user experience nor does it significantly influence the participants' attitude towards such technologies. This contributes to the concept of machine assistance, whereby a machine could help a user but is not critical to complete the task. Participants were able to ignore class suggestions and to disable sentence highlights if they considered them to be irrelevant. Therefore, we argue that the possibility to dismiss machine assistance is crucial for a system's usability.

The presented interface and the findings from this work are not exclusively applicable to the scholarly domain but can be transferred to other domains as well. As mentioned in Section 2, the PDF format is widely used in various fields and applications (legal documents, patents etc.) where they dominate as a digital means to share knowledge. With minor adjustments, the presented interface can be adopted to annotate such documents and ultimately generate structured data from them. In principle, merely the ontology for annotation classes has to be changed to support other use cases. Furthermore, our findings related to users' attitudes towards machine-assisted user interfaces are relevant to interface design in general.

6.1 Limitations and Future Work

Arguably, our evaluation could be larger and include more participants, which would improve the validity of the results. We target participants with an academic research background, which are notoriously hard to recruit. Moreover, the task is not suitable for online crowdsourcing platforms such as Amazon Mechanical Turk. Additionally, a more thorough evaluation of the effectiveness of the intelligent system components is required. We acknowledge that the evaluation is limited in scope and are considering to conduct a broader evaluation with a more diverse audience for future research. Despite these limitations, the evaluation still provides helpful insights and clear indications on the participants' attitudes towards the overall approach. Moreover, Tullis and Stetson [55] have shown that the System Usability Scale (SUS) provides reliable results even with relatively small sample sizes (e.g., n = 12).

Most of the participants have a Computer Science related background (Figure 5). This could introduce a bias affecting the usability score and overall attitude towards intelligent technologies. Indeed, computer scientists are generally more experienced in adopting novel computer user interfaces. However, the interface was designed to also allow non-technical users to annotate papers. For example, technical jargon is avoided to make the interface understandable for users with different backgrounds. Furthermore, text annotation has been successfully employed in other domains (e.g., [6, 22]), indicating that the task itself is generalizable across domains. The effectiveness of these measures and the generalizability of the method outside Computer Science will be further investigated in future work. Furthermore, the difference between annotations made by authors and by readers is a compelling future research direction.

7 CONCLUSION

We presented a web-based user interface to crowdsource scholarly discourse annotations. The interface integrates several machine-assisted components to guide users during the annotation process. This work is part of a larger research agenda and a corresponding open science infrastructure development. We deem that the integration of human and machine intelligence for creating a comprehensive knowledge graph representing research findings is a key prerequisite for solving scholarly communication deficiencies such as the proliferation of publications, the reproducibility crisis or the deterioration of peer-review. In particular, we envision that the interface is integrated in paper submission processes where paper authors are requested to annotate their own papers. A scholarly knowledge graph is created using the annotated sentences combined with the paper's metadata. Our user study results indicate that the annotation interface has a good usability and that the annotation task does not require significant cognitive workload. This suggests that sentence annotation is a feasible task to be performed by researchers. In future work, we will focus on implementing the use cases. Furthermore, future work includes the semi-automated extraction of entities from annotated sentences.

ACKNOWLEDGMENTS

This work was co-funded by the European Research Council for the project ScienceGRAPH (Grant agreement ID: 819536) and the TIB Leibniz Information Centre for Science and Technology. The publication of this article was funded by the Open Access Fund of Technische Informationsbibliothek (TIB). We want to thank our colleague Mohamad Yaser Jaradeh for his contributions to this work.

REFERENCES

[1] Karl Aberer and Alexey Boyarsky. 2011. ScienceWISE: a Web-based Interactive Semantic Platform for scientific collaboration. *10th International Semantic Web Conference (ISWC 2011-Demo)* (2011). https://doi.org/10.1007/978-3-662-46641-4_33

[2] Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. 2013. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing* 17, 2 (2013), 76–81. https://doi.org/10.1109/MIC.2013.20

[3] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine Van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 3 (2018), 84–91. https://doi.org/10.18653/v1/n18-3011

[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehman, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. *The semantic web* (2007), 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

[5] Antonin Bergeaud, Yoann Potiron, and Juste Raimbault. 2017. Classifying patents based on their semantic content. *PLoS ONE* 12, 4 (2017), 1–22. https://doi.org/10.1371/journal.pone.0176310

[6] Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. GATE Teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation* 47, 4 (2013), 1007–1029. https://doi.org/10.1007/s10579-013-9215-6

[7] Kalina Bontcheva, Hamish Cunningham, Ian Roberts, and Valentin Tablan. 2010. Web-based collaborative corpus annotation: Requirements and a framework implementation. *New Challenges for NLP Frameworks* (2010), 20–27.

[8] John Brooke. 1996. SUS: a "quick and dirty'usability. *Usability evaluation in industry* (1996), 189.

[9] Cristina-Iulia Bucur, Tobias Kuhn, and Davide Ceolin. 2020. A Unified Nanopublication Model for Effective and User-Friendly Access to the Elements of Scientific Publishing. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 104–119. https://doi.org/10.1007/978-3-030-61244-3_7

[10] Sarven Capadisli, Amy Guy, Ruben Verborgh, Christoph Lange, Sören Auer, and Tim Berners-Lee. 2017. Decentralised authoring, annotations and notifications for a read-write web with dokieli. In *International Conference on Web Engineering*. Springer, 469–481. https://doi.org/10.1007/978-3-319-60131-1_33

[11] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. *Conference on Human Factors in Computing Systems - Proceedings* 2017-May (2017), 2334–2346. https://doi.org/10.1145/3025453.3026044

[12] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. 2016. The Document Components Ontology (DoCO). *Semantic Web* 7, 2 (2016), 167–181. https://doi.org/10.3233/SW-150177

[13] Andreiwid Sheffer Corrêa and Pär-Ola Zander. 2017. Unleashing Tabular Content to Open Data. *Proceedings of the 18th Annual International Conference on Digital Government Research* (2017), 54–63. https://doi.org/10.1145/3085228.3085278

[14] Bart Custers and Daniel Bachlechner. 2018. Advancing the EU Data Economy: Conditions for Realizing the Full of Potential of Data Reuse. *SSRN Electronic Journal* (2018), 1–19. https://doi.org/10.2139/ssrn.3091038

[15] Joe Davison. 2020. Zero-Shot Learning in Modern NLP. *(accessed on 2020-09-30)* (2020). https://joeddav.github.io/blog/2020/05/29/ZSL.html

[16] Hélène de Ribaupierre and Gilles Falquet. 2018. Extracting discourse elements and annotating scientific documents using the SciAnnotDoc model: a use case in gender documents. *International Journal on Digital Libraries* 19, 2-3 (2018), 271–286. https://doi.org/10.1007/s00799-017-0227-5

[17] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1, Mlm (2019), 4171–4186.

[18] Henrik Eriksson. 2007. An Annotation Tool for Semantic Documents (System Description). *4th European Semantic Web Conference (ESWC)* (2007), 759–768. https://doi.org/10.1007/978-3-540-72667-8_54

[19] Ad Hoc Working Group for Critical Appraisal of the Medical Literature. 1987. A proposal for more informative abstracts of clinical articles. *Annals of Internal Medicine* 106, 4 (1987), 598–604.

[20] Robert L Fowler and Anne S Barker. 1974. Effectiveness of highlighting for retention of text material. *Journal of Applied Psychology* 59, 3 (1974), 358.

[21] Paul Ginsparg. 2011. ArXiv at 20. *Nature* 476, 7359 (2011), 145–147. https://doi.org/10.1038/476145a

[22] Glenn T Gobbel, Jennifer Garvin, Ruth Reeves, Robert M Cronin, Julia Heavirland, Jenifer Williams, Allison Weaver, Shrimalini Jayaramaraja, Dario Giuse, Theodore Speroff, et al. 2014. Assisted annotation of medical free text using RapTAT. *Journal of the American Medical Informatics Association* 21, 5 (2014), 833–841. https://doi.org/10.1136/amiajnl-2013-002255

[23] Ben Green and Yiling Chen. 2019. The principles and limits of algorithm-in-the-loop decision making. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019). https://doi.org/10.1145/3359152

[24] Rebecca Grier. 2015. How high is high? A metanalysis of NASA TLX global workload scores. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 59. https://doi.org/10.1177/1541931215591373

[25] Sandra G. Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomics Society* (2006), 904–908. https://doi.org/10.1177/154193120605000909

[26] 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* (2019), 243–246. https://doi.org/10.1145/3360901.3364435

[27] Arif Jinha. 2010. Article 50 million: An estimate of the number of scholarly articles in existence. *Learned Publishing* 23, 3 (2010), 258–263. https://doi.org/10.1087/20100308

[28] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E Kraut. 2011. Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 43–52. https://doi.org/10.1145/2047196.2047202

[29] Maxim Kolchin, Eugene Cherny, Fedor Kozlov, Alexander Shipilo, and Liubov Kovriguina. 2015. CEUR-WS-LOD: Conversion of CEUR-WS Workshops to Linked Data. *Semantic Web Evaluation Challenges* 1, September 2016 (2015), 51–62. https://doi.org/10.1007/978-3-319-25518-7

[30] Marios Koniaris, George Papastefanatos, and Ioannis Anagnostopoulos. 2018. Solon: A holistic approach for modelling, managing and mining legal sources. *Algorithms* 11, 12 (2018), 1–22. https://doi.org/10.3390/a11120196

[31] Rachael Lammey. 2014. CrossRef developments and initiatives: An update on services for the scholarly publishing community from CrossRef. *Science Editing* 1, 1 (2014), 13–18. https://doi.org/10.6087/kcse.2014.1.13

[32] Christoph Lange and Angelo Di Iorio. 2014. Semantic Publishing Challenge – Assessing the Quality of Scientific Output. *Semantic Web Evaluation Challenge* 1 (2014), 61–76. https://doi.org/10.1007/978-3-319-12024-9

[33] Patrice Lopez. 2009. GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. *International conference on theory and practice of digital libraries* (2009), 473–474. https://doi.org/10.1007/978-3-642-04346-8_62

[34] Robert F. Lorch. 1989. Text-signaling devices and their effects on reading and memory processes. *Educational Psychology Review* 1, 3 (1989), 209–234. https://doi.org/10.1007/BF01320135

[35] Domhnall MacAuley. 1995. Critical appraisal of medical literature: An aid to rational decision making. *Family Practice* 12, 1 (1995), 98–103. https://doi.org/10.1093/fampra/12.1.98

[36] Inderjeet Mani. 2001. *Automatic summarization*. Vol. 3. John Benjamins Publishing.

[37] Derek Miller. 2019. Leveraging BERT for Extractive Text Summarization on Lectures. (2019).

[38] Barend Mons and Jan Velterop. 2009. Nano-publication in the e-science era. *CEUR Workshop Proceedings* 523 (2009).

[39] David Nadeau and Satoshi Sekine. 2007. A Survey on Named Entity Recognition. *Lingvisticae Investigationes* 30, 1 (2007), 3–26. https://doi.org/10.1007/978-981-13-9409-6_218

[40] Takeo Nakayama, Nobuko Hirai, Shigeaki Yamazaki, and Mariko Naito. 2005. Adoption of structured abstracts by general medical journals and format for a structured abstract. *Journal of the Medical Library Association* 93, 2 (2005), 237–242.

[41] Allard Oelen, Mohamad Yaser Jaradeh, Markus Stocker, and Sören Auer. 2020. Generate FAIR Literature Surveys with Scholarly Knowledge Graphs. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020* (2020), 97–106. https://doi.org/10.1145/3383583.3398520

[42] Heiko Paulheim. 2017. Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semantic Web* 8, 3 (2017), 489–508. https://doi.org/10.3233/SW-160218

[43] Silvio Peroni and David Shotton. 2018. The SPAR Ontologies. *International Semantic Web Conference* (2018), 119–136. https://doi.org/10.1007/978-3-030-00668-6_8

[44] Francesco Ronzano, Gerard Casamayor del Bosque, and Horacio Saggion. 2014. Semantify CEUR-WS Proceedings: towards the automatic generation of highly descriptive scholarly publishing Linked Datasets. *Communications in Computer and Information Science* 475, June (2014), V–VI. https://doi.org/10.1007/978-3-319-12024-9

[45] Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27, 2 (2015), 443–460. https://doi.org/10.1109/TKDE.2014.2327028

[46] Hiroyuki Shindo, Yohei Munesada, and Yuji Matsumoto. 2019. PDFanno: A web-based linguistic annotation tool for PDF documents. *LREC 2018 - 11th International Conference on Language Resources and Evaluation* (2019), 1082–1086.

[47] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (MAS) and applications. *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web* (2015), 243–246. https://doi.org/10.1145/2740908.2742839

[48] Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. 2008. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*. 254–263.

[49] Sasha Spala, Franck Dernoncourt, Walter Chang, and Carl Dockhorn. 2018. A Web-based Framework for Collecting and Assessing Highlighted Sentences in a Document. *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations* (2018), 78–81.

[50] Pontus Stenetorp, Sampo Pyysalo, and Goran Topi. 2012. BRAT : a Web-based Tool for NLP-Assisted Text Annotation. Figure 1 (2012), 102–107.

[51] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. *16th International World Wide Web Conference, WWW2007* (2007), 697–706. https://doi.org/10.1145/1242572.1242667

[52] 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-Sept (2015), 1–8. https://doi.org/10.1145/2814864.2814887

[53] Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The Penn Treebank: An Overview. (2003), 5–22. https://doi.org/10.1007/978-94-010-0201-1_1

[54] Mark Traquair, Ertugrul Kara, Burak Kantarci, and Shahzad Khan. 2019. Deep Learning for the Detection of Tabular Information from Electronic Component Datasheets. *Proceedings - International Symposium on Computers and Communications* 2019-June (2019), 0–5. https://doi.org/10.1109/ISCC47284.2019.8969682

[55] Thomas S Tullis and Jacqueline N Stetson. 2004. A Comparison of Questionnaires for Assessing Website Usability. *Usability Professional Association Conference* (2004), 1–12.

[56] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. https://doi.org/10.1145/2629489

[57] Thomas Weber, Heinrich Hußmann, Zhiwei Han, Stefan Matthes, Yuanting Liu, and Yuant-Ing Liu. 2020. Draw with Me: Human-in-the-Loop for Image Restoration. 20 (2020), 243–253. https://doi.org/10.1145/3377325.3377509

[58] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2020. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference* (2020), 3914–3923. https://doi.org/10.18653/v1/d19-1404