Surveying the FAIRness of Annotation Tools: Difficult to find, difficult to reuse

Ekaterina Borisova[1], **Raia Abu Ahmad**[1], **Leyla Jael Garcia-Castro**[2],
Ricardo Usbeck[3], **Georg Rehm**[1]

[1]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany,
[2]ZB MED Information Centre for Life Sciences, Germany,
[3]Leuphana Universität Lüneburg, Germany
[1]ekaterina.borisova@dfki.de, raia.abu_ahmad@dfki.de, georg.rehm@dfki.de,
[2]ljgarcia@zbmed.de, [3]ricardo.usbeck@leuphana.de

Abstract

In the realm of Machine Learning and Deep Learning, there is a need for high-quality annotated data to train and evaluate supervised models. An extensive number of annotation tools have been developed to facilitate the data labelling process. However, finding the right tool is a demanding task involving thorough searching and testing. Hence, to effectively navigate the multitude of tools, it becomes essential to ensure their *findability*, *accessibility*, *interoperability*, and *reusability* (FAIR). This survey addresses the FAIRness of existing annotation software by evaluating 50 different tools against the FAIR principles for research software (FAIR4RS). The study indicates that while being accessible and interoperable, annotation tools are difficult to find and reuse. In addition, there is a need to establish community standards for annotation software development, documentation, and distribution.

1 Introduction

Coinciding with the rise of Machine Learning (ML), annotation projects started being conducted to create labelled datasets for the training and testing of models, especially those based on supervised approaches (Ide and Pustejovsky, 2017). A wide range of software has been developed to address data labelling. Existing annotation tools vary in terms of supported modality (i. e., text, image, video, audio), access type (e. g., desktop, web-based, etc.), license (commercial vs. open-source), and annotation task(s) they are designed for (Neves and Ševa, 2019). This sheer abundance of different annotation tools hinders the search, access, and choice of an appropriate tool. Researchers often spend a lot of time downloading and installing tools that turn out to be irrelevant to their projects (Neves and Ševa, 2019).

To improve the *findability*, *accessibility*, *interoperability*, and *reusability* (FAIR) of research artifacts such as annotation tools, the FAIR (Wilkinson et al., 2016) and FAIR for Research Software (FAIR4RS, Chue Hong et al., 2022; Barker et al., 2022) principles have been proposed. Although previous surveys of annotation tools exist (Dasiopoulou et al., 2011; Neves and Leser, 2012; Nixon and Troncy, 2014; Neves and Ševa, 2019; Oliveira and Rocha, 2013; Aljabri et al., 2022), none of them evaluated their FAIRness. However, the development of FAIR annotation tools is essential to facilitate knowledge discovery and to ensure transparent research. This paper addresses this gap by assessing a range of 50 different annotation tools according to the FAIR4RS principles.

Our contributions can be summarised as follows:

- We offer interpretations of the FAIR4RS principles tailored to the specific use-case of annotation tools. These can be a valuable resource for developers and researchers while designing and reusing annotation software.
- We perform a thorough FAIRness assessment of 50 different annotation tools, providing insights into the current documentation and sharing strategies. These findings can serve as a basis for defining best practices for annotation tool management.
- Through our assessment, we provide a comparison of annotation tools, addressing their different features and adherence to community standards. This can be used by researchers as a reference while searching for appropriate tools for a specific task.
- We define ten essential functionalities that ideally should be incorporated into an annotation tool for an easier user experience.

Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII), pages 29–45
March 22, 2024 ©2024 Association for Computational Linguistics

The rest of the paper is structured as follows: Section 2 discusses the annotation lifecycle, FAIR and FAIR4RS principles in more detail. Section 3 describes the annotation tools selection process and introduces our interpretation of the FAIR4RS principles. Section 4 and Section 5 present the evaluation results and main findings, respectively. Section 6 discusses the limitations of our study. Concluding remarks are provided in Section 7.

2 Background

2.1 Annotation Lifecycle

Annotated data is fundamental for training, evaluating and validating ML models. In particular, supervised and semi-supervised algorithms directly rely on labelled data and their performance is highly dependent on the annotation quality (Hao et al., 2020; Alhazmi et al., 2021). Furthermore, in the context of transfer learning and fine-tuning, annotated data is essential for fostering the adaptation of models to specific tasks and domains (Pan and Yang, 2010).

Annotation can be performed for diverse modalities of data, i.e., text, image, audio, and video as well as at various levels depending on the data type and task at hand (Ide and Pustejovsky, 2017). For instance, in the case of text annotation, labels can be assigned to an entire document, paragraph, sentence, phrase, word or character. Annotation approaches range from manual (e. g., crowdsourcing, Vander Schee, 2009), semi-automatic (e. g., active learning, Settles, 2009) to fully automatic, relying on ML and Natural Language Processing (NLP) techniques. Each method has its advantages and drawbacks, and the choice usually depends on specific project goals, data, and resources.

Annotation is a complex process which usually involves a wide range of activities such as collecting the data, preparing an annotation schema and guidelines, recruiting and training annotators, curating the assigned labels, and computing inter-annotator agreement (IAA) scores (Ide and Pustejovsky, 2017). To facilitate the annotation lifecycle (Rehm, 2016), various tools have been developed which deal with the data labelling stage. These tools vary in complexity and functionality, ranging from simple *desktop* interfaces, such as TagEditor[1] and ELAN (Wittenburg et al., 2006), to advanced *web-based* applications, such as INCEpTION (Klie et al., 2018) and Doccano (Nakayama et al., 2018), supporting teams, user roles, automatic IAA calculation, ML models, etc. Annotation tools also come in various types ranging from those tailored towards *specific domains*, e. g., MedTag (Giachelle et al., 2021) and BioQRator (Kwon et al., 2013), *modalities*, e. g., ELAN and Annotation Web (Smistad et al., 2021), and *tasks*, e. g., PDF sentence annotator[2] and Praat (Boersma and Weenink, 2023), to *general-purpose* applications, e. g., prodigy[3] and Label Studio (Tkachenko et al., 2020-2022), also see Rehm (2020).

2.2 FAIR and FAIR4RS

Sharing research data is essential for accelerating scientific progress as it encourages collaborative research and decision-making. However, research data management techniques vary greatly across disciplines leading to inconsistencies in documentation and sharing of scientific artifacts (Akers and Doty, 2013). Such heterogeneous and disjoint data management practices hinder the validation, replication, and improvement of previous solutions. Given the rapid progress in ML and Artificial Intelligence coupled with the ever increasing number of new datasets, models, and software, it has become crucial to define common data sharing policies to ensure transparency and reproducibility.

In order to promote the *findability*, *accessibility*, *interoperability*, and *reuse* of scholarly data from both human and machine perspectives, the FAIR guiding principles (Wilkinson et al., 2016) were proposed. This set of principles is meant to be directly applied to all digital objects such as datasets, algorithms, software, and toolkits. However, several studies (Patel et al., 2023; Katz et al., 2016) demonstrated that the FAIR principles are not fully applicable to research software (RS). As was highlighted by Katz et al. (2016), even though data and software share certain characteristics, e. g., potential for having a license or a Digital Object Identifier (DOI), these two digital objects possess several significant differences. In contrast to data, software is an inherently executable and continuously evolving object characterised by a composite structure as it is frequently developed based on other components. Unlike data, software requires maintenance due to its dependency on other packages, tools, and software which are subject to constant change. Software also tends to have a shorter lifespan than

[1] http://tinyurl.com/TagEditor

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

[3] https://prodi.gy

data due to technological progress.

In response to the need for software-specific principles, the original FAIR principles have been revised several times (Lamprecht et al., 2020; Katz et al., 2021). As a result of those efforts, the FAIR4RS principles geared towards ensuring a FAIR lifecycle of RS were developed (Barker et al., 2022; Chue Hong et al., 2022). According to these principles, software should be thoroughly described through metadata, it should be possible to execute, replicate, combine, reinterpret, reimplement, and expand upon it as well as to utilise it in diverse settings (Chue Hong et al., 2022).

The FAIR and FAIR4RS principles gave rise to a range of tools for the FAIRness evaluation of digital objects. Those include manual questionnaires and checklists (Do I-PASS for FAIR de Bruin et al., 2020, FAIR Data Self Assessment Tool[4], FAIR Aware[5]) as well as automated tests (FAIR Evaluation Services Wilkinson et al., 2019, howfairis Spaaks et al., 2022, FAIR Enough[6], FAIR-Checker Gaignard et al., 2023). However, since both FAIR and FAIR4RS are open to interpretation, the assessment results can vary depending on the tool.

3 Methods

In this study, we performed a *manual* assessment of annotation tools against the FAIR4RS principles. We refrained from using automatic solutions primarily due to the variability in results mentioned in Section 2.2. Furthermore, *howfairis* is the only tool based on FAIR4RS, designed specifically for analysing the compliance of GitHub/GitLab repositories with the principles. Consequently, its applicability is limited to tools hosted on those platforms.

Due to the huge amount of existing tools (Neves and Ševa, 2019) and time constraints, we limited the evaluation to 50 annotation tools. As a first step, we randomly selected annotation tools surveyed by Neves and Ševa (2019). However, those are specifically developed for text data annotation. To make the set of tools more diverse in terms of covered modalities, we conducted a search on Google Scholar to find publications related to annotation and corpus creation that mention or cite annotation software. In addition, we looked for tools on platforms such as European Language Grid (ELG, Rehm, 2023), Zenodo[7], SourceForge[8] and Software Heritage[9]. We did not consider tools that were archived or have become part of another project (e. g., WebAnno, Eckart de Castilho et al., 2016), were not found (e. g., a publication exists but the link to the home page or source code does not work) or are for crowd-sourcing purposes (Amazon Mechanical Turk[10]).

Since the FAIR4RS principles do not serve as a set of strict rules but rather as a guideline, they are not rigidly defined, sometimes allowing for a broad range of interpretations. Therefore, below we introduce our interpretations and the evaluation strategies defined and followed in this study[11].

F1. Software is assigned a globally unique and persistent identifier. An annotation tool should have a globally unique and persistent identifier (PID), such as DOI, which assures longevity and consistently points to the software despite changes in its location, content or other attributes. Thus, we investigated whether a tool is available on platforms that provide PIDs, i. e., Software Heritage and Zenodo. Other widely utilised software publishing services, such as GitHub, GitLab, or SourceForge, are not suitable as the URLs they offer cannot be considered persistent.

F1.1. Components of the software representing levels of granularity are assigned distinct identifiers. In addition to the annotation tool itself, distinct PIDs should be assigned to all its components. Thus, following the software granularity levels schema offered by Chue Hong et al. (2022), we researched whether files, directories, commits, releases, and other tool attributes possess PIDs. In contrast to F1, we considered only Software Heritage since it assigns distinct PIDs to every digital object component compared to Zenodo, which provides individual DOIs only for various versions.

F1.2. Different versions of the software are assigned distinct identifiers. Each release of an annotation tool should be assigned a distinct PID allowing users to track its development and refer to a specific version they utilised. Similar to F1, we checked the presence of tools on both Software Heritage and Zenodo.

F2. Software is described with rich metadata.

[4] https://ardc.edu.au/resource/fair-data-self-assessment-tool/

[5] https://fairaware.dans.knaw.nl

[6] https://fair-enough.semanticscience.org

[7] https://zenodo.org

[8] https://sourceforge.net

[9] https://www.softwareheritage.org

[10] https://www.mturk.com

[11] The interpretations of the FAIR4RS principles are also summarised in Appendix A.

Metadata should be semantically structured, i. e., being both human and machine-readable. It should contain a thorough description of an annotation tool allowing users to understand how to utilise and replicate it without looking into its source code. Metadata is considered to be rich when it goes beyond basic information. To define the minimum metadata we followed the Bioschemas ComputationalTool[12]. Bioschemas is an effort to improve findability in Life Sciences by relying on the widely used Schema.org[13] vocabulary. Although Bioschemas is domain-specific, it includes general types and properties to describe research artifacts such as datasets and software. Unlike other vocabularies, Bioschemas offers minimum, recommended, and optional property types, making it easier to define rich metadata. Accordingly, the minimum metadata of an annotation tool should include *name*, *URL*, and *description*. The metadata is considered to be rich if at least one additional property from any marginality level is provided[14].

F3. Metadata clearly and explicitly include the identifier of the software they describe. If an annotation tool is assigned a PID, it should be referenced by it in the respective structured metadata. Thus, in case F1 is not fulfilled, F3 fails as well.

F4. Metadata are FAIR, searchable and indexable. Metadata is FAIR when it is semantically structured. Therefore, if an annotation tool fails F2 due to the lack of structured metadata, it automatically fails F4. Any metadata exposed via web pages in a format understood by search engines or deposited in a repository/registry with search functionality is indexable.

A1. Software is retrievable by its identifier using a standardised communications protocol. An annotation tool should be accessed through a commonly used communication protocol such as Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS) or File Transfer Protocol (FTP).

A1.1. The protocol is open, free, and universally implementable. There should be no restrictions and fees to implement the communication protocol.

A1.2. The protocol allows for an authentication and authorisation procedure, where necessary. The protocol should include mechanisms to verify the identity of users and to determine their access rights where necessary. Authentication and authorisation are supported by HTTP/HTTPS and FTP protocols, therefore if a tool is retrievable via those, it automatically fulfills A1.2.

A2. Metadata are accessible, even when the software is no longer available. As software tend to be deprecated over time, ideally structured metadata should be published separately with its own PID. In practice, it is often embedded into the source code of software. Therefore, this principle is satisfied when metadata is assigned a distinct PID and published either separately from a tool or along with it on an archive which ensures longevity.

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards. While file conversion is possible, RS support for standard formats is more user-friendly (Ide and Pustejovsky, 2017). In the context of annotation, this allows the reuse of labelled data across the tools, e. g., for error corrections, active learning or automatic predictions (Neves and Ševa, 2019). Currently, there do not seem to be well-defined standards for annotation tools' input/output formats. The formats vary depending on the input modality, domain, and specific task at hand. For instance, *CoNLL* is widely used in linguistic annotation projects (Ide and Pustejovsky, 2017), while *Dicom* is commonly utilised in medical imaging (Larobina and Murino, 2014; Aljabri et al., 2022). Therefore, to evaluate I1, we relied on formats mentioned in previous surveys on annotation tools (Neves and Ševa, 2019; Oliveira and Rocha, 2013; Dasiopoulou et al., 2011; Aljabri et al., 2022) and by Ide and Pustejovsky (2017)[15]. We searched for input/output format information in both structured and unstructured (e. g., README) metadata. The principle is considered to be fully fulfilled if an annotation tool supports at least one of the standard formats for both input and output.

I2. Software includes qualified references to other objects. This principle calls for references to any objects other than software such as datasets, hardware, programming language, operating system or browser. Qualified references include identifiers (URLs, PIDs, etc.) and controlled vocabularies. We investigated whether such references are provided in structured or unstructured metadata.

R1. Software is described with a plurality of

[12] `https://bioschemas.org/profiles/ComputationalTool/1.0-RELEASE`

[13] `https://schema.org`

[14] For the full list of structured and unstructured metadata sources per annotation tool, please see Appendix B.

[15] For the full list of formats, please see Appendix C.

accurate and relevant attributes.

An annotation tool should be described in terms of metadata categories (F2), license (R1.1), and provenance (R1.2). The relevance of attributes is usually determined by repositories and/or communities that create and use a tool. Whenever feasible, multiple terms for the same, similar or overlapping concepts should be provided to allow reuse. However, to the best of our knowledge, there are no community-agreed standards for the metadata vocabulary of annotation tools. Therefore, R1 is considered to be fulfilled if a tool fully adheres to F2, R1.1 and R1.2 and partially fulfilled if one of the principles is partially met.

R1.1. Software is given a clear and accessible license. The annotation tool's license should be clearly stated in either structured or unstructured metadata.

R1.2. Software is associated with detailed provenance. This principle calls for an explanation of the annotation tool's origins and development history. To this end, we evaluated whether structured or unstructured metadata provides answers to the following questions: *Why and how a tool came to be? Who contributed what, when, and where? How to cite a tool?* The principle is fully satisfied when all questions receive complete answers. However, in case some questions are only partially addressed, the principle is considered partially met.

R2. Software includes qualified references to other software. As with any other software, annotation tools usually have dependencies. Thus, in contrast to I2, we researched whether qualified references to other software (e. g., libraries, packages) are provided in either structured or unstructured metadata.

R3. Software meets domain-relevant community standards. Currently, there do not seem to be well-established community standards for annotation tools. The desired capabilities of software are influenced by the annotation project scope and goals. According to Ide and Pustejovsky (2017), there are two main requirements affecting the choice of a tool: Support for *custom schemas* and *multiple languages*. We defined eight additional criteria that could ease the annotation workflow, thus being potentially important for any project (Ide and Pustejovsky, 2017; Neves and Ševa, 2019). First, since annotation usually involves a team of experts, a tool should be *web-based* supporting *teams and roles* to enable remote collaboration and user rights settings (Ide and Pustejovsky, 2017). Second, following I1, it is essential that an annotation software reads and writes using *standard file formats*. Furthermore, a tool should support *importing/exporting multiple file formats* to allow integrating annotations with existing datasets or directly importing/exporting of data in a desired format. An application should also offer *document-level annotation* as document classification is one of the core NLP tasks. Additionally, given that the same object or entity can belong to various categories, support for *multi-label annotation* should be available. Built-in automatic *IAA score calculation* should be provided as well since it is fundamental for any annotation project. Finally, data under annotation can be sensitive (e. g., patient data) requiring certain privacy measures. Therefore, a tool should guarantee *data privacy* by possessing authentication and authorisation features along with the local installation option.

To test whether a tool complies with R3, we searched for information on the described features in both structured and unstructured metadata. The principle is fully fulfilled when all 10 community standards are met. Partial fulfilment is considered when a tool meets an established threshold based on the average number of fulfilled standards across the tools[16]. The threshold is equal to 6, thus a tool that meets less than 6 of the defined community standards fails R3.

4 Results

We present an assessment of 50 annotation tools based on the 17 FAIR4RS principles. The complete list of tools and the results are provided in Table 1. The annotation tools vary in terms of their complexity, license, supported features, and modalities (see Table 2).

According to our results, none of the tools fully adhere to all of the 17 principles. The maximum number of fulfilled principles across the tools is 13, while the minimum is 3. Out of 50 tools, 6 have reached the maximum of 13 and only one fulfills 3 principles. On average, tools comply with approximately 9 principles.

When it comes to *findability*, our analysis reveals that 29 annotation tools satisfy the requirements for F1, while the rest fall short as they are hosted on

[16] Note that a community standard is satisfied only when a tool fully supports a feature.

Tools	F1	F1.1	F1.2	F2	F3	F4	A1	A1.1	A1.2	A2	I1	I2	R1	R1.1	R1.2	R2	R3
INCEpTION (Klie et al., 2018)	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	⯪	✓	⯪	✓	⯪
brat (Stenetorp et al., 2012)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	⯪	✓	⯪
Doccano (Nakayama et al., 2018)	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	⯪	✓	⯪	✓	⯪
BioQRator (Kwon et al., 2013)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
Catma (Gius et al., 2023)	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	⯪	✓	⯪
Djangology (Apostolova et al., 2010)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	⯪
ezTag (Kwon et al., 2018)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✗
FLAT	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	⯪
LightTag (Perry, 2021)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✗	✗	✗	✗	✗	⯪
MAT	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	⯪
PDFAnno (Shindo et al., 2018)	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗
prodigy	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	⯪
TextAE	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗
WAT-SL (Kiesel et al., 2017)	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗
Hypothesis	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	⯪	✓	✗	✓	✗	✓	✗
Haystack	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	⯪
PDF sentence annotator	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
PAWLS (Neumann et al., 2021)	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
TeamTat (Islamaj et al., 2020)	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	⯪
TagEditor	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	⯪
TS-ANNO (Stodden and Kallmeyer, 2022)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
MedTator (He et al., 2022)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
DocTAG (Giachelle et al., 2022)	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	⯪	✓	✓
PubTator (Wei et al., 2013)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✗
Ellogon (Ntogramatzis et al., 2022)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	⯪
Markup	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	⯪	✓	✗	✗	✗	✓	✗
Label Studio	✗	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	⯪
MedTag (Giachelle et al., 2021)	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	⯪	✓	⯪
BAT (Meléndez-Catalán et al., 2017)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✓	✗	✓	✗	✓	✗
Seshat (Titeux et al., 2020)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✓	✗	✓	✗	✓	✗
VIA (Dutta and Zisserman, 2019)	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗
Potato (Pei et al., 2023)	✗	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	⯪
Annotation Web (Smistad et al., 2021)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✓	✗	✓	✗	✓	✗
audino (Grover et al., 2020)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✓	✗	✓	✗	✓	✗
MATILDA	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✓	✗	✓	✗	✓	⯪
ELAN (Wittenburg et al., 2006)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	⯪	✓	✗
Praat (Boersma and Weenink, 2023)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
Pundit	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	⯪	✓	✗	✓	✗	✓	✗
UAM CorpusTool (O'Donnell, 2008)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✗
TIARA (Putra et al., 2020)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
COCO Annotator (Brooks, 2019)	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	⯪	✓	✗	✓	⯪	✓	✗
Gate Teamware (Karmakharm et al., 2023)	✓	✗	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✓
ActiveAnno (Wiechmann et al., 2021)	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	⯪
YEDDA (Yang et al., 2018)	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	⯪	✓	✗
Textinator (Kalpakchi and Boye, 2022)	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	⯪	✓	⯪	✓	⯪
Argilla (Vila-Suero and Aranda, 2023)	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	⯪	✓	⯪	✓	⯪
Orbis Annotator (Süsstrunk et al., 2023)	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
CVAT (Corporation, 2023)	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	⯪
DataGym.ai	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗
DeepLabel (Veitch-Michaelis, 2021)	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗

Table 1: **Annotation tools assessment results according to FAIR4RS principles.**
Symbols: ✓ = fulfills principle, ✗ = does not fulfill principle, ⯪ = partially fulfills principle.

platforms such as GitHub, GitLab[17], SourceForge, ELG, Hugging Face[18] or the official project website. The same set of tools except one adheres to F1.1. However, the number of applications meeting the criteria for F1.2 is considerably less and is equal to 14. This is due to cases where one version of a tool is published on Software Heritage/Zenodo but other releases are available on different platforms like GitHub. Only 13 tools adhere to F2, and the most frequently provided additional metadata for those is *dependencies*, *license*, *author*, and *version*. Several tools (8) fail the principle as they lack some of the required metadata, most commonly description and/or URL. Other software, with the exception of DataGym.ai, do not have structured metadata at all. None of the annotation tools comply with F3 due to one of the following reasons: 1. no semantically structured metadata is available, 2. the tool fails to meet F1, 3. PID exists but is not referenced in the metadata. Finally, less than half of the tools (22) have semantically structured metadata, and thus satisfy F4.

In terms of *accessibility*, all annotation tools fulfill A1-1.2 as they are retrievable without any restrictions via HTTP/HTTPs. However, only 14 tools comply with the A2 principle as they are available on Software Heritage. It is worth noting that only INCEpTION comes with structured metadata, published separately from the software on ELG.

Most tools fully support *interoperability* as they

[17] https://about.gitlab.com
[18] https://huggingface.co

Tools	Modality	License	Web-based	Custom schemas	Multiple languages	Users and roles	Standard file formats	Multiple file formats	Document-level annotation	Overlapping labels	IAA	Data privacy
INCEpTION	🗎	Apache-2.0	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
brat	🗎	MIT	✓	✓	✓	⯪	✓	✗	✗	✓	✗	✓
Doccano	🗎	MIT	✓	✓	✓	✓	✓	⯪	✓	✓	✗	✓
BioQRator	🗎	Apache-2.0	✓	✓	✗	⯪	✓	✓	✗	✗	✗	✗
Catma	🗎	GNU GPL-3.0	✓	✓	✓	✓	✓	✓	⯪	✓	✗	✗
Djangology	🗎	–	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓
ezTag	🗎	–	✓	✓	✗	⯪	✓	✗	✗	✓	✗	✓
FLAT	🗎	GNU GPL-3.0	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓
LightTag	🗎	–	✓	✓	✓	✓	⯪	⯪	✓	✓	✓	✓
MAT	🗎	BSD	✓	✓	✓	✗	✓	⯪	✓	✓	✗	✗
PDFAnno	🗎	MIT	✓	✗	✓	⯪	✓	✗	✗	✗	✗	✓
prodigy	🗎 📹 🎙 🖼	–	✓	✓	✓	⯪	✓	⯪	✓	✓	✓	✓
TextAE	🗎	MIT	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓
WAT-SL	🗎	MIT	✓	✓	✗	⯪	✓	✗	✗	✗	✗	✓
Hypothesis	🗎	BSD-2-Clause	✓	✗	✗	⯪	⯪	⯪	✗	✗	✗	✗
Haystack	🗎	Apache-2.0	✓	✓	✓	✓	✓	⯪	✗	✓	✗	✓
PDF sentence annotator	🗎	Apache-2.0	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗
PAWLS	🗎	Apache-2.0	✓	✗	✗	✗	✓	✗	✗	✓	⯪	✗
TeamTat	🗎	MIT	✓	✓	✓	✓	✓	⯪	✗	✓	✓	✗
TagEditor	🗎	MIT	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓
TS-ANNO	🗎	GNU GPL-3.0	✓	✗	✓	✓	✓	⯪	✗	✗	✓	✓
MedTator	🗎	Apache-2.0	✓	✓	✗	✗	✓	✓	✓	✗	✓	✗
DocTAG	🗎	MIT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PubTator	🗎	–	✓	✗	✗	✗	✓	⯪	✗	✗	✗	✗
Ellogon	🗎	GNU LGPL-3.0	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓
Markup	🗎	–	✓	✓	✗	⯪	⯪	✗	✗	✓	✗	✓
Label Studio	🗎 📹 🎙 🖼	Apache-2.0	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
MedTag	🗎	MIT	✓	✓	✓	✓	✓	⯪	✓	✓	✓	✓
BAT	🎙	GNU AGPL-3.0	✓	✓	✗	✓	⯪	✗	✗	⯪	✗	✓
Seshat	🎙	EUPL-1.2	✓	✓	✗	✓	⯪	⯪	✗	✗	✓	✓
VIA	📹 🎙 🖼	BSD-2-Clause	✓	✗	✗	✗	✓	⯪	✗	✗	✗	✓
Potato	🗎 📹 🖼	Polyform Shield	✓	✓	✓	⯪	✓	✓	✓	✗	✗	✓
Annotation Web	🖼	MIT	✓	✗	✗	✓	⯪	✗	✓	✗	✗	✓
audino	🎙	MIT	✓	✓	✓	✓	⯪	✗	✗	✗	✗	✓
MATILDA	🗎	GNU GPL-2.0	✓	✓	✓	✓	⯪	✗	✓	✗	✗	✓
ELAN	📹 🎙	GPL-3.0	✗	✓	✓	✗	✓	⯪	✗	✗	✗	✓
Praat	🎙	GNU GPL	✗	✓	✓	✗	✓	⯪	✗	✗	✗	✓
Pundit	🗎	GNU AGPL-3.0	✓	✗	✗	⯪	⯪	⯪	✗	✗	✗	✗
UAM CorpusTool	🗎	–	✗	✓	✗	✗	✓	✗	✓	✓	✗	✓
TIARA	🗎	MIT	✓	✓	✗	✗	✓	⯪	✗	✗	✗	✓
COCO Annotator	🖼	MIT	✓	✓	✗	✓	⯪	⯪	✗	✓	✗	✓
Gate Teamware	🗎	GNU AGPL-3.0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ActiveAnno	🗎	MIT	✓	✗	✓	✓	✓	✗	✓	✗	✓	✓
YEDDA	🗎	Apache-2.0	✗	✗	✓	⯪	✓	✗	✗	✗	✗	✓
Textinator	🗎	GNU AGPL-3.0	✓	✓	✓	✓	✓	⯪	✓	✗	✗	✓
Argilla	🗎	Apache-2.0	✓	✓	✓	✓	✓	⯪	✓	✓	✗	✓
Orbis Annotator	🗎	Apache-2.0	✓	✗	✗	⯪	✓	✓	✗	✗	✗	✓
CVAT	📹 🖼	MIT	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓
DataGym.ai	📹 🖼	MIT	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓
DeepLabel	📹 🖼	MIT	✗	✓	✗	✗	✓	✓	✗	✓	✗	✓

Table 2: **Evaluation criteria for FAIR4RS principle R3 along with details on modality and license.** Symbols: 🗎 = text, 📹 = video, 🎙 = audio, 🖼 = image, ✓ = fulfills criterion, ✗ = does not fulfill criterion, ⯪ = partially fulfills criterion.

fulfill both I1 and I2. In particular, 40 annotation tools support standard data formats for both input and output. Only 10 tools partially adhere to I1, most of which (9 tools) do not provide details on either input or output formats, and one application (Pundit) writes data into notebooks that are not considered standard. All but one annotation software (LightTag) fulfill I2 by offering qualified reference to other objects in either structured JSON/XML metadata files or in unstructured documentation using standard naming conventions. Commonly referenced objects are programming languages, compatible browsers, and datasets.

Assessing *reusability*, we note that none of the tools fully comply with R1 as they fail F2 or R1.2. There are 11 tools that achieve partial fulfillment. The majority of tools (43) provide a clear description of the license in either semantically structured or unstructured metadata, thereby adhering to R1.1. However, only 12 tools offer license information in both metadata types. Most annotation tools (31) specify license in unstructured README/LICENSE files or on the project webpage. Doccano, Textinator, and Gate Teamware do not update license information in a structured metadata file. In terms of provenance, only VIA fully complies with R1.2. There are 11 tools that partially adhere to the principle since they provide citation information and details on origin, development history, and/or contributors/authors. The remaining software either have minimal provenance, limited to release history or authors/contributors or lack it entirely. Similar to the I2 principle, only LighTag fails to satisfy the criteria for R2.

Finally, when it comes to R3, results indicate that only DocTAG and Gate Teamware fully fulfil it. Among the rest of the tools, 21 have partial compliance and 27 fail the principle. Table 2 presents an overview of the results per application and community standard. As can be seen, most of the tools (44) are web-based. However, only 25 of those support users and roles, while 12 allow collaborative work but do not offer user rights functionality. For the remaining 13 tools, either no information on this feature was available or it was clear from the software architecture that there is no support for teams and roles (e. g., ELAN, PDF sentence annotator). A vast majority of the tools (37) allow custom ontologies/schemas and about half of them (27) are compatible with several languages. A limited number of annotation tools (13) offer data import and export in multiple file formats, while 19 applications partially fulfill the criteria as details on either only input or output were found. As was already discussed above, 40 tools support standard file formats. Table 2 also demonstrates that 19 tools allow document-level annotation. Only Catma partially supports this feature since a user has to manually highlight the whole text. Overlapping labels functionality is available for almost half of the tools (24). BAT is a single tool that partially meets the criteria as the overlaps have to be resolved before finishing annotating. Less than half of tools (14) include built-in IAA calculation. In the case of PAWLS, IAA is available but not integrated into the tool (the score can be computed separately through the command line interface). Hence, we categorise it as a partial fulfillment. Finally, 40 tools ensure data privacy since they either allow local installation and/or require a user to log into the system.

5 Discussion

The FAIRness assessment indicates that while annotation tools are accessible and interoperable, there is a strong need for improving their findability and reusability. In particular, a large number of tools lack PIDs on various levels (files, releases, etc.). Even in cases when an annotation software possesses a PID, there is no reference to it in the respective metadata. These factors hinder the accurate and unambiguous citation of a tool as well as tracking its developmental changes over time.

The results also show that the vast majority of annotation software suffers from the absence of semantically structured metadata. As was noted in Section 4, even if there is one, some details such as license or input/output data formats tend to be provided in unstructured formats. Furthermore, such tools do not have structured metadata published separately either. Consequently, tools are less discoverable and linkable to other related systems. It becomes difficult to reuse and replicate these annotation applications without delving into the implementation details or testing them. The situation is even worse when it comes to provenance descriptions. Most tools do not have fine-grained documentation of their origin and development history. The lack of sufficient provenance information contributes to low reproducibility and makes it difficult to build upon existing annotation tools.

Additionally, there is a clear need for agreed-upon community standards and best practices regarding annotation tools functionalities, metadata vocabulary and import/export file formats. The absence of those influenced the assessment results.

6 Limitations

While our analysis is rather comprehensive, it is not without limitations. First, the manual approach to evaluation is error prone. It would be beneficial to align our findings with the results from automatic FAIRness assessment solutions, namely using howfairis (at least for tools hosted on GitHub/GitLab). Second, tools were not tested with respect to their executability. Thus, there is no guarantee that all surveyed tools can actually be installed and run properly. This also means that a tool could possibly have a specific feature but it is not stated in its metadata. Third, when it comes to annotation tools, not only the software itself should be FAIR but the labelled data it produces should be FAIR, too. However, the FAIRness evaluation of annotated data produced by or with annotation tools is out of scope with regard to this study and we leave it for future work. Finally, as previously noted, the FAIR4RS principles are aspirational in nature. Thus, the interpretations defined in this paper should be treated as initial suggestions rather than rigid definitions.

7 Conclusion

In this paper, we investigated how annotation tools comply with the FAIR4RS principles. We performed a manual evaluation of 50 tools following interpretations of the FAIR4RS principles adapted specifically to annotation software. The findings reveal that the findability and reusability of annotation tools require improvement. Specifically, the lack of PIDs, semantically structured metadata and detailed provenance are the most problematic aspects. Additionally, the study shows that there is a demand for agreed-upon community standards for annotation software management.

Ethics Statement

No private or sensitive data was used, stored or shared during this study.

Acknowledgements

The research presented in this article was supported by the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[19] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.). The NFDI is funded by the Federal Republic of Germany and its states. The paper received funding through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). The authors wish to thank both for funding and support. A special thanks goes to all institutions and actors engaging for the association and its goals.

References

Katherine Akers and Jennifer Doty. 2013. Disciplinary differences in faculty research data management practices and perspectives. *International Journal of Digital Curation*, 8(2):5–26.

Khaled Alhazmi, Walaa Alsumari, Indrek Seppo, Lara Podkuiko, and Martin Simon. 2021. Effects of annotation quality on model performance. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 063–067.

Manar Aljabri, Manal AlAmir, Manal AlGhamdi, Mohamed Abdel-Mottaleb, and Fernando Collado-Mesa. 2022. Towards a better understanding of annotation tools for medical imaging: A survey. *Multimedia Tools and Applications*, 81(18):25877–25911.

Emilia Apostolova, Sean Neilan, Gary An, Noriko Tomuro, and Steven Lytinen. 2010. Djangology: A light-weight web-based tool for distributed collaborative text annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Michelle Barker, Neil P. Chue Hong, Daniel S. Katz, Anna-Lena Lamprecht, Carlos Martinez-Ortiz, Fotis Psomopoulos, Jennifer Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, and Tom Honeyman. 2022. Introducing the FAIR principles for research software. *Scientific Data*, 9(622).

Paul Boersma and David Weenink. 2023. Praat: Doing phonetics by computer [computer program].

Justin Brooks. 2019. COCO annotator.

Neil P. Chue Hong, Daniel S. Katz, Michelle Barker, Anna-Lena Lamprecht, Carlos Martinez, Fotis E. Psomopoulos, Jen Harrow, Leyla Jael Castro, Morane Gruenpeter, Paula Andrea Martinez, Tom Honeyman, Alexander Struck, Allen Lee, Axel Loewe, Ben van Werkhoven, Catherine Jones, Daniel Garijo, Esther Plomp, Francoise Genova, Hugh Shanahan, Joanna Leng, Maggie Hellström, Malin Sandström, Manodeep Sinha, Mateusz Kuzak, Patricia Herterich, Qian Zhang, Sharif Islam, Susanna-Assunta Sansone, Tom Pollard, Udayanto Dwi Atmojo, Alan Williams, Andreas Czerniak, Anna Niehues, Anne Claire Fouilloux, Bala Desinghu, Carole Goble, Céline Richard, Charles Gray, Chris Erdmann, Daniel Nüst, Daniele Tartarini, Elena Ranguelova, Hartwig Anzt, Ilian Todorov, James McNally, Javier Moldon, Jessica Burnett, Julián Garrido-Sánchez, Khalid Belhajjame, Laurents Sesink, Lorraine Hwang, Marcos Roberto Tovani-Palone, Mark D. Wilkinson, Mathieu Servillat, Matthias Liffers, Merc Fox, Nadica Miljković, Nick Lynch, Paula Martinez Lavanchy, Sandra Gesing, Sarah Stevens, Sergio Martinez Cuesta, Silvio Peroni, Stian Soiland-Reyes, Tom Bakker, Tovo Rabemanantsoa, Vanessa Sochat, Yo Yehudi, and RDA FAIR4RS WG. 2022. FAIR principles for research software (FAIR4RS principles). Zenodo.

CVAT.ai Corporation. 2023. Computer vision annotation tool (CVAT). Zenodo.

Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. 2011. A survey of semantic image and video annotation tools. In Georgios Paliouras, Constantine D. Spyropoulos, and George Tsatsaronis, editors, *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution: Bridging the Semantic Gap*, pages 196–239. Springer Berlin Heidelberg, Berlin, Heidelberg.

Taco de Bruin, Sarah Coombs, Jutta de Jong, Irene Haslinger, Henk van den Hoogen, Frans Huigen, Mijke Jetten, Jacko Koster, Margriet Miedema, Sjef Öllers, Inge Slouwerhof, Ingeborg Verheul, and Jacquelijn Ringersma. 2020. Do I-PASS for FAIR. A self assessment tool to measure the FAIR-ness of an organization. Zenodo.

Abhishek Dutta and Andrew Zisserman. 2019. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*. ACM.

[19] https://www.nfdi4datascience.de

Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan. The COLING 2016 Organizing Committee.

Alban Gaignard, Thomas Rosnet, Frédéric De Lamotte, Vincent Lefort, and Marie-Dominique Devignes. 2023. FAIR-Checker: Supporting digital resource findability and reuse with knowledge graphs and semantic web standards. *Journal of Biomedical Semantics*, 14(7).

Fabio Giachelle, Ornella Irrera, and Gianmaria Silvello. 2021. Medtag: A portable and customizable annotation tool for biomedical documents. *BMC Medical Informatics and Decision Making*, 21.

Fabio Giachelle, Ornella Irrera, and Gianmaria Silvello. 2022. Doctag: A customizable annotation tool for ground truth creation. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part II*, volume 13186 of *Lecture Notes in Computer Science*, pages 288–293. Springer.

Evelyn Gius, Jan Christoph Meister, Malte Meister, Marco Petris, Mareike Schumacher, and Dominik Gerstorfer. 2023. Catma. Zenodo.

Manraj Singh Grover, Pakhi Bamdev, Yaman Kumar Singla, Mika Hama, and Rajiv Ratn Shah. 2020. audino: A modern annotation tool for audio and speech. *arXiv*, abs/2006.05236.

Degan Hao, Lei Zhang, Jules Sumkin, Aly Mohamed, and Shandong Wu. 2020. Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance. *IEEE Journal of Biomedical and Health Informatics*, 24(9):2701–2710.

Huan He, Sunyang Fu, Liwei Wang, Sijia Liu, Andrew Wen, and Hongfang Liu. 2022. Medtator: A serverless annotation tool for corpus development. *Bioinformatics*, 38(6):1776–1778.

Nancy Ide and James Pustejovsky. 2017. *Handbook of Linguistic Annotation*. Heidelberg: Springer.

Rezarta Islamaj, Dongseop Kwon, Sun Kim, and Zhiyong Lu. 2020. Teamtat: A collaborative text annotation tool. *Nucleic Acids Research*, 48(W1):W5–W11.

Dmytro Kalpakchi and Johan Boye. 2022. Textinator: An internationalized tool for annotation and human evaluation in natural language processing and generation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 856–866, Marseille, France. European Language Resources Association.

Twin Karmakharm, David Wilby, Ian Roberts, and Kalina Bontcheva. 2023. GATE teamware 2. Zenodo.

Daniel S. Katz, Morane Gruenpeter, and Tom Honeyman. 2021. Taking a fresh look at FAIR for research software. *Patterns*, 2(3):100222.

Daniel S. Katz, Kyle E. Niemeyer, Arfon M. Smith, William L. Anderson, Carl Boettiger, Konrad Hinsen, Rob Hooft, Michael Hucka, Allen Lee, Frank Löffler, Tom Pollard, and Fernando Rios. 2016. Software vs. data in the context of citation. *PeerJ Preprints*, 4(e2630v1).

Johannes Kiesel, Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. 2017. WAT-SL: A customizable web annotation tool for segment labeling. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 13–16, Valencia, Spain. Association for Computational Linguistics.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9.

Dongseop Kwon, Sun Kim, Soo Yong Shin, and John Wilbur. 2013. Bioqrator: A web-based interactive biomedical literature curating system.

Dongseop Kwon, Sun Kim, Chih-Hsuan Wei, Robert Leaman, and Zhiyong Lu. 2018. eztag: Tagging biomedical concepts via interactive learning. *Nucleic Acids Research*, 46(W1):W523–W529.

Anna-Lena Lamprecht, Leyla Garcia, Mateusz Kuzak, Carlos Martinez, Ricardo Arcila, Eva Martin Del Pico, Victoria Dominguez Del Angel, Stephanie van de Sandt, Jon Ison, Paula Andrea Martinez, Peter McQuilton, Alfonso Valencia, Jennifer Harrow, Fotis Psomopoulos, Josep Ll. Gelpi, Neil Chue Hong, Carole Goble, and Salvador Capella-Gutierrez. 2020. Towards FAIR principles for research software. *Data Science*, 3(1):37–59.

Michele Larobina and Loredana Murino. 2014. Medical image file formats. *Journal of Digital Imaging*, 27:200–206.

Blai Meléndez-Catalán, Emilio Molina, and Emilia Gómez. 2017. Bat: An open-source, web-based audio events annotation tool. In *Proceedings of the 3rd Web Audio Conference*.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from `https://github.com/doccano/doccano`.

Mark Neumann, Zejiang Shen, and Sam Skjonsberg. 2021. PAWLS: PDF annotation with labels and structure. *arXiv*.

Mariana Neves and Ulf Leser. 2012. A survey on annotation tools for the biomedical literature. *Briefings in Bioinformatics*, 15(2):327–340.

Mariana Neves and Jurica Ševa. 2019. An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1):146–163.

Lyndon Nixon and Raphaël Troncy. 2014. Survey of semantic media annotation tools for the web: Towards new media applications with linked media. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 100–114, Cham. Springer International Publishing.

Alexandros Fotios Ntogramatzis, Anna Gradou, Georgios Petasis, and Marko Kokol. 2022. The ellogon web annotation tool: Annotating moral values and arguments. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3442–3450, Marseille, France. European Language Resources Association.

Michael O'Donnell. 2008. The UAM CorpusTool: Software for corpus annotation and exploration. In Carmen M. Bretones Callejas, José Francisco Fernández Sánchez, José Ramón Ibáñez Ibáñez, María Elena García Sánchez, Ma Enriqueta Cortés de los Ríos, Sagrario Salaberri Ramiro, Ma Soledad Cruz Martínez, Nobel Perdú Honeyman, and Blasina Cantizano Márquez, editors, *Applied Linguistics Now: Understanding Language and Mind / La Lingüística Aplicada Hoy: Comprendiendo el Lenguaje y la Mente*, pages 1433–1447. Universidad de Almería.

Pedro Oliveira and João Rocha. 2013. Semantic annotation tools survey. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 301–307.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Bhavesh Patel, Sanjay Soundarajan, Hervé Ménager, and Zicheng Hu. 2023. Making biomedical research software FAIR: Actionable step-by-step guidelines with a user-support tool. *Scientific Data*, 10(557).

Jiaxin Pei, Aparna Ananthasubramaniam, Xingyao Wang, Naitian Zhou, Jackson Sargent, Apostolos Dedeloudis, and David Jurgens. 2023. Potato: The portable text annotation tool. *arXiv*.

Tal Perry. 2021. Lighttag: Text annotation platform. *arXiv*.

Jan Wira Gotama Putra, Simone Teufel, Kana Matsumura, and Takenobu Tokunaga. 2020. TIARA: A tool for annotating discourse relations and sentence reordering. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6912–6920, Marseille, France. European Language Resources Association.

Georg Rehm. 2016. The language resource life cycle: Towards a generic model for creating, maintaining, using and distributing language resources. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2450–2454, Portorož, Slovenia. European Language Resources Association (ELRA).

Georg Rehm. 2020. Observations on annotations. In Julia Nantke and Frederik Schlupkothen, editors, *Annotations in Scholarly Edition and Research. Functions, Differentiation, Systematization*, pages 299–324. De Gruyter, Berlin, Boston.

Georg Rehm, editor. 2023. *European Language Grid: A language technology platform for multilingual europe*. Cognitive Technologies. Springer, Cham, Switzerland.

Burr Settles. 2009. Active learning literature survey. *University of Wisconsin-Madison Department of Computer Sciences*.

Hiroyuki Shindo, Yohei Munesada, and Yuji Matsumoto. 2018. PDFAnno: A web-based linguistic annotation tool for PDF documents. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Erik Smistad, Andreas Østvik, and Lasse Løvstakken. 2021. Annotation web - an open-source web-based annotation tool for ultrasound images. In *2021 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4.

Jurriaan H. Spaaks, Stefan Verhoeven, Erik Tjong Kim Sang, Faruk Diblen, Carlos Martinez-Ortiz, Edidiong Etuk, Mateusz Kuzak, Ben Werkhoven, Abel Soares Siqueira, Shyam Saladi, and Andrew Holding. 2022. howfairis.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Regina Stodden and Laura Kallmeyer. 2022. TS-ANNO: An annotation tool to build, annotate and evaluate text simplification corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 145–155, Dublin, Ireland. Association for Computational Linguistics.

Norman Süsstrunk, Andreas Fraefel, Albert Weichselbraun, and Adrian M. P. Brasoveanu. 2023. Orbis annotator: An open source toolkit for the efficient annotation and refinement of text. In *Proceedings of the 4th Conference on Language, Data and Knowledge*, pages 294–305, Vienna, Austria. NOVA CLUNL, Portugal.

Hadrien Titeux, Rachid Riad, Xuan-Nga Cao, Nicolas Hamilakis, Kris Madden, Alejandrina Cristia, Anne-Catherine Bachoud-Lévi, and Emmanuel Dupoux. 2020. Seshat: A tool for managing and verifying annotation campaigns of audio data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6976–6982, Marseille, France. European Language Resources Association.

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2022. Label Studio: Data labeling software. Open source software available from https://github.com/heartexlabs/label-studio.

Brian Vander Schee. 2009. Crowdsourcing: Why the power of the crowd is driving the future of business. *Journal of Consumer Marketing*, 26:305–306.

Josh Veitch-Michaelis. 2021. jveitch-michaelis/deeplabel: v0.16.1. Zenodo.

Daniel Vila-Suero and Francisco Aranda. 2023. Argilla - open-source framework for data-centric NLP.

Chih-Hsuan Wei, Hung-Yu kao, and Zhiyong Lu. 2013. Pubtator: A web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41:W518–W522.

Max Wiechmann, Seid Muhie Yimam, and Chris Biemann. 2021. ActiveAnno: General-purpose document-level annotation tool with active learning integration. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 99–105, Online. Association for Computational Linguistics.

Mark D. Wilkinson, Michel Dumontier, I. Jsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3(160018).

Mark D. Wilkinson, Michel Dumontier, Susanna-Assunta Sansone, Luiz Olavo Bonino da Silva Santos, Mario Prieto, Dominique Batista, Peter McQuilton, Tobias Kuhn, Philippe Rocca-Serra, Mercè Crosas, and Erik Schultes. 2019. Evaluating FAIR maturity through a scalable, automated, community-governed framework. *Scientific Data*, 6:2052–4463.

Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. 2006. ELAN: A professional framework for multimodality research. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2018. Yedda: A lightweight collaborative text span annotation tool. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

A FAIR4RS Principles Interpretations

F: Software, and its associated metadata, is easy for both humans and machines to find.	
F1. Software is assigned a globally unique and persistent identifier.	An annotation tool should have a globally unique and persistent identifier (PID), such as Digital Object Identifier (DOI), which assures longevity and consistently points to the software despite changes in its location, content or other attributes.
F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.	In addition to the annotation tool itself, distinct PIDs should be assigned to all its components (files, directories, commits, releases, and other attributes).
F1.2. Different versions of the software are assigned distinct identifiers.	Each release of an annotation tool should be assigned a distinct PID allowing users to track its development and refer to a specific version they utilised.
F2. Software is described with rich metadata.	Metadata should be semantically structured, i. e., being both human and machine-readable. It should contain a thorough description of an annotation tool allowing users to understand how to utilise and replicate it without looking into its source code. Metadata is considered to be rich when it goes beyond basic information. Following Bioschemas ComputationalTool, the minimum metadata of an annotation tool should include *name*, *URL*, and *description*. The metadata is considered to be rich if at least one additional property from any marginality level is provided.
F3. Metadata clearly and explicitly include the identifier of the software they describe.	If an annotation tool is assigned a PID, it should be referenced by it in the respective structured metadata. Thus, in case F1 is not fulfilled, F3 fails as well.
F4. Metadata are FAIR, searchable and indexable.	Metadata is FAIR when it is semantically structured. Therefore, if an annotation tool fails F2 due to the lack of structured metadata, it automatically fails F4. Any metadata exposed via web pages in a format understood by search engines or deposited in a repository/registry with search functionality is indexable.
A: Software, and its metadata, is retrievable via standardized protocols.	
A1. Software is retrievable by its identifier using a standardised communications protocol.	An annotation tool should be accessed through a commonly used communication protocol such as Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS) or File Transfer Protocol (FTP).
A1.1. The protocol is open, free, and universally implementable.	There should be no restrictions and fees to implement the communication protocol.
A1.2. The protocol allows for an authentication and authorisation procedure, where necessary.	The protocol should include mechanisms to verify the identity of users and to determine their access rights where necessary. Authentication and authorisation are supported by HTTP/HTTPS and FTP protocols, therefore if an annotation tool is retrievable via those, it automatically fulfills A1.2.
A2. Metadata are accessible, even when the software is no longer available.	As software tend to be deprecated over time, ideally structured metadata should be published separately with its own PID. In practice, it is often embedded into the source code of software. Therefore, this principle is satisfied when metadata is assigned a distinct PID and published either separately from a tool or along with it on an archive which ensures longevity.
I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.	

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.	Currently, there do not seem to be well-defined standards for annotation tools' input/output formats. The formats vary depending on the input modality, domain, and specific task at hand. Thus, input formats for text documents include *DOC*, *PDF*, *TXT*, *RTF*, *CSV*, *TSV*, *XML*, and *JSON*. Audio inputs are commonly available in *WAV*, *MP3*, *OGG*, *AIF*, and *FLAC*. Image inputs tend to be *PNG*, *JPEG*, and *JPG*, while video data typically comes as *MPEG*, *DIVX*, *AVI*, and *MOV*. Semantic annotation outputs are usually in *XML* (or *XMI*), *RDF*, *RDFa*, *RDFS*, *OWL*, *CSV*, *TXT*, *JSON*, *TEI*, *EAF*, *CMML*, *IRI*, *SMIL*, and *TFRecord* formats. The principle is considered to be fully fulfilled if an annotation tool supports at least one of the standard formats for both input and output.
I2. Software includes qualified references to other objects.	This principle calls for references to any objects other than software such as datasets, hardware, programming language, operating system or browser. Qualified references include identifiers (URLs, PIDs, etc.) and controlled vocabularies.
R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).	
R1. Software is described with a plurality of accurate and relevant attributes.	An annotation tool should be described in terms of metadata categories (F2), license (R1.1), and provenance (R1.2). The relevance of attributes is usually determined by repositories and/or communities that create and use a tool. Whenever feasible, multiple terms for the same, similar or overlapping concepts should be provided to allow reuse. However, to the best of our knowledge, there are no community-agreed standards for the metadata vocabulary of annotation tools. Therefore, R1 is considered to be fulfilled if a tool fully adheres to F2, R1.1 and R1.2.
R1.1. Software is given a clear and accessible license.	The annotation tool's license should be clearly stated in either structured or unstructured metadata.
R1.2. Software is associated with detailed provenance.	This principle calls for an explanation of the annotation tool's origins and development history. To this end, structured or unstructured metadata should provide answers to the following questions: *Why and how a tool came to be? Who contributed what, when, and where? How to cite a tool?*
R2. Software includes qualified references to other software.	As with any other software, annotation tools usually have dependencies. Thus, qualified references to other software (e. g., libraries, packages) should be provided in either structured or unstructured metadata.
R3. Software meets domain-relevant community standards.	Currently, there do not seem to be well-established community standards for annotation tools. The desired capabilities of software are influenced by the annotation project scope and goals. However, there are ten essential functionalities that ideally should be incorporated into an annotation tool for an easier user experience: 1. custom schemas support, 2. multilingual support, 3. web-based access, 4. support for teams and roles, 5. support for standard input/output file formats, 6. allowance for import/export of multiple file formats, 7. allowance for document-level annotation, 8. support for multi-label annotation, 9. allowance for automatic IAA score calculation, 10. data privacy support.

Table 3: Interpretations of the FAIR4RS principles (Chue Hong et al., 2022) tailored to the specific use-case of annotation tools.

B Metadata Sources

Annotation tool	Unstructured metadata	Structured metadata
INCEpTION (Klie et al., 2018)	README, Documentation	GitHub, ELG
brat	README, Documentation	–
Doccano	README	GitHub
BioQRator (Kwon et al., 2013)	README, Documentation	–
Catma (Gius et al., 2023)	README, Documentation	GitHub
Djangology (Apostolova et al., 2010)	SourceForge	–
ezTag (Kwon et al., 2018)	README, Documentation	–
FLAT	README, Documentation	GitHub
LightTag (Perry, 2021)	Documentation	–
MAT	Documentation	–
PDFAnno (Shindo et al., 2018)	README	GitHub
prodigy	–	–
TextAE	Documentation	GitHub
WAT-SL	README	GitHub
Hypothesis	GitHub, Homepage	GitHub
Haystack	README, Documentation	GitHub
PDF sentence annotator	README	–
PAWLS (Neumann et al., 2021)	README	GitHub
TeamTat (Islamaj et al., 2020)	README, Documentation	GitHub
TagEditor	README	–
TS-ANNO (Stodden and Kallmeyer, 2022)	README	–
MedTator (He et al., 2022)	README	–
DocTAG (Giachelle et al., 2022)	README	–
PubTator	–	–
Ellogon (Ntogramatzis et al., 2022)	ELG, Documentation	–
Markup	README	GitHub
Label Studio	README, Documentation	GitHub
MedTag (Giachelle et al., 2021)	README	–
BAT (Meléndez-Catalán et al., 2017)	README	–
Seshat (Titeux et al., 2020)	README, Documentation	–
VIA (Dutta and Zisserman, 2019)	README, Homepage	–
Potato (Pei et al., 2023)	README, Documentation	GitHub
Annotation Web (Smistad et al., 2021)	README	–
audino (Grover et al., 2020)	README	–
MATILDA	README	–
ELAN (Wittenburg et al., 2006)	Homepage, Documentation	–
Praat (Boersma and Weenink, 2023)	README, Documentation	–
Pundit	README	GitHub
UAM CorpusTool (O'Donnell, 2008)	Homepage	–
TIARA (Putra et al., 2020)	README, Documentation	–
COCO Annotator (Brooks, 2019)	README	–
Gate Teamware (Karmakharm et al., 2023)	README, Homepage, Documentation	GitHub
ActiveAnno (Wiechmann et al., 2021)	README	GitHub
YEDDA (Yang et al., 2018)	README	–
Textinator (Kalpakchi and Boye, 2022)	README, Documentation	GitHub
Argilla (Vila-Suero and Aranda, 2023)	README, Documentation	GitHub
Orbis Annotator (Süsstrunk et al., 2023)	README	GitHub
CVAT (Corporation, 2023)	README	GitHub
DataGym.ai	README, Documentation	GitHub
DeepLabel (Veitch-Michaelis, 2021)	README	–

Table 4: Sources for structured and unstructured metadata for each annotation tool.

C Input and Output Formats

The input and output formats of annotation tools vary depending on the data modality at hand. However, output annotations should ideally be semantically structured using formats such as XML or RDF. We surveyed the literature in order to identify the most commonly used input/output formats per modality and to refer to those while evaluating the principle I1. The results are as follows: Input formats for text documents include *DOC*, *PDF*, *TXT*, *RTF*, *CSV*, *TSV*, *XML*, and *JSON* (Dasiopoulou et al., 2011; Oliveira and Rocha, 2013; Ide and Pustejovsky, 2017). Audio input formats are commonly available in terms of *WAV*, *MP3*, *OGG*, *AIF*, and *FLAC* formats (Dasiopoulou et al., 2011). Image inputs tend to be *PNG*, *JPEG*, and *JPG*, while video data is typically supported in the form of *MPEG*, *DIVX*, *AVI*, and *MOV* (Dasiopoulou et al., 2011). Semantic annotation outputs are usually in *XML* (or *XMI*), *RDF*, *RDFa*, *RDFS*, *OWL*, *CSV*, *TXT*, *JSON*, *TEI*, *EAF*, *CMML*, *IRI*, *SMIL*, and *TFRecord* formats (Dasiopoulou et al., 2011; Oliveira and Rocha, 2013; Ide and Pustejovsky, 2017; Aljabri et al., 2022). Table 5 summarises the input and output formats available in the 50 surveyed tools according to the information found in their metadata.

Annotation tool	**Input formats**	**Output formats**
INCEpTION	BioC (experimental), CoNLL 2000, CoNLL 2002, CoNLL 2003, CoNLL 2006, CoNLL 2009, CoNLL 2012, CoreNLP CoNLL-like format, CoNLL-U, HTML (legacy), HTML, IMS CWB (aka VRT), NLP Interchange Format (NIF), PDF Format, PDF Format (legacy), Perseus Ancient Greek and Latin Dependency Treebank 2.1 XML, WebLicht TCF, TEI P5 XML, Plain Text, Plain Text (one sentence per line), Plain Text (pretokenized), UIMA Binary CAS, UIMA Inline XML, UIMA CAS JSON (experimental), UIMA CAS JSON (legacy), UIMA CAS XMI, WebAnno TSV 1 (legacy), WebAnno TSV 2 (legacy), WebAnno TSV 3.x, XML (generic)	BioC (experimental), CoNLL 2000, CoNLL 2002, CoNLL 2003, CoNLL 2006, CoNLL 2009, CoNLL 2012, CoreNLP CoNLL-like format, CoNLL-U, HTML (legacy), HTML, IMS CWB (aka VRT), NIF, PDF Format, PDF Format (legacy), Perseus Ancient Greek and Latin Dependency Treebank 2.1 XML, WebLicht TCF, TEI P5 XML, Plain Text, Plain Text (one sentence per line), Plain Text (pretokenized), UIMA Binary CAS, UIMA Inline XML, UIMA CAS JSON (experimental), UIMA CAS JSON (legacy), UIMA CAS XMI, WebAnno TSV 1 (legacy), WebAnno TSV 2 (legacy), WebAnno TSV 3.x, XML (generic)
brat	Plain Text	.ann
Doccano	JSON, Plain Text, CoNLL	XML
BioQRator	PubMed, BioC	CSV, BioC
Catma	HTML, Plain Text	CSV, Plain Text, TEI XML
Djangology	Plain Text	Plain Text
ezTag	BioC	BioC
FLAT	FoLiA	FoLiA
LightTag	Plain Text, JSON, WebAnno TSV, TSV, CSV	–
MAT	XML	XML, JSON
PDFAnno	PDF	TOML
prodigy	JSONL, JSON, CSV, Plain Text, JPG, JPEG, PNG, GIF, SVG, MP3, M4A, WAV, MPEG, MPG, MP4	JSON
TextAE	JSON	JSON
WAT-SL	Plain Text	Plain Text
Hypothesis	HTML, PDF	–
Haystack	Plain Text	SQuAD JSON, XLSX, CSV
PDF sentence annotator	PDF	RDF
PAWLS	PDF	JSON
TeamTat	BioC, PDF, Plain Text	BioC
TagEditor	Plain Text, JSON, .spacy	JSON, .spacy
TS-ANNO	HTML, Plain Text	CSV
MedTator	Plain Text, XML	WebAnno TSV, TSV, JSONL
DocTAG	CSV, JSON, Plain Text	JSON, Plain Text
PubTator	Plain Text	BioC, JSON, POST, PubTator
Ellogon	Plain Text, TEI XML	JSON, CSV, XLSX, image formats(for the case of the charts)
Markup	Plain Text	ZIP
Label Studio	HTML, HTM, XML, BMP, GIF, JPG, PNG, SVG, WebP, JSON, Plain Text, TSV, CSV, FLAC, M4A, MP3, OGG, WAV, MP4, WebM, AVI	ASR MANIFEST (JSON manifest), NumPy, PNG, COCO, CoNLL 2023, CSV, JSON, JSON MIN, Pascal VOC XML, spacy, TSV, YOLO
MedTag	CSV	XML, JSON, CSV, BioC
BAT	WAV	–
Seshat	WAV, FLAC, MP3	ZIP
VIA	JPEG, PNG, URL of a webpage	CSV, JSON, COCO
Potato	JSON, TSV, CSV	JSON, TSV, CSV, JSONL
Annotation Web	PNG, MHD, RAW	–
audino	WAV, MP3, OGG	–
MATILDA	JSON	–

ELAN	MPG, MP4, WAV, etc.	HTML, Plain Text, XML, JSON, CSV, FLEx, CHAT, SMIL3-compliant clips, EAF, etc.
Praat	able to read most standard types of sound files, e.g. WAV files	UIMA Binary CAS/Binary
Pundit	URL of a webpage, PDF, Plain Text	Notebooks
UAM CorpusTool	Plain Text	HTML
TIARA	Plain Text	HTML, TSV
COCO Annotator	–	JSON, COCO
Gate Teamware	CSV, JSON	CSV, JSON
ActiveAnno	JSON	JSON
YEDDA	Plain Text	.ann
Textinator	Plain Text, JSON	JSON
Argilla	JSON	JSON, CSV, Parquet, XLSX, PKL (Python pickle file)
Orbis Annotator	NIF turtle, CareerCoach JSON	CareerCoach 2022, NIF turtle
CVAT	CVAT for images, CVAT for videos, Datumaro, PASCAL VOC, Segmentation masks from PASCAL VOC, YOLO, MS COCO Object Detection, MS COCO Keypoints Detection, TFrecord, MOT, MOTS PNG, LabelMe 3.0, ImageNet, CamVid, WIDER Face, VGGFace2, Market-1501, ICDAR13/15, Open Images V6, Cityscapes, KITTI, Kitti Raw Format, LFW, Supervisely Point Cloud Format	CVAT for images, CVAT for videos, Datumaro, PASCAL VOC, Segmentation masks from PASCAL VOC, YOLO, MS COCO Object Detection, MS COCO Keypoints Detection, TFrecord, MOT, MOTS PNG, LabelMe 3.0, ImageNet, CamVid, WIDER Face, VGGFace2, Market-1501, ICDAR13/15, Open Images V6, Cityscapes, KITTI, Kitti Raw Format, LFW, Supervisely Point Cloud Format
DataGym.ai	JPEG, PNG	JSON
DeepLabel	Darknet (provide image list and names), COCO (provide an annotation JSON file and image folder), MOT, TFRecord (parsing works, but full import is not possible yet) Pascal VOC	KITTI, Darknet for YOLO Pascal VOC, COCO (experimental), Google Cloud Platform, TFRecord, Video (experimental, command line only)

Table 5: Available input and output formats for each of the 50 surveyed tools.