LDK 2023

Orbis Annotator: An Open Source Toolkit for the Efficient Annotation and Refinement of Text Corpora

Norman Süsstrunk and **Andreas Fraefel** and **Albert Weichselbraun**
Fachhochschule Graubünden
Chur, Switzerland
{first.lastname}@fhgr.ch

Adrian M.P. Braşoveanu
Modul University and
Modul Technology GmbH
Vienna, Austria
adrian.brasoveanu@modul.ac.at

Abstract

Annotated language data plays an important role in training, fine-tuning and evaluating natural language processing components. Nevertheless, manually annotating language data is still a cumbersome task.

This paper presents the Orbis Annotator framework, a user-friendly, easy to install, web-based software that supports users in efficiently annotating language data. Orbis Annotator supports standard and collaborative workflows, reuse of language resources through corpus versioning, and provides built-in tools for assessing corpus quality. In addition, it offers an API which enables the use of different clients (e.g., web-based, command line, etc.) and the use of third-party tools that accelerate the annotation process by pre-annotating corpora.

The paper concludes with an evaluation that compares its features to other open-source annotation frameworks and the description of two use cases that outline its use in more sophisticated settings.

1 Introduction

With the emergence of deep neural networks, unsupervised pre-training on massive datasets has gained in importance. Although pre-trained language models require a considerably lower number of training examples when compared to the early deep learning models, these models still benefit tremendously from further fine-tuning on labelled data. Gold standard corpora play a pivotal role in adapting models to concrete tasks, and in evaluating model performance. This is particularly true when considering the rise of machine learning approaches in research and industry.

Creating annotated gold standard corpora is still a labor-intensive task, although many toolkits such as Annotation Study[1], BRAT[2] (Brat Rapid Annotation Tool; Stenetorp et al. (2012)), Prodigy[3], Docanno[4], Gate Teamware[5], and INCEpTION[6] that support the annotation process exist.

But even with specialized tools, annotators lose valuable time with marking annotation spans and assigning them to the corresponding annotations. Drawing upon automatically generated silver standard annotations, has the potential to significantly improve efficiency. More sophisticated annotation tools support pre-annotating text, and in some cases even online learning, which ensures that human feedback (e.g., corrections of machine-generated annotation annotations) is leveraged for improving the automated pre-annotation process.

Unfortunately, many solutions are either difficult to install, lack vital functionality such as support for pre-annotated corpora, collaborative work flows and computation of corpus statistics (e.g., the inter-rater agreement), or are only available under commercial licenses.

Orbis Annotator addresses these shortcomings and builds upon prior work by providing a solution which

- is easy to install and use
- integrates tightly with machine learning approaches, that provide silver-standard annotations
- allows refining and improving existing corpora
- supports collaborative annotation processes
- increases annotator efficiency through (optional) pre-annotations, keyboard shortcuts and mouse actions (i.e., it supports both keyboard-centric and mouse-centric annotators)

[1] https://annotation-study.org
[2] https://brat.nlplab.org
[3] https://prodi.gy
[4] https://github.com/doccano/doccano
[5] https://gate.ac.uk/teamware
[6] https://github.com/inception-project/inception

In addition, Orbis Annotator will be coupled with the next version of the Orbis Visual Benchmarking Platform (github.com/orbis-eval) which will bundle the creation of gold standards with a suite of explainable benchmarking tools that supports evaluating human and machine annotators on the created datasets.

The presented research, therefore, provides the following contributions:

1. the introduction of Orbis Annotator, a text annotation framework that is easy to use and considerably improves the efficiency of creating gold standards;
2. an overview and comparison of existing open-source annotation tools,
3. the presentation of two uses cases (machine-based corpus pre-annotation of custom entity types, and corpus migration to a new knowledge graph) that demonstrate how Orbis Annotator has been successfully deployed in real-world settings.

The rest of this paper is organized as follows: Section 2 provides an overview of related work. Afterwards, Section 3 introduces *Orbis Annotator*. Section 4 discusses the strengths and weaknesses of Orbis Annotator based on two use cases, compares it to related frameworks, and outlines the gains in productivity achieved by drawing upon the system. The paper closes with the conclusions and an outlook presented in Section 5.

2 Related Work

Deep Learning requires large text collections for unsupervised training. Depending on the chosen learning tasks, unsupervised training might be complemented with fine-tuning on annotated data to help in improving systems' performance. This has led to an increase in the number of annotation tools developed in the past five years, as can be seen by examining the papers accepted at leading natural language processing and machine learning conferences such as ACL, EMNLP, CoNLL, COLING, LREC, etc. Therefore, the following discussion on related research had to be narrowed to a limited number of papers. The criteria used in this paper were: (i) historical significance (e.g., tools supported by larger number of users who are still popular within the academia and industry); (ii) availability (e.g., published in open-source repositories or free to use); (iii) ease of use (i.e., tools can be installed and operated without specialized training and in-depth knowledge of their implementation); and (iv) support for current NLP trends (e.g., if the tools support machine-aided annotation generation mechanisms like active learning).

Readers interested in a comprehensive survey on annotation tools, may refer to a recent overview paper by Neves and Seva (2021) that surveyed 78 tools and provides a detailed comparison of 15 of them. Although their survey is mostly focused on the domain of bioinformatics, it also includes well-known general tools such as BRAT, ezTag and Prodigy. Nevertheless, none of the tools included was able to cover all the needs of the survey's authors.

Perhaps the oldest, and best known software in the space is GATE (Cunningham, 2002) which started as a single annotator tool in the late 1990s and morphed into a collaborative tool called GATE-Teamware (Bontcheva et al., 2013) a decade ago. GATE was created for multiple span annotations and turned out to be ideal for tasks like tokenization, named entity recognition (NER), sentiment analysis, dependency parsing (DP), part-of-speech tagging (POS), and coreference resolution (CR).

UIMA (Unstructured Information Management Architecture; Ferrucci and Lally (2004)) is a generalized annotation architecture that supports interoperability. Various annotation toolkits such as DKPro WSD (Miller et al., 2013) and TextAnnotator (Abrami et al., 2020) are built around UIMA's philosophy.

BRAT (Stenetorp et al., 2012) gained some traction a decade ago, but was eventually abandoned. BRAT can be used for similar tasks as GATE. WebAnno (Yimam et al., 2013) builds directly on top of the BRAT functionality. More recent tools such as APletny (Nghiem and Ananiadou, 2018), ActiveAnno (Wiechmann et al., 2021) and Paladin (Nghiem et al., 2021) adapt WebAnno's functionality to new active learning use cases.

The Stanford CoreNLP (Manning et al., 2014) toolkit supports the creation of custom annotators, and provides a regular expression-based mechanism (RegexNER) for pre-annotating documents. CoreNLP was the first annotator widely used for Deep Learning tasks, and its description in Manning et al. (2014) provides good definitions for the supported annotation tasks.

In addition to domain-specific tools (e.g., for the medical and finance domain), many frameworks that have been tailored towards specific text annotation tasks exist. Yedda (Yang et al., 2018), for instance, was built for annotating specialized entity types (e.g., events). TAG (Forbes et al., 2018) is optimized towards showcasing complex relations between sentences and documents. ALIGNMEET (Polák et al., 2022) and EZCAT (Guibon et al., 2022) focus on annotating meetings and conversations and support a wide array of languages, symbols, and emojis. Ellogon (Ntogramatzis et al., 2022) annotates moral values and arguments. Textinator (Kalpakchi and Boye, 2022) was created for internationalization and language evolution use cases. Semantic storytelling (Raring et al., 2022) is another use case that led to the development of a specialized tool.

AWOCATo (Daudert, 2020) is a recent tool that supports various annotation formats. Although not used for creating annotations, Spicy Salmon (Fäth and Chiarcos, 2022) deserves mentioning, since it provides an interface for converting between 50 different annotation formats. An early attempt towards interoperable annotations was NIF (Hellmann et al., 2012), an RDF-based language for producing customized annotation, although it is primarily used within the European data spaces.

Inception[7](Klie et al., 2018) builds upon UIMA's interoperability concepts and WebAnno's annotation functionalities. Inception offers several new concepts, like recommender algorithms that help improve annotation efficiency, and advanced customization capabilities.

Some open-source annotation tools that stand out include Argilla[8] and Docanno[9]. Since they are produced collaboratively under open licenses, these tools have a wider reach than the academic ones. Argilla supports active learning through its HuggingFace integration, provides a simple API, and has recently gained a significant following. Docanno offers collaborative editing, REST APIs and emoji support. Another famous but proprietary tool, Prodigy[10], was introduced by the Explosion team that created Spacy. Also powered by active learning, Prodigy offers classic text annotation features, supports A/B testing, and zero-shot prompts.

While not necessarily direct competitors to Orbis or other annotation solutions, instrumentation and explainability tools such as MLFlow [11], Weights and Biases[12] and neptune.ai, also deserve attention since their APIs allow for quick and easy instrumentation of AI components that train upon annotated corpora. An overview of these tools can be found in Braşoveanu and Andonie (2022).

3 Method

Several years ago, we started developing a benchmarking ecosystem after an early study about named entity linking evaluations (Brasoveanu et al., 2018) showcased a significant number of errors in existing gold standards and knowledge graphs. The initial version of Orbis (Odoni et al., 2018) was the first step in this direction. The first version only focused on named entity linking (NEL) evaluations, but later versions included support for content extraction evaluations (Weichselbraun et al., 2020), NER and basic slot filling evaluations. In time, it became clear that focusing only on the visual evaluation issue was not enough, and that there was a need for integrated platforms that support both the annotation and evaluation workflows. The Orbis Annotator, the tool presented in this paper, is focused on annotation workflows. Since this tool represents both a reimplementation and a significant expansion upon the previous generation, it was named Orbis 2. The design of the current version is modular (e.g., backend, frontend, or corpus exporter components are already included).

Major barriers towards deploying specialized software for annotating complex corpora are the software's availability (i.e., whether it is free to use or requires licenses), skill and effort required for setting up the software, and time necessary for using it efficiently. Many state-of-the-art solutions are either limited in terms of functionality, freedom of use, or are really difficult to setup and operate. Orbis Annotator aims at addressing these shortcomings by bundling all necessary components into a docker container, and providing an efficient, intuitive Web-based workflow that covers its basic functionality and does not require any prior training. In addition, Orbis Annotator supports more complex workflows through its data model (Section 3.1) and backend API (Section 3.2). The software has been released under the Apache 2.0 license and is available on Github[13] for download.

[7]https://inception-project.github.io/publications/
[8]https://github.com/argilla-io/argilla
[9]https://github.com/doccano/doccano
[10]https://prodi.gy/
[11]https://mlflow.org/
[12]https://wandb.ai/site
[13]https://github.com/orbis-eval/orbis2-frontend

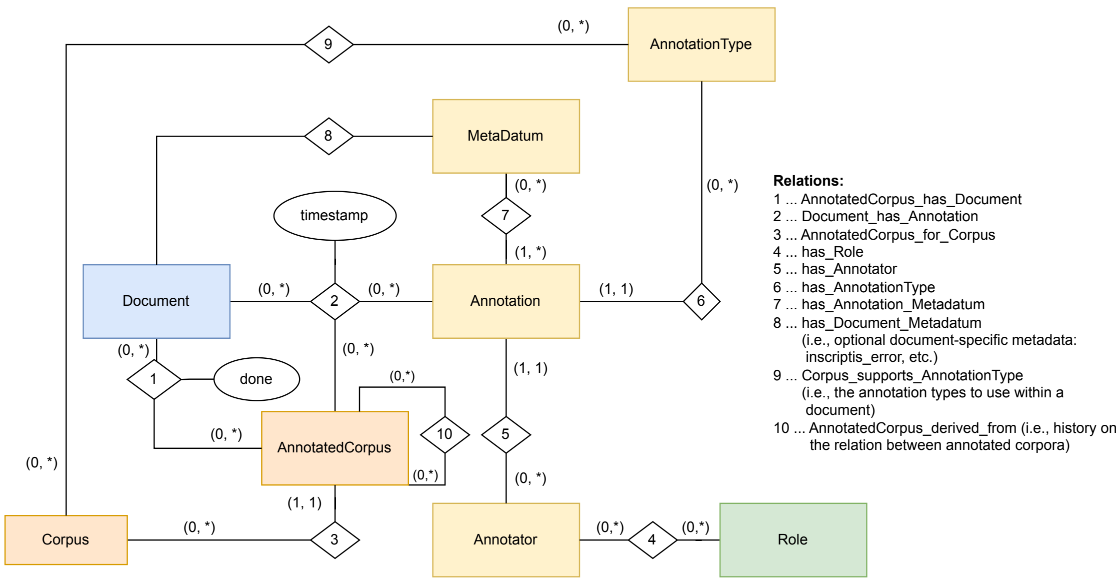

Figure 1: Entity Relationship model of the Orbis database (the attributes of the entities have been excluded, due to limited space)

3.1 Orbis data model

Orbis stores corpora, documents, annotations, and metadata (e.g., annotators, corpus versions, etc.) in a relational PostgreSQL[14] database. Its data model supports corpus and annotation versioning, atomic real-time updates and the export and import to popular formats such as JSON, Excel and NIF.

Use case studies and analysis of existing annotation and benchmarking suites yielded the following requirements for the Orbis data model:

1. *Interoperable:* Although Orbis does not aim at introducing another annotation format, its data model is required to support importing and exporting existing formats without information loss.

2. *Reusable*: Orbis promotes reuse of existing corpora by refining and improving them. This requirement comprises use cases such as using human annotators to promote automatically annotated silver standards to gold standards, updating corpora to newer versions of the knowledge base (e.g, DBpedia 2015-10 to a more recent version), and improving upon existing gold standard annotations.

3. *Multi-user capable*: Orbis supports groups of annotators that collaboratively add, correct and improve annotations. The data model records individual contributions, and supports multiple task designs (e.g., annotators working independently, versus collaborative settings).

4. *Workflow agnostic*: The data model shall enable multiple workflows with different levels of complexity (e.g., manual annotation by a single annotator, by multiple annotators; machine learning for pre-annotating corpora with silver standard annotations; hybrid workflows that combine machine and human annotators).

5. *Process metrics oriented*: The data model supports computing process metrics on individual annotators (e.g., throughput in terms of documents and number of annotations), and shared metrics (e.g., different kinds of inter-rater agreement).

Figure 1 provides the Entity Relationship model of the Orbis database.

Central element of the model is an *AnnotatedCorpus* which represents a certain version of a *Corpus* with all its documents, annotations and metadata. Importing a corpus creates a *Corpus* entity and the corresponding *AnnotatedCorpus*, which might either be empty (if an unannotated corpus has been imported) or contain initial annotations (e.g., from a gold standard, automated annotators, etc.) alongside the documents. Each *AnnotatedCorpus* consists of *Documents* and the corresponding *Annotations*. Orbis also records the *AnnotationType*, the *Annotator* and optional *MetaData* for all

[14] https://www.postgresql.org

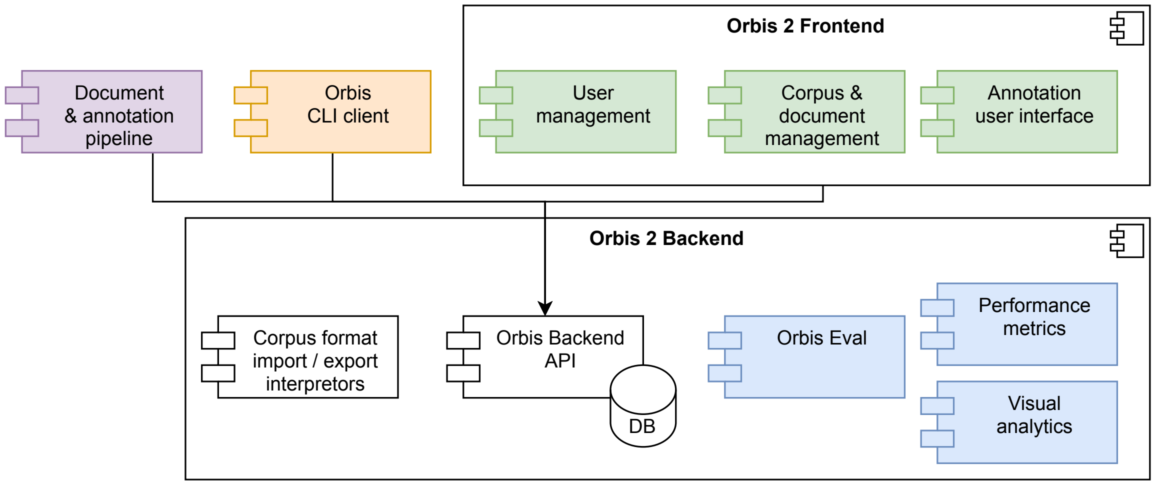

Figure 2: Overview of Orbis 2 architecture which outlines important frontend and backend components. The Orbis API also allows interaction with third-party pipelines and the Orbis Command Line (CLI) client.

annotations. In addition, Orbis implements user management and access control via the relations between *Annotators* and their respective *Roles*.

The relation between *Corpus* and *AnnotationType* allows specifying the set of annotation types to use within a corpus, and the *derivedFrom* relation enables tracking the relationship between different corpus versions. The chosen data model also allows tracking changes between *AnnotatedCorpus* entities (e.g., gold standard annotations, annotations provided by different persons, machine-generated annotations, etc.) which represent different corpus versions. These versions may be derived from

- gold standard labels which have been provided with the corpus;
- automated approaches such as named entity linking, named entity recognition and sentiment analysis which provide silver standard labels for evaluations or to accelerate manual annotation processes;
- manual annotations provided by annotators. Depending on the use case requirements, annotators might work on the same or different *AnnotatedCorpora* (i.e, produce common or separate corpus versions).

Orbis also supports computing standard metrics such as precision, recall, F1-measure and inter-rater agreement between these versions (Section 3.5).

3.2 Orbis backend

Figure 2 outlines how Orbis exposes its data model through a publicly available backend API. The Orbis backend API currently supports (i) the Orbis Annotator frontend used for annotating and refining corpora, (ii) the Orbis command line interface (CLI) client which focuses on performing evaluations and computing metrics, and (iii) integrating custom document and annotation pipelines which can add new documents to existing corpora, and manipulate corpus annotations (e.g., to provide silver standard annotations). As outlined in Section 4.2, the machine aided pre-annotations may be used to further enhance the efficiency of human annotators.

The backend also contains interpreters for corpus formats such as NIF, JSON and Excel which allow native consumption and production of these formats through the Orbis API. These interpreters are essential for compatibility with publicly available corpora, other annotation frontends, and existing software libraries such as SpaCy.

Future versions of Orbis Annotator will tightly integrate with the Orbis Explainable Benchmarking framework which will enable performing evaluations, and drill-down analyses on top of the created corpora.

3.3 Orbis Annotator frontend

The following design goals led to the development of the Orbis Annotator frontend: (i) the user interface should be intuitive and responsive, (ii) changes (i.e., added, modified and deleted annotations) should be automatically serialized to prevent data-loss, (iii) the interface should contain usability optimizations that are tailored towards annotator efficiency and support both mouse- and keyboard-centred workflows.

Figure 3: Visualization of the rendered tree structure in Orbis Annotator. The borders were added to illustrate the underlying tree-structure, and are invisible in the Orbis Annotator interface. The border color is used to indicate whether elements are annotated (yellow) or unannotated (grey).

3.3.1 Responsiveness and real-time updates

Converting the list of annotations into a tree using the nested set algorithms yields a tree structure from a list of annotations with start and end indices. The obtained tree structure offers several advantages:

1. It provides a more efficient way to query, retrieve and modify annotations, especially when dealing with large numbers of annotations;
2. the tree structure simplifies the rendering process by providing a clear hierarchy of the annotations;
3. it also allows for easier management of annotations, including sorting, filtering and adding or removing annotations in the text.

Figure 3 visualizes how the annotation tree is rendered into an HTML document. Boxes with a yellow border indicate the annotations rendered from the tree structure. Grey borders outline text blocks between annotations and line breaks.

Figure 4 illustrates the rendering of the document shown in Figure 3 within the Orbis Annotator user interface. Edits by annotators trigger calls to the Orbis API which ensures that changes are serialized in real-time.

3.4 Usability optimizations

Orbis supports both mouse- and keyboard-centred workflows. The mouse-centred workflow allows users to perform annotation tasks without any use of the keyboard. The keyboard-centred workflow is currently in beta.

3.5 Corpus metrics

The current version of Orbis Annotator implements the following corpus quality metrics which may be computed through the Orbis evaluation command line client.

1. *Average F1 measure*: The average F1 measure computes the F1 metric between n annotators, to assess the amount of agreement between them.

$$\bar{F_1} = \frac{1}{n \cdot (n-1)} \sum_{i}^{n} \sum_{j \neq i}^{n} F_1(i,j) \quad (1)$$

2. *Modified Kappa:* The modified Kappa metric is based on the Fleiss' Kappa but does not correct for random agreement since it is usually negligible for corpus annotation tasks. It is computed by obtaining the average probability (P_i) of agreement among raters for each annotation i. Equation 3 shows the computation of P_i for annotation i based on the number of total raters n_i for that particular annotation and the number of raters considering it to be valid ($n_{i,vd}$) and invalid ($n_{i,\neg vd}$).

$$P_i = \frac{\sum_{j \in \{vd, \neg vd\}} n_{ij}(n_{ij}-1)}{n_i(n_i-1)} \quad (2)$$

$$\kappa^* = \frac{1}{n} \sum_{i=1}^{n} P_i \quad (3)$$

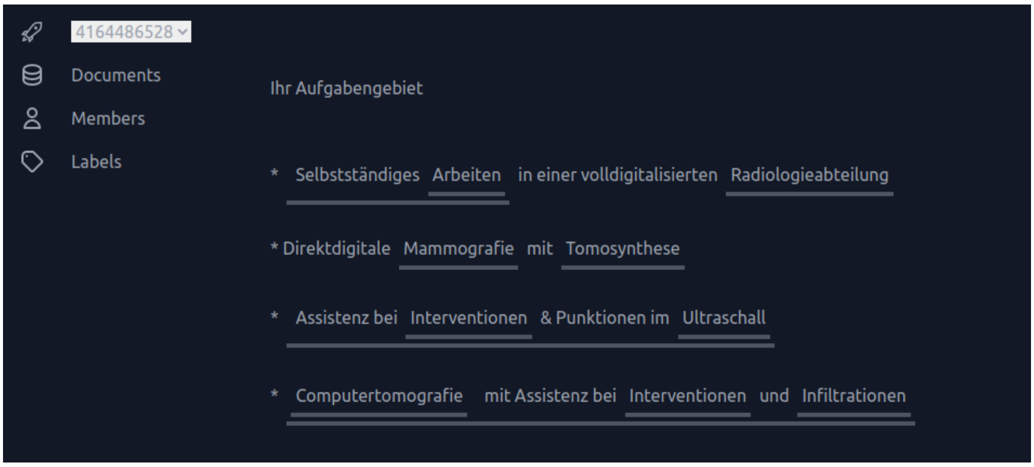

Figure 4: Orbis Annotator user interface for a given document with overlapping annotations.

Future version of Orbis Annotator will fully integrate with the Orbis Evaluation framework, which will allow conducting comprehensive evaluations and visual analytics on all annotated datasets.

3.6 Extensibility

Orbis Annotator includes the basic functionality required for uploading, annotating, evaluating and downloading corpora. In addition, it supports more complex use cases, such as automatically pre-annotating corpora through its API.

Future Orbis Annotator versions, will provide a plugin framework which allows extending both its user interface and API. Bundling these plugins in docker images that also include dependencies will provide additional functionality which is accessible to any user capable of starting a docker image and working with a web browser. Pre-configured docker images with automatic annotators such as SpaCy [15], DBpedia Spotlight [16] or Recognyze (Weichselbraun et al., 2019b), for instance, can enrich Orbis Annotator with active learning support.

4 Evaluation

The following section performs a qualitative evaluation which compares Orbis Annotator to other open-source annotation tools (Section 4.1), and presents its application to two sophisticated real-world use cases (Section 4.2).

4.1 Comparison of Annotation Frameworks

The following comparison of annotation frameworks focuses on open-source software, that is still under active development.

We excluded proprietary tools, since they are limited in transparency, customizability, and interoperability with other software. Moreover, commercial tools often require payment of high licensing fees, which are a significant barrier for researchers with limited resources or those who require extensive customization or experimentation with the software. Commercial solutions are, therefore, not considered in the comparison.

The comparison also excludes software which might not be maintained any more. As criteria for assessing a software's maintenance status, we investigated its code repository and excluded tools that haven't received any fixes or updates within the last two years, as we wanted to focus on systems that are still actively developed. This constraint led to the exclusions of Callisto[17], CoSACT[18] and Gate Teamware[19].

We assess popular annotation tools based on the following criteria:

Custom Types: The ability to define custom annotation types in an annotation tool is essential for adapting annotation tools to new domains and use cases. Custom annotation types enable domain-specific annotations that capture the unique features and nuances of the data being annotated, improving the accuracy of downstream analyses. Furthermore, the ability to define custom annotation types enables collaboration and reproducibility by allowing researchers to use a standardized annotation schema. Overall, custom annotation types are crucial for achieving high-quality annotations and advancing scientific research.

[15] https://spacy.io/
[16] https://www.dbpedia-spotlight.org/
[17] https://mitre.github.io/callisto/
[18] https://github.com/TDaudert/CoSACT
[19] https://gate.ac.uk/teamware/

Table 1: Comparison of popular open-source annotation tools.

	Nested Annotations	Custom types	Machine-aided annotations	Metrics	Multi User	Easy setup	License
Orbis Annotator	⊕	⊕	⊕	⊕	⊕	⊕ (Docker)	Apache 2.0
Argilla	⊕	⊕	⊕	⊕	⊕	⊕ (Docker)	Apache 2.0
Doccano	-	⊕	⊕	⊕	⊕	⊕ (Docker)	MIT
TagEditor	-	⊕	-	-	-	- (EXE-file)	MIT
Inception	-	⊕	⊕	⊕	⊕	- (Runnable Jar)	Apache 2.0
Annotation Studio	⊕	⊕	-	-	⊕	- (multi-step setup)	GPL 2.0
BRAT	-	⊕	-	-	⊕	- (Installer-Script)	MIT

Machine-Aided Annotations: Due to the sheer volume of data that needs to be annotated, machine-aided automatic annotations have become increasingly important recently. Machine learning algorithms can assist human annotators by automatically suggesting annotations for a given input based on pre-existing labelled data. This can significantly reduce the time and cost associated with manual annotation.

Multi-User: Multi-user-support in an annotation-tool is crucial for collaborative annotation projects in scientific research. With the ability to support multiple users, teams can work together to complete annotations more efficiently and effectively. This feature enables team members to view and edit annotations made by others, fostering collaboration and enhancing the accuracy and completeness of the annotations. Additionally, multi-user-support can provide a platform for experts to review and validate annotations made by less experienced annotators, improving the quality of the annotations.

Nested Annotations: Often, named entities are not linear but rather nested (i.e., a single entity can contain other entities). For instance, the mention "Barack Obama" refers to a person, but is nested within the mention "Barack Obama's administration" which points to an organization. Being able to annotate such nested annotations is crucial for accurately capturing the complexity of named entities in text. Annotating nested entities can improve the quality of the corpus and the performance of named entity recognition systems trained on it, as they can learn to recognize more complex named entity structures.

Easy Setup: Ease of setup is an essential factor to consider. With the increasing complexity of NLP and machine learning models, researchers require efficient and user-friendly tools to streamline their work. Single-platform executables were generally excluded, as we wanted to focus on tools for a larger audience. Software that is difficult to set up and configure can pose significant barriers to adoption, hindering the progress of research. In contrast, tools that are easy to set up and use can save researchers valuable time and effort, allowing them to focus on their research questions and hypotheses. Additionally, software with straightforward setup processes can encourage collaboration and community-building, as they make it easier for researchers to share their work and replicate experiments.

License Type: Open-source tools have revolutionized the fields of natural language processing (NLP) and machine learning research by providing researchers with accessible and customizable software. The use of open-source software has contributed towards increasing the reproducibility and transparency of research, since code and data are freely available for inspection and modification. In addition, open-source tools facilitate collaboration and community-building, by enabling researchers to share resources, expertise, and best practices.

Table 1 summarizes the evaluation results. The ⊕ symbol indicates that a criterion has been fully fulfilled, a minus refers to missing or only partially met criteria.

Support for nested annotations, machine-added annotations and corpus metrics are the areas that are most often neglected in the compared tools. Both Argilla and Orbis excel in these areas. In addition, future versions of Orbis Annotator will offer a tight integration with the Orbis Visual Benchmarking framework which will allow performing comprehensive evaluations of the created datasets and enable features designed toward improving the explainability of benchmarking results, such as drill-down analyses and aids for visualizing and interpreting evaluation results.

4.2 Use cases

This section discusses the use of the Orbis Annotator in two sophisticated real-world use cases which have significantly benefited from its development.

4.2.1 Machine-aided corpus annotation with non-standard, complex entity types

The first use cases showcased how machine-aided pre-annotations of complex entity types can lead to significant productivity gains of human annotators.

This use case design has been triggered by an applied research project in which the industry partner used a custom composite entity type to represent employee skills. This custom type combines a noun which specifies the skill's topic (e.g., Python) with a verb that indicates the skill's scope (e.g., programming). The composite skill type, therefore, enables a much more fine-grained distinction of a skill's required depth and direction (e.g., knowledge versus application or use). The skill scope may range from a shallow understanding ("knowing Python"), to different levels of practical experience ("programming Python", "debugging Python"), and the expertise required to actually teach a skill ("teaching Python").

Initially, human annotators identified these skills manually in real-time job posting feeds. They then copied sentences mentioning skills into a Google spreadsheet and provided a list of topic+scope tuples for these sentences.

The low productivity of the described process triggered the development of Orbis Annotator and migration to the machine-aided processes outlined in Figure 5. A machine learning pipeline splits job announcements into sentences, and then identifies sentences that are likely to contain composite skills. Afterward, an entity linking component provides a silver standard of annotated skill topics and skill scopes, which is then fed into the Orbis Annotator. Domain experts validate, extend and correct the provided silver standard annotations, creating a corpus of gold standard annotations, and the corresponding composite skills required for the industry partner's skill database. The annotation pipeline also queries the Orbis API for feedback on corrected annotations that is then used for enhancing the pipeline's machine learning components. The new process has considerably improved the productivity of the human annotators and helped in identifying over 80,000 different composite skills.

4.2.2 Knowledge Graphs migration

Knowledge graphs (KG) such as DBpedia and Wikidata have considerably grown recently (Hogan et al., 2021). Consequently, named entities that haven't been available in earlier KG versions (i.e., so called *nil entities*), are often present in more recent graphs. The issue of nil entities is particularly important when evaluating machine learning components with older gold standards. The Reuters 128 corpus, for instance, has been published in 2014 (Röder et al., 2014) and consequently misses entities that haven't been available in DBpedia at annotation time (Brasoveanu et al., 2018).

Also, shifts in a graph's popularity or the need to collaborate with partners that rely on a specific KG may trigger the need to migrate to either a newer KG version or even to another KG (e.g., from DBpedia to Wikidata).

Orbis supports such use cases by recording the history between annotated corpora. It, therefore, supports comparative evaluations and the computation of standard metrics which outline the differences between these annotated corpus versions. Orbis' corpus versioning also tracks relations between corpora, making changes more traceable and explicit (Weichselbraun et al., 2019a).

Figure 6 outlines a semi-automatic process for efficiently translating a language resource to a new KG. An automatic KG translation component aims at linking existing entities to the new KG. Depending on the involved KGs either knowledge rich approaches (e.g., based on `owl:sameAs` links between the KGs) or named entity linking might be deployed at this stage. Afterwards, a named entity recognition component enriches the corpus with candidate entities. Human annotators create a new version of the gold standard by correcting the automatically generated silver standard annotations. Finally, feedback on these corrections is leveraged for improving the machine learning components used in this process.

5 Outlook and Conclusions

This paper introduces the Orbis Annotator framework, a user-friendly, easy to install software that supports users in efficiently annotating language data. Orbis Annotator supports standard use cases through a pre-configured docker image and supports advanced setups through its API. Orbis Annotator also supports use cases that require tracking corpus versions and changes between these ver-

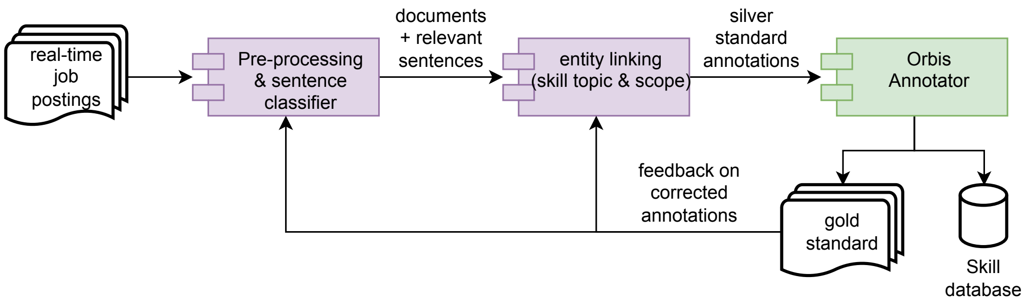

Figure 5: Machine-aided corpus pre-annotation with human feedback.

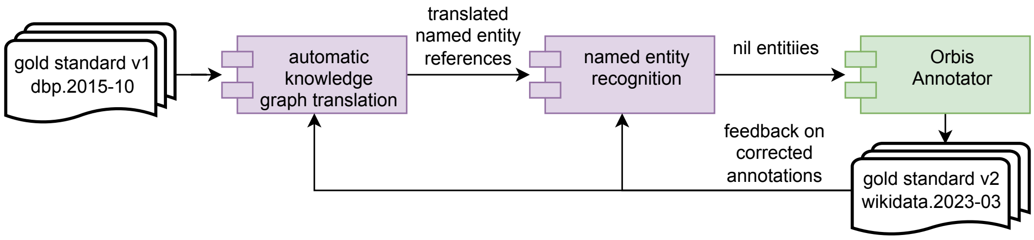

Figure 6: Migrate gold standards to a new knowledge graph or knowledge graph version.

sions. In addition, it aids researchers in tracking and assessing annotator reliability by computing corpus metrics such as inter-rater agreement.

Future work will focus on improving annotator efficiency (e.g., by adding support for additional workflows), and will integrate Orbis Annotator with the Orbis Visual Benchmarking framework. This will enable researchers to conduct evaluations of human, machine and hybrid annotators from within the Orbis Web Interface and to draw upon tools that help in explaining evaluation results such as drill-down analysis and visualizations. Orbis is currently built around JSON, NIF and CSV formats, but since many other formats are used within the research community, we aim at considerably increasing the number of supported formats by integrating software such as Spicy Salmon (Fäth and Chiarcos, 2022) into the toolkit.

Acknowledgements

The research presented in this paper has been conducted within the Future of Work project (https://www.fhgr.ch/future-of-work) funded by Innosuisse. Adrian M.P. Brasoveanu has been partially funded by the Vienna Science and Technology Fund (WWTF) [10.47379/ICT20096] and SDG-HUB (FFG, G.A. No. 892212). The authors would like to thank Philipp Kuntschik, Alexander van Schie, and Marc-Alexander Iten who created an annotation system that has inspired the creation of Orbis Annotator. We would also like to express our gratitude towards Fabian Odoni who has authored an early version of the Orbis visual benchmarking platform.

References

Giuseppe Abrami, Manuel Stoeckel, and Alexander Mehler. 2020. Textannotator: A UIMA based tool for the simultaneous and collaborative annotation of texts. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 891–900. European Language Resources Association.

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. *Lang. Resour. Evaluation*, 47(4):1007–1029.

Adrian Brasoveanu, Giuseppe Rizzo, Philipp Kuntschik, Albert Weichselbraun, and Lyndon J. B. Nixon. 2018. Framing named entity linking error types. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Adrian M.P. Braşoveanu and Răzvan Andonie. 2022. Visualizing and explaining language models. In *Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery*, pages 213–237. Springer.

Hamish Cunningham. 2002. Gate, a general architecture for text engineering. *Comput. Humanit.*, 36(2):223–254.

Tobias Daudert. 2020. A web-based collaborative annotation and consolidation tool. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 7053–7059. European Language Resources Association.

Christian Fäth and Christian Chiarcos. 2022. Spicy salmon: Converting between 50+ annotation formats with fintan, pepper, salt and powla. In *Proceedings of the 8th Workshop on Linked Data in Linguistics within the 13th Language Resources and Evaluation Conference, LDL@LREC 2022, Marseille, France, June 20-25, 2022*, pages 61–68. European Language Resources Association.

David A. Ferrucci and Adam Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348.

Angus G. Forbes, Kristine Lee, Gus Hahn-Powell, Marco Antonio Valenzuela-Escárcega, and Mihai Surdeanu. 2018. Text annotation graphs: Annotating complex natural language phenomena. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Gaël Guibon, Luce Lefeuvre, Matthieu Labeau, and Chloé Clavel. 2022. EZCAT: an easy conversation annotation tool. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 1788–1797. European Language Resources Association.

Sebastian Hellmann, Jens Lehmann, Sören Auer, and Marcus Nitzschke. 2012. NIF combinator: Combining NLP tool output. In *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, volume 7603 of *Lecture Notes in Computer Science*, pages 446–449. Springer.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. *Knowledge Graphs*. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool Publishers.

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, LREC 2022, Marseille, France, 20-25 June 2022*, pages 856–866. European Language Resources Association.

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 *COLING 2018, The 27th International Conference on Computational Linguistics: System Demonstrations, Santa Fe, New Mexico, August 20-26, 2018*, pages 5–9. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.

Tristan Miller, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych. 2013. Dkpro WSD: A generalized uima-based framework for word sense disambiguation. In *51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Proceedings of the Conference System Demonstrations, 4-9 August 2013, Sofia, Bulgaria*, pages 37–42. The Association for Computer Linguistics.

Mariana Neves and Jurica Seva. 2021. An extensive review of tools for manual annotation of documents. *Briefings Bioinform.*, 22(1):146–163.

Minh-Quoc Nghiem and Sophia Ananiadou. 2018. Aplenty: annotation tool for creating high-quality datasets using active and proactive learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 108–113. Association for Computational Linguistics.

Minh-Quoc Nghiem, Paul Baylis, and Sophia Ananiadou. 2021. Paladin: an annotation tool based on active and proactive learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021*, pages 238–243. Association for Computational Linguistics.

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, LREC 2022, Marseille, France, 20-25 June 2022*, pages 3442–3450. European Language Resources Association.

Fabian Odoni, Philipp Kuntschik, Adrian M. P. Brasoveanu, and Albert Weichselbraun. 2018. On the importance of drill-down analysis for assessing gold standards and named entity linking performance. In *Proceedings of the 14th International Conference on Semantic Systems, SEMANTiCS 2018, Vienna, Austria, September 10-13, 2018*, volume 137 of *Procedia Computer Science*, pages 33–42. Elsevier.

Peter Polák, Muskaan Singh, Anna Nedoluzhko, and Ondrej Bojar. 2022. ALIGNMEET: A comprehensive tool for meeting annotation, alignment, and evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 1771–1779. European Language Resources Association.

Michael Raring, Malte Ostendorff, and Georg Rehm. 2022. Semantic relations between text segments for semantic storytelling: Annotation tool - dataset - evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 4923–4932. European Language Resources Association.

Michael Röder, Ricardo Usbeck, Sebastian Hellmann, Daniel Gerber, and Andreas Both. 2014. N^3 - A collection of datasets for named entity recognition and disambiguation in the NLP interchange format. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 3529–3533. European Language Resources Association (ELRA).

Pontus Stenetorp, Sampo Pyysalo, Goran Topic, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for nlp-assisted text annotation. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*, pages 102–107. The Association for Computer Linguistics.

Albert Weichselbraun, Adrian M. P. Brasoveanu, Philipp Kuntschik, and Lyndon J. B. Nixon. 2019a. Improving named entity linking corpora quality. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2019, Varna, Bulgaria, September 2-4, 2019*, pages 1328–1337. INCOMA Ltd.

Albert Weichselbraun, Adrian M. P. Brasoveanu, Roger Waldvogel, and Fabian Odoni. 2020. Harvest - an open source toolkit for extracting posts and post metadata from web forums. In *IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI/IAT 2020, Melbourne, Australia, December 14-17, 2020*, pages 438–444. IEEE.

Albert Weichselbraun, Philipp Kuntschik, and Adrian M. P. Brasoveanu. 2019b. Name variants for improving entity discovery and linking. In *2nd Conference on Language, Data and Knowledge, LDK 2019, May 20-23, 2019, Leipzig, Germany*, volume 70 of *OASIcs*, pages 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 99–105. Association for Computational Linguistics.

Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2018. YEDDA: A lightweight collaborative text span annotation tool. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 31–36. Association for Computational Linguistics.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, Proceedings of the Conference System Demonstrations, 4-9 August 2013, Sofia, Bulgaria*, pages 1–6. The Association for Computer Linguistics.