ASReview Software Documentation

ASReview LAB developers, Utrecht University

Jun 08, 2022

Contents

Welcome to the ASReview LAB Documentation!

Get Started

1.1 What is ASReview LAB?

ASReview LAB is a free (Libre) open-source machine learning tool for screening and systematically labeling a large collection of textual data. It's sometimes referred to as a tool for title and abstract screening in systematic reviews or meta-analyses, but it can handle any type of textual data that must be screened systematically.

ASReview LAB implements three different options:

- **Oracle:** Screen textual data in interaction with the active learning model. The reviewer is the 'oracle', making the labeling decisions.

- **Simulation:** Evaluate the performance of active learning models on fully labeled data.

- **Exploration:** Explore or demonstrate ASReview LAB with a completely labeled dataset. This mode is suitable for teaching purposes.

ASReview LAB is one of the products of the ASReview research project initiated at Utrecht University, which has grown into a vivid community of researchers, users, and developers from around the world.
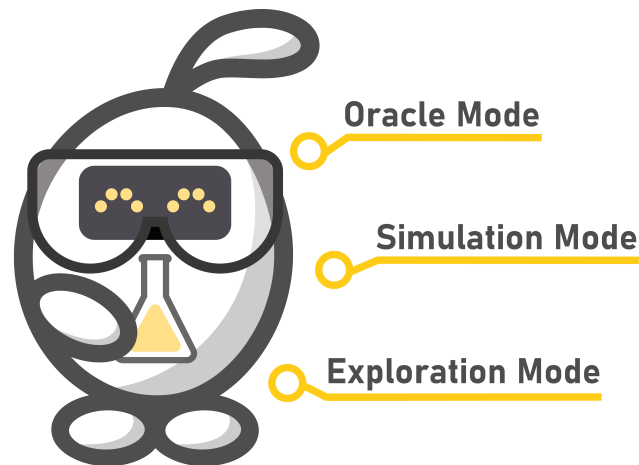
1.2 What is active learning?

Artificial Intelligence (AI) and machine learning has allowed the development of AI-aided pipelines that assist in finding relevant texts for search tasks. A well-established approach to increasing the efficiency of screening large amounts of textual data is screening prioritization through Active Learning: a constant interaction between a human who labels records and a machine learning model which selects the most likely relevant record based on a minimum training dataset. It allows the screening of large amounts of text in an intelligent and time-efficient manner. ASReview LAB, published in Nature Machine Intelligence, has shown the benefits of active learning, reducing up to 95% of the required screening time.

1.3 Labeling workflow with ASReview

Start and finish a systematic labeling process with ASReview LAB by following these steps:

1. Create a dataset with potentially relevant records you want to screen systematically. Improve the quality of the data and specify clear reviewing (inclusion/exclusion) criteria

2. Specify a stopping criterium

3. *Start ASReview LAB*

4. *Create a project*

5. *Import your dataset*

6. *Select Prior Knowledge*

7. Select the four components of the active learning model (feature extractor, classifier, balancing method, query strategy)

8. Wait until the warm up of the AI is ready (the software is extracting the features and trains the classifier on the prior knowledge)

9. Start *Screening* until you reach your stopping criterium

10. *Export results* and *Export Project*

1.4 Quick start

1. Check if Python 3.7 or later is installed (if not, install Python)

```
python --version
```

2. Install ASReview LAB

```
pip install asreview
```

3. Open ASReview LAB

```
asreview lab
```

4. Click Create to create a project.

5. Select a mode (Oracle, Exploration, Simulation)

6. Name the project, and if you want, add an author name(s) and type a description.

7. **Import a dataset** you want to review, or select a benchmark dataset (only available for **Exploration** and **Simulation**).

8. **Add prior knowledge**. Select at least 1 relevant and 1 irrelevant record to warm up the AI. You can search for a specific record or request random records.

9. Select the four components of the active learning model, or rely on the default settings that have shown fast and excellent performance in many simulation studies.

10. ASReview LAB starts extracting the features and runs the classifier with the prior knowledge.

You're ready to start labeling your data! All your labeling actions are automatically saved, so there is no need to click the save button (we don't even have one).

1.5 ASReview LAB terminology

When you do text screening for a systematic review in ASReview LAB, it can be useful to know some basic concepts about systematic reviewing and machine learning to understand. The following overview describes some terms you might encounter as you use ASReview LAB.

Active learning model An active learning model is the combination of four elements: a feature extraction technique, a classifier, a balance, and a query strategy.

ASReview ASReview stands for *Active learning for Systematic Reviews* or *AI-assisted Systematic Reviews*, depending on context. Avoid this explanation, only use as tagline.

ASReview CLI ASReview CLI is the command line interface that is developed for advanced options or for running simulation studies.

Data Data includes *dataset*, prior knowledge, labels, and *notes*.

Dataset A d ataset is the collection of *records* that the *user imports* and *exports*.

ELAS ELAS stands for "Electronic Learning Assistant". It is the name of *ASReview* mascot. It is used for storytelling and to increase explainability.

Export Export is the action of exporting a *dataset* or a *project* from ASReview LAB.

Extension An extension is the additional element to the ASReview LAB, such as the ASReview visualisation extension, or the ASReview CORD-19 extension.

Import Import is the action of importing a *dataset* or a *project* into ASReview LAB.

Model configuration Model configuration is the action of the *user* to configure the *active learning model*.

Note A note is the information added by the *user* in the note field and stored in the *project file*. It can be edited on the History page.

Project A project is a project created in ASReview LAB.

Projects dashboard The project dashboard is the landing page containing an overview of all *projects* in ASReview LAB.

Project file The project file is the `.asreview` file containing the *data* and *model configuration*. The file is *exported* from ASReview LAB and can be *imported* back.

Project mode the project mode includes oracle, simulation, and exploration in ASReview LAB:

> **Oracle** mode is used when a *user* reviews a *dataset* systematically with interactive artificial intelligence (AI).
>
> **Exploration** mode is used when a user explores or demonstrates ASReview LAB with a completely labeled dataset. This mode is suitable for teaching purposes.
>
> **Simulation** mode is used when a user simulates a review on a completely labeled dataset to see the performance of ASReview LAB.

Status The project status is the stage that a *project* is at in ASReview LAB.

> **Setup** refers to the fact that the *user* adds project information, *imports* the *dataset*, selects the prior knowledge, *configures the model* and initiates the first iteration of *model* training.
>
> **In Review** refers to the fact that in oracle or exploration mode, the user adds labels to *records*, or in simulation mode, the simulation is running.
>
> **Finished** refers to the fact that in oracle or exploration mode, the user decides to complete the *reviewing* process or has labeled all the records, or in simulation mode, the simulation has been completed.
>
> **Published** refers to the fact that the user publishes the dataset and *project file* in a repository, preferably with a Digital Object Identifier (DOI).

Record A record is the data point that needs to be labeled. A record can contain both information that is used for training the *active learning model*, and information that is not used for this purpose.

> In the case of systematic reviewing, a record is meta-data for a scientific publication. Here, the information that is used for training purposes is the text in the title and abstract of the publication. The information that is not used for training typically consists of other metadata, for example, the authors, journal, or DOI of the publication.

Reviewing Reviewing is the decision-making process on the relevancy of *records* ("irrelevant" or "relevant"). It is interchangeable with Labeling, Screening, and Classifying.

User The human annotator is the person who labels *records*.

Screener Replacement term when the context is PRISMA-based reviewing.

1.6 Privacy

The ASReview LAB software doesn't collect any information about the usage or user. Great, isn't it!

Research

The open source ASReview LAB software is one of the products of the ASReview research project. The ASReview research project is a fundamental and applied research project studying the application of AI in the field of systematic reviews. The research is conducted by the research team, partners, and contributors. ASReview LAB is stable and validated by the scientific community (Van de Schoot et al., 2020). In short, ASReview LAB is written by researchers for researchers.

The team defined and implements the five fundamental principles of ASReview:

1. Humans are the oracle;

2. Code is open & results are transparent;

3. Decisions are unbiased;

4. The interface shows an AI is at work;

5. Users are responsible for importing high quality data.

2.1 Cite

For scientific use, we encourage you to cite both the software and research project.

- The paper published in Nature Machine Intelligence can be used to cite the **ASReview project**.

- For citing the software **ASReview LAB**, refer to the specific release of the software. The menu on the right (in Zenodo) can be used to find the citation format of prevalence.

- For citing the documentation (or to download the pdf) go to Zenodo.

- More studies related to the project can be found on the asreview.ai/research.

Support

3.1 Donate

The ASReview software is Free and Open Source Software (FOSS). To support the development, you can donate via the ASReview crowdfunding platform. Even small donations are highly appreciated!

3.2 Collaborate

If you are interested in (scientific) collaboration, contact Prof. Dr. Rens van de Schoot or send an email to asreview@uu.nl.

3.3 Contribute

If you discover issues please let us know via Github. If you have ideas to solve your own or other issues, it is highly appreciate to contribute to the development.

Questions can be asked on GitHub Discussions. For bug reports and feature requests, please submit an issue on GitHub.

Installation

4.1 Install ASReview

ASReview software requires an installation of Python 3.7 or later. Detailed step-by-step instructions to install Python (and ASReview) are available for Windows and macOS/Linux users.

Install the ASReview software with Pip by running the following command in the *CMD.exe* (Windows) or *Terminal* (MacOS/Linux):

```
pip install asreview
```

Start the application with the following command (in CMD.exe or Terminal):

```
asreview lab
```

The ASReview LAB software starts in the browser. For more options on starting ASReview LAB, see *Start ASReview LAB*.

Note: See *Troubleshooting* for common problems during installation.

Note: For users with Apple M1 computers, if you experience problems, follow the instructions.

4.2 Upgrade ASReview

Upgrade ASReview software with

```
pip install --upgrade asreview
```

4.3 Uninstall ASReview

Remove ASReview with

```
pip uninstall asreview
```

Enter y to confirm.

> **Warning:** Note that your project files will **not** delete with this action. You find them in the *.asreview* folder in your home folder.

4.4 Server Installation

It is possible to run the ASReview software on a server or custom domain. Use the flags *ip* and *port* for configuration. ASReview should only be used in closed networks.

```
asreview lab --port 5555 --ip xxx.x.x.xx
```

> **Warning:** Don't use the development server in production. Read the Flask documentation about deploying a Flask app to production.

4.5 Install with Docker

ASReview LAB is also available as a Docker container. Make sure you have Docker installed on your machine.

To install and start ASReview LAB at http://localhost:5000, run the following:

```
docker run -p 5000:5000 asreview/asreview:latest
```

More advanced command line options can be given afterward, like this:

```
docker run -p 9000:9000 asreview/asreview  --port 9000
```

> **Tip:** ASReview LAB is now installed. Open the URL in your host web browser: `http://localhost:5000` and get started.

4.5.1 Mount local volume

To mount the container to your local project folder (or any other local folder), the *-v* flag can be used. To do so, adjust path-to-your-folder to your local folder. When a project folder is specified, ASReview LAB will store and load all its projects from this folder. Note that multiple containers can access the same folder.

```
docker create --name asreview-lab -p 5000:5000 -v path-to-your-folder:/project_folder
→asreview/asreview
```

4.5.2 Build a local image

For more information, see ASReview LAB GitHub.

Start ASReview LAB

After you install ASReview LAB, start the program via the command line to start using it.

```
asreview lab
```

When you are using Windows, open *CMD.exe* and run the command. When you use MacOS or Linux, you can open *Terminal* and run the command.

The information in the sections below is more advanced and not needed for the majority of the ASReview LAB users.

5.1 Command line arguments for starting ASReview LAB

ASReview LAB provides a powerful command line interface for running ASReview LAB with other options or even run tasks like simulations. For a list of available commands in ASReview LAB, type `asreview lab --help`.

asreview lab launches the ASReview LAB software (the frontend).

```
asreview lab [options]
```

-h, --help
> Show help message and exit.

--ip IP
> The IP address the server will listen on.

--port PORT
> The port the server will listen on.

--port-retries NUMBER_RETRIES
> The number of additional ports to try if the specified port is not available.

--no-browser NO_BROWSER
> Do not open ASReview LAB in a browser after startup.

--certfile CERTFILE_FULL_PATH
> The full path to an SSL/TLS certificate file.

—keyfile KEYFILE_FULL_PATH
> The full path to a private key file for usage with SSL/TLS.

—embedding EMBEDDING_FP
> File path of embedding matrix. Required for LSTM models.

—clean-project CLEAN_PROJECT
> Safe cleanup of temporary files in project.

—clean-all-projects CLEAN_ALL_PROJECTS
> Safe cleanup of temporary files in all projects.

—seed SEED
> Seed for the model (classifiers, balance strategies, feature extraction techniques, and query strategies). Use an integer between 0 and 2^32 - 1.

The following evironment variables are available.

ASREVIEW_PATH
> The path to the folder with project. Default *~/.asreview*.

5.1.1 Set environment variables

How you set environment variables depends on the operating system and the environment in which you deploy ASReview LAB.

In MacOS or Linux operating systems, you can set environment variables from the command line. For example:

```
export ASREVIEW_PATH=~/.asreview
```

On Windows, you can use the following syntax:

```
set ASREVIEW_PATH=~/.asreview
```

To check if you set an environment variable successfully, run the following on *nix operating systems:

```
echo $ASREVIEW_PATH
```

Or the following on Windows operating systems:

```
echo %ASREVIEW_PATH%
```

5.2 Run ASReview LAB on localhost with a different port

By default, ASReview LAB runs on port 5000. If that port is already in use or if you want to specify a different port, start ASReview LAB with the following command:

```
asreview lab --port <port>
```

For example, start ASReview LAB on port 5001:

```
asreview lab --port 5001
```

Troubleshooting

ASReview LAB is advanced machine learning software. In some situations, you might run into unexpected behavior. See below for solutions to problems.

6.1 Unknown Command "pip"

The command line returns one of the following messages:

```
-bash: pip: No such file or directory
```

```
'pip' is not recognized as an internal or external command, operable program or batch␣
→file.
```

First, check if Python is installed with the following command:

```
python --version
```

If this doesn't return 3.7 or higher, then Python isn't or not correctly installed.

Most likely, the enviroment variables aren't configured correctly. Follow the step-by-step installation instruction on the ASReview website (Windows and MacOS).

However, there is a simple way to deal with correct environment variables by ading *python -m* in front of the command. For example:

```
python -m pip install asreview
```

6.2 Unknown command "asreview"

In some situations, the entry point "asreview" can not be found after installation. First check whether the package is correctly installed. Do this with the command *python -m asreview -h*. If this shows a decription of the program, use

python -m in front of all your commands. For example:

```
python -m asreview lab
```

6.3 Build dependencies error

The command line returns the following message:

```
"Installing build dependencies ... error"
```

This error typically happens when the version of your Python installation has been released very recently. Because of this, the dependencies of ASReview are not compatible with your Python installation yet. It is advised to install the second most recent version of Python instead. Detailed step-by-step instructions to install Python (and ASReview) are available for Windows and MacOS users.

6.4 Remove temporary files

In case ASReview runs into unexpected errors or doesn't work as expected, it is advised to try to remove temporary files from the project first. These files can be found in the `.asreview/` folder in your home directory. However, the easiest way to remove these files is with:

```
asreview lab --clean-all-projects
```

This will safely remove temporay files, nothing will harm your review. To clean a specific project, use

```
asreview lab --clean-project my-project
```

in which `my_project` is your project name.

Prepare your data

ASReview LAB requires a dataset containing a set of textual records (e.g., titles and abstracts of scientific papers, newspaper articles, or policy reports) obtained via a systematic search. The goal is to review all records systematically using predetermined inclusion and exclusion criteria. Also, it should be expected that only a fraction of the records in the dataset is relevant.

Datasets can be unlabeled as well as *Partially labeled data* and *Fully labeled data*. The latter ones are useful in the Simulation and Exploration mode. See *Project modes* for more information.

The easiest way to obtain a dataset is via a search engine or with the help of a reference manager. See *Compatibility* for reference managers export formats supported by ASReview. For more information about the format of the dataset, see *Data format*.

7.1 High quality data

When you import your dataset, remove duplicates and retrieve the text in as many empty fields as possible (See Importance-of-abstracts blog for help). With clean data, you benefit most from what active learning has to offer.

7.2 Compatibility

7.2.1 Citation Managers

The following table provides an overview of export files from citation managers which are accepted by ASReview.

	.ris	.csv	.xlsx
EndNote		N/A	N/A
Excel	N/A		
Mendeley		N/A	N/A
Refworks		N/A	N/A
Zotero			N/A

- = The data can be exported from the citation manager and imported in ASReview.

- N/A = This format does not exist.

RIS files used for screening in ASReview LAB can be imported back into the reference software and the decision labels can be found in the notes field. For more information see this instruction video.

Note: the RIS-pipeline is extensively tested for reference managers Zotero and EndNote. However, it might also work for other reference managers but is currently not supported.

Note: When using EndNote use the following steps to export a RIS file (.ris):

- In EndNote, click on the style selection dropdown menu from the main EndNote toolbar.

- Click "Select Another Style".

- Browse to RefMan (RIS) Export and click "Choose".

- Click on the file menu and select "Export".

- Pick a name and location for the text file.

- Choose the output format RefMan (RIS) Export and click "Save".

7.2.2 Search Engines

When using search engines, it is often possible to store the articles of interest in a list or folder within the search engine itself. Thereafter, you can choose from different ways to export the list/folder. When you have the option to select parts of the citation to be exported, choose the option which will provide the most information.

The export files of the following search engines have been tested for their acceptance in ASReview:

	.ris	.tsv	.csv	.xlsx
CINHAL (EBSCO)	X	N/A	X	N/A
Cochrane		N/A		N/A
Embase		N/A		
Eric (Ovid)	X	N/A	N/A	X
Psychinfo (Ovid)	X	N/A	N/A	X
Pubmed	X	N/A	X	N/A
Scopus		N/A		N/A
Web of Science	X	X	N/A	N/A

- = The data can be exported from the search engine and imported in ASReview.

- N/A = This format does not exist.

- X = Not supported.

Warning: If the export of your search engine is not accepted in ASReview, you can also try the following: import the search engine file first into one of the citation managers mentioned in the previous part, and export it again into a format that is accepted by ASReview.

7.2.3 Systematic Review Software

There are several software packages available for systematic reviewing, see for an overview. Some of them use machine learning, while other focus on screening and management. The overview below shows an overview of alternative software programs and the compatibility with ASReview.

	.ris	.tsv	.csv	.xlsx
Abstrackr		N/A		N/A
Covidence*		N/A		N/A
Distiller	X	N/A	**	**
EPPI-reviewer		N/A	N/A	X
Rayyan		N/A		N/A
Robotreviewer	N/A	N/A	N/A	N/A

- • = The data can be exported from the third-party review software and imported in ASReview.

- N/A = This format does not exist.

- X = Not supported.

* When using Covidence it is possible to export articles in `.ris` format for different citation managers, such as EndNote, Mendeley, Refworks and Zotero. All of these are compatible with ASReview.

** When exporting from Distiller and if the following error occurs `Unable to parse string "Yes (include)" at position 0` set the `sort references by` to `Authors`. Then the data can be imported in ASReview.

Data format

To carry out a systematic review with ASReview on your own dataset, your data file needs to adhere to a certain format. ASReview accepts the following formats:

- **RIS file format** (wikipedia) with extensions `.ris` or `.txt`. RIS file formats are used by digital libraries, like IEEE Xplore, Scopus and ScienceDirect. Citation managers Mendeley, RefWorks, Zotero, and EndNote support the RIS file format as well.

For parsing RIS file format, the software uses a Python RIS files parser and reader (rispy). Successful import/export depends on a proper data set structure. To validate your data set, the complete default mapping can be found on the developer's GitHub page.

Tabular datasets with extensions `.csv`, `.tab`, `.tsv`, or `.xlsx`. CSV and TAB files are preferably comma, semi-colon, or tab-delimited. The preferred file encoding is *UTF-8* or *latin1*.
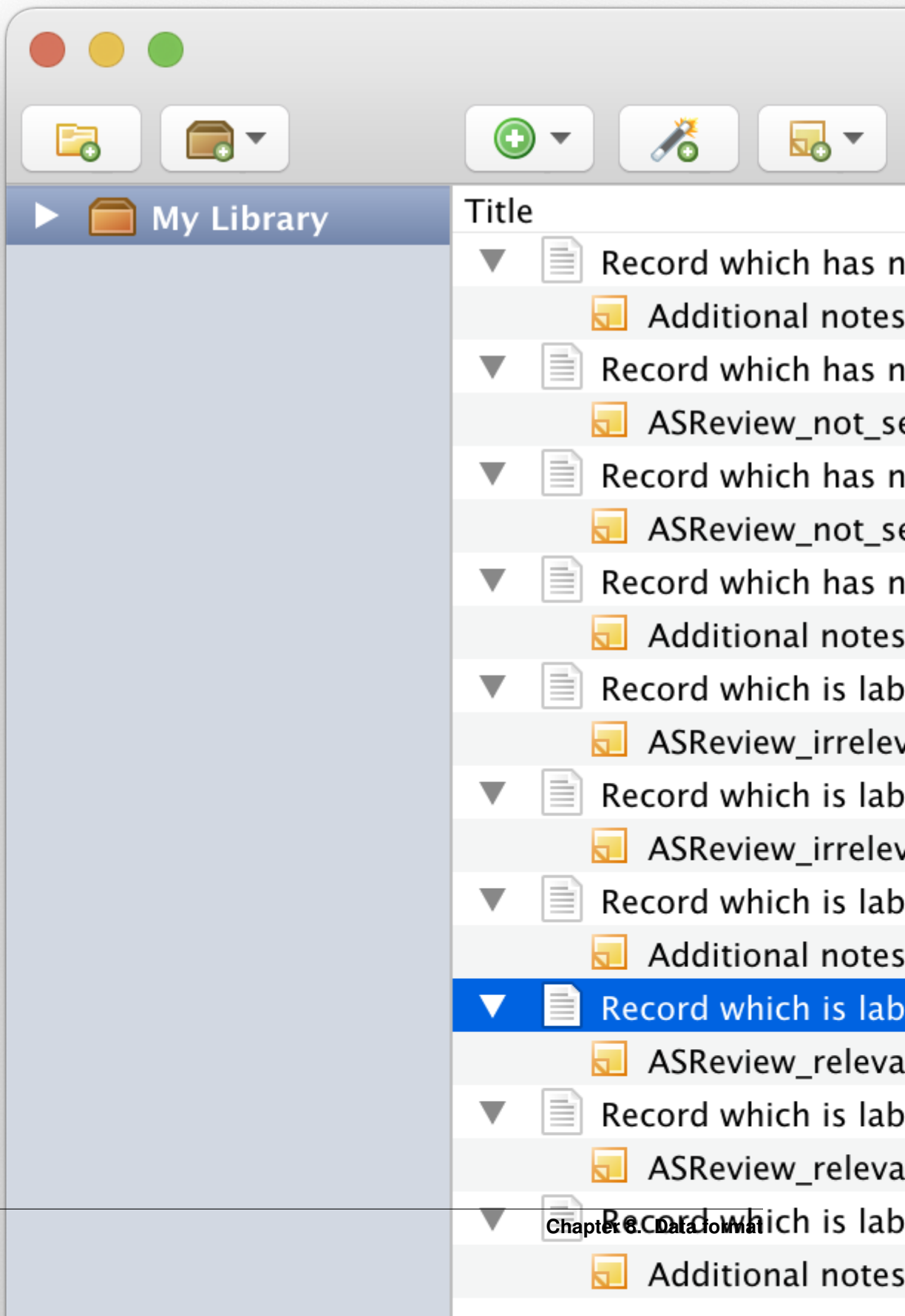
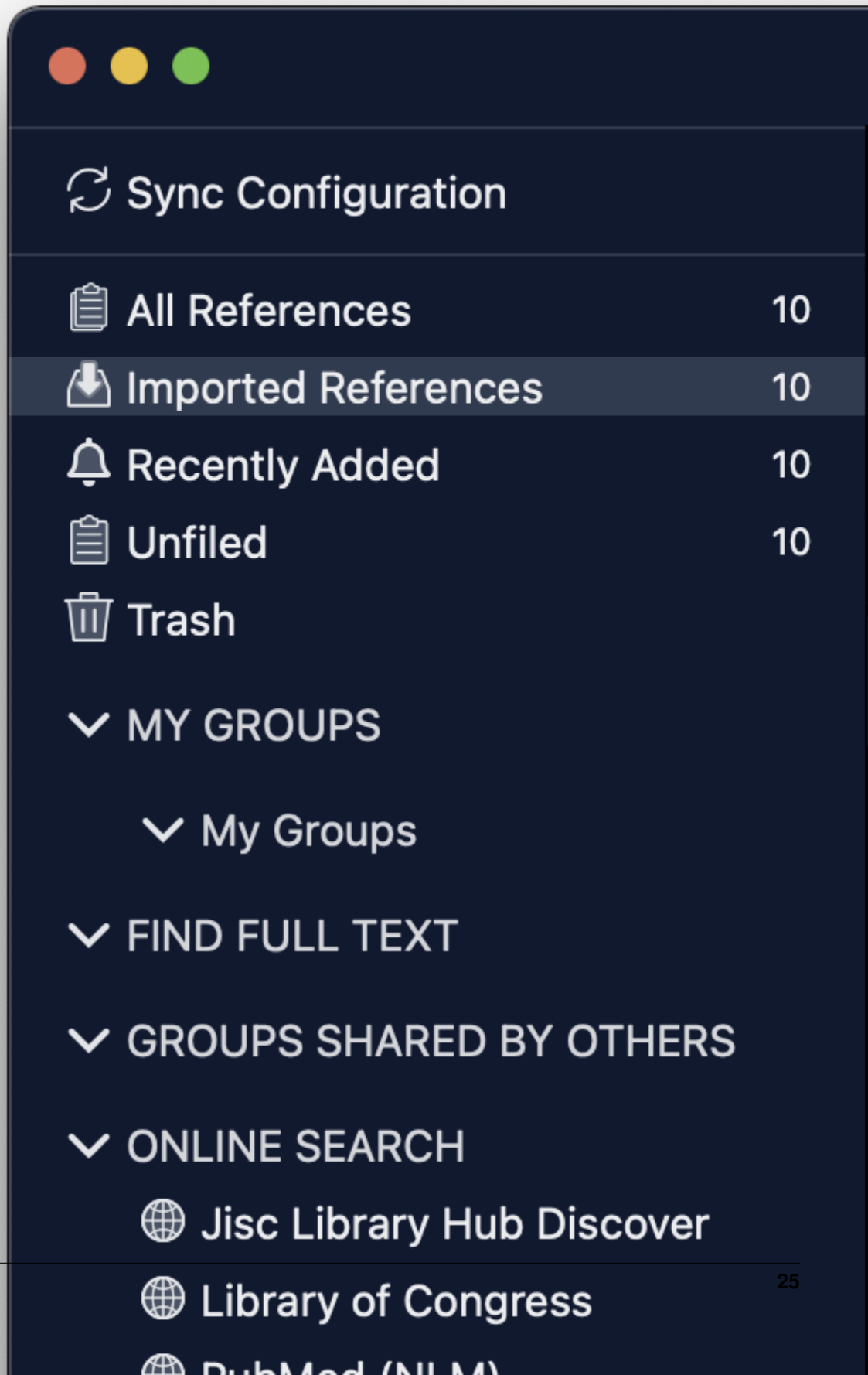For tabular data files, the software accepts a set of predetermined column names:

Table 1: Table with column name definitions

Name	Column names	Mandatory
ID	record_id	no
Title	title, primary_title	yes*
Abstract	abstract, abstract note	yes*
Keywords	keywords	no
Authors	authors, author names, first_authors	no
DOI	doi	no
Included	final_included, label, label_included, included_label, in-cluded_final, included, included_flag, include	no

* Only a title or an abstract is mandatory.

ID If your data contains a column titled `record_id` it needs to consists only of integers, and it should contain no missing data and no duplicates, otherwise you will receive an error. If there is no `record_id` it will be automtically generated by the software. This column can also be used for the Simulation Mode to select prior knowledge.

⟳ Sync Configuration

📋 All References   10

⤓ Imported References   10

🔔 Recently Added   10

📋 Unfiled   10

🗑 Trash

∨ MY GROUPS

    ∨ My Groups

∨ FIND FULL TEXT

∨ GROUPS SHARED BY OTHERS

∨ ONLINE SEARCH

    🌐 Jisc Library Hub Discover

    🌐 Library of Congress

Title, Abstract Each record (i.e., entry in the dataset) should hold metadata on a paper. Mandatory metadata are only `title` or `abstract`. If both title and abstract are available, the text is combined and used for training the model. If the column `title` is empty, the software will search for the next column `primary_title` and the same holds for `abstract` and `abstract_note`.

Keywords, Authors If `keywords` and/or `author` (or if the column is empty: `author names` or `first_authors`) are available it can be used for searching prior knowledge. Note the information is not shown during the screening phase and is also not used for training the model, but the information is available via the API.

DOI If a Digital Object Identifier (`DOI`) is available it will be displayed during the screening phase as a clickable hyperlink to the full text document. Note by using ASReview you do *not* automatically have access to full-text and if you do not have access you might want to read this blog post.

Included A binary variable indicating the existing labeling decisions with `0` = irrelevant/excluded, and `1` = relevant/included. Different column names are allowed, see the table. It can be used for:

- **Screening**: In ASReview LAB, if labels are available for a part of the dataset (see *Fully and partially labeled data*), the labels will be automatically detected and used for prior knowledge. The first iteration of the model will then be based on these decisions and used to predict relevance scores for the unlabeled part of the data.

- **Exploration**: You can explore a completely labeled dataset in the Exploration Mode. The relevant/irrelevant label in the dataset will be displayed on each record. This option is useful for training purposes, presentations, and workshops.

- **Simulation**: In the *ASReview command line interface for simulations*, the column containing the labels is used to simulate a systematic review run. Only records containing labels are used for the simulation, unlabeled records are ignored.

Note: Files exported with ASReview LAB contain the column `included` and can be used for prior knowledge.

CHAPTER 9

Fully and partially labeled data

Fully and partially labeled datasets serve a special role in the ASReview context. These datasets have review decisions for a subset of the records or for each record in the dataset. The labels are dichotomous: relevant or irrelevant. *Partially labeled data* is useful in the Oracle mode, whereas *Fully labeled data* is useful in the Simulation and Exploration mode. See *Project modes* for more information.

All datasets exported from ASReview LAB can be imported into ASReview LAB again. All labels are recognized by the software. In Oracle mode, all labels are directly added as *Prior Knowledge*.

9.1 Labeled data format

For tabular datasets (*e.g., CSV, XLSX*), the dataset should contain a column called "included" or "label" (See *Data format* for all naming conventions), which is filled with 1's or 0's for the records that are already screened. The value is left empty for the records that you haven't screened yet.

For the RIS file format, the labels `ASReview_relevant`, `ASReview_irrelevant`, and `ASReview_not_seen`) can be stored with the N1 (Notes) tag. An example of a RIS file with labels in the N1 tag can be found in the ASReview GitHub repository. All labels in this example are valid ways to label the data. Exported RIS file from ASReview LAB can be imported into ASReview LAB again, and whereafter all labels are recognized.

9.2 Partially labeled data

Note: Useful for Oracle projects. Read more about *Project modes*.

Partially labeled datasets are datasets with a review decision for a subset of the records in the dataset. A partially labeled dataset can be obtained by exporting results from ASReview LAB or other software. It can also be constructed given the format described above.

Partially labeled datasets are useful as the labels will be recognized by ASReview LAB as *Prior Knowledge*, and labels are used to train the first iteration of the active learning model.

Note: Merging labeled with unlabeled data should be done outside ASReview LAB, for example, with *Citation Managers*.

9.3 Fully labeled data

Note: Useful for Simulation and Exploration projects. Read more about *Project modes*.

Fully labeled datasets are datasets with a review decision for each record in the dataset. Fully labeled datasets are useful for exploration or simulation purposes (see also *What is a simulation?* and *Project modes*). See *Benchmark Datasets* for built-in, fully labeled datasets in ASReview LAB.

9.3.1 Benchmark Datasets

The ASReview research project collects fully labeled datasets published open access. The labeled datasets are PRISMA-based reviews on various research topics. They can be useful for benchmark projects such as testing the performance of new active learning models. The datasets and their metadata are available via the Systematic Review Datasets repository. In ASReview LAB, these datasets are referred to as "Benchmark Datasets".

These Benchmark Datasets are directly available in the software. During the *Add Dataset* step of the project setup, there is a panel with all the datasets. The datasets can be selected and used directly. Benchmark datasets are also available via the *Command Line*. Use the prefix `benchmark:` followed by the identifier of the dataset (see Systematic Review Datasets repository). For example, to use the Van de Schoot et al. (2017) dataset, use `benchmark: van_de_schoot_2017`.

You can donate your dataset to the Systematic Review Datasets collection.

Create a project

To start reviewing a dataset with ASReview LAB, you first need to create a project. The project will contain your dataset, settings, labeling decisions, and machine learning models. You can choose from three different project types: Oracle, Exploration, and Simulation. The project setup consists of 4 steps: Project information, Data, Model, and Warm up. The sections below explain each of the steps of the setup.

To create a project:

1. *Start ASReview LAB*.

2. Go to the *Projects dashboard* if you are not already there (http://localhost:5000/projects)

3. Click on the Create on the bottom left

10.1 Project information

In this step of the project setup, step 1, you provide all relevant information about your project as well as the type of project you want (the mode). The sections below provide more information on the input fields. After you complete this step, click next.

10.1.1 Project modes

In this step, you have to select a mode. The default is "Oracle". Most users are looking for this one. Oracle mode is used to screen an unlabeled dataset (it's fine if you already have some labels) with the help of AI. The other two modes, Simulation, and Exploration require fully labeled datasets. They are useful for experts studying the performance of active learning or demonstrating the workings of active learning and ASReview.

In short:

- You have an unlabeled dataset (a few labels is fine) -> Oracle

- You have a *Fully labeled data* -> Simulation or Exploration.

Create a new project

1

Project information

Project information

Mode

Oracle

Review your dataset with

Oracle

Review your dataset with

Chapter 10. Create a project

Exploration

10.1.2 Project details

Provide project details like name of the project, authors (for example, the name of the screener), and a description. You can edit these values later in the *Details* page.

10.2 Data

In this step of the project setup, step 1, you import a dataset and select prior knowledge. Read *Prepare your data* for information about dataset formats. Prior knowledge is used to come up with a first sorting of the dataset.

10.2.1 Add Dataset

Click on *Add* to select a dataset. The data needs to adhere to a *specific format*. You will benefit most from what active learning has to offer with *High quality data*.

Depending on the *Project mode*, you are offered the following options for adding a dataset. Keep in mind that in Oracle mode, your dataset is unlabeled or *Partially labeled data*. For Exploration and Simulation mode, you need *Fully labeled data*.

Note: You will benefit most from what active learning has to offer with *High quality data*.

From File

Drag and drop your file or select your file. Click on Save on the top right.

From URL

Use a link to a dataset on the Internet. For example, a link from this dataset repository. Click on Save on the top right.

From Extension

Oracle and Exploration only. Select a file available via an extension. Click on Save on the top right.

Benchmark Datasets

Simulation and Exploration only. Select one of the *Benchmark Datasets*. Click on Save on the top right.

10.2.2 Select Prior Knowledge

The first iteration of the active learning cycle requires prior knowledge to work. This knowledge is used to train the first model. In this step, you need to provide **at least** one relevant and one irrelevant record in your dataset. To facilitate this, it is possible to search within your dataset. This is especially useful for finding records that are relevant based on your prior knowledge or expertise. You can also let ASReview LAB present you a couple of random documents. This can be useful for finding irrelevant records.

The interface works as follows; on the left, you will see methods to find records to use as prior knowledge, on the right, you will see your selected prior knowledge. If you have **at least** one relevant and one irrelevant record, you can click *Close* and go to the next step.

ASReview LAB

Pro

Prior knowledge

Label at least 1 relevant and 1
the AI. Learn more

(i) Editing prior knowledge

Select a way to add prior k

Search

Random

Search

Let's start with finding a prior relevant document. The most efficient way to do this is by searching for a specific document that you already know is relevant. Click on Search and search your dataset by authors, keywords or title, or a combination thereof. Make sure to be precise with the search terms, as only the first 10 results are shown to you. After entering your search terms, press enter to start searching.

Click the document you had in mind and answer, "Is this record relevant?". Note, don't label all items here. Only the one you are looking for.

The prior knowledge will now show up on the right. There are no restrictions on the number of publications you provide but preferably provide 1-5 relevant records. If you are done, click *Close*.

Random

You also need to provide at least one prior irrelevant document. You can do this by searching it, but this can be challenging as you don't know what you are looking for. One way to find an irrelevant document is by labeling a set of random records from the dataset. Given that the majority of records in the dataset are irrelevant (extremely imbalanced data problem), the records presented here are likely to be irrelevant for your study. Click on random to show a few random records. Indicate for each document whether it is relevant or irrelevant.

The prior knowledge will now show up on the right. Use the buttons to see all prior knowledge or irrelevant items. There are no restrictions on the number of publications you provide but preferably provide 1-5 relevant records. If you are done, click *Close*.

After labeling a couple of randomly selected records, ASReview LAB will ask you whether you want to stop. Click on **STOP** and click **Next**.

10.3 Model

In the next step of the setup, you can select a model. The default settings (Naïve Bayes, TF-IDF, Max) have fast and excellent performance. Most users can skip this step and click *Next*.

10.3.1 Select model (advanced)

It is possible to change the settings of the Active learning model. There are four ingredients that can be changed in the software: the type of classifier, the query strategy, balance strategy, and the feature extraction technique.

The classifier is the machine learning model used to compute the relevance scores. The available classifiers are Naive Bayes, Support Vector Machine, Logistic Regression, and Random Forest. More classifiers can be selected via the *API*. The default is Naive Bayes, though relatively simplistic, it seems to work quite well on a wide range of datasets.

The query strategy determines which document is shown after the model has computed the relevance scores. The three options are: certainty-based, mixed and random. When certainty-based is selected, the documents are shown in the order of relevance score. The document most likely to be relevant is shown first. When mixed is selected, the next document will be selected certainty-based 95% of the time, and randomly chosen otherwise. When random is selected, documents are shown in a random order (ignoring the model output completely). **Warning**: selecting this option means your review is not going to be accelerated by using ASReview.

The feature extraction technique determines the method how text is translated into a vector that can be used by the classifier. The default is TF-IDF (Term Frequency-Inverse Document Frequency) from SKLearn. It works well in combination with Naive Bayes and other fast training models. Another option is Doc2Vec provided by the gensim package which needs to be installed manually. To use it, install the gensim package manually:

ASReview LAB

Pro

Prior knowledge

← schoot

ⓘ Label records that you wa

PTSD Symptom Tra

Disaster Volunteers

Efficacy, Social Ack

and Tasks Carried

Millions of volunteers re

Chapter 10. Create a project

with a 24% to 46% risk c

ASReview LAB

Prior knowledge

Pro

← schoot

(i) Label records that you wa

Prolonged grief dis

and posttraumatic

distinguishable syr

Background: This study

distinctiveness of symp

Disorder (PGD), depress

ASReview LAB

Pro

Prior knowledge

← Show 5 ▼ random

ⓘ Label records that you wa

Trajectory of post-t
following traumatic
follow-up

BACKGROUND: Traumat
of patients each year, ar

traumatic stress disorde

```
pip install gensim
```

It takes relatively long to create a feature matrix with this method. However, this only has to be done once per simulation/review. The upside of this method is the dimension-reduction that generally takes place, which makes the modeling quicker.

During the screening phase, it is not possible to change the model. However, it is possible to select a first model, screen part of the data, and export the dataset with the labeling decisions of the first model. This partly-labeled dataset can be imported into a new project and the labels based on the first model will be recognized as prior knowledge. Then, a second model can be trained on the partly-labeled data, and the new predictions will be based on the second model.

10.4 Warm up

In the last step of the setup, step 4, ASReview LAB trains a model and ranks the records in your dataset. Depending on the model and the size of your dataset, this can take a couple of minutes (or even longer). After the project is successfully initialized, you can start reviewing.

Note: In Simulation mode, this step starts the simulation. As simulations usually take longer to complete, the simulation will run in the background. After a couple of seconds, you will see a message and a button "Got it". You will navigate to the *Analytics* page, where you can follow the progress (see *Refresh* button on the top right)

ASReview LAB

Pro

Create a new project

Project information

Screening

Note: Only for Oracle and Exploration projects. Read more about *Project modes*.

11.1 Introduction

As soon as your project is initiated, you can start reviewing. Click on *Review* in the left menu if your project is not on the review page yet. ASReview LAB presents you a title and abstract to screen and label.

You are asked to make a decision: relevant or irrelevant?

Click on the decision of choice. A new record is presented to you. While you review the records, ASReview LAB continuously improves its understanding of your decisions, constantly updating the underlying ordering of the records.

As you keep reviewing documents and providing labels, you will probably see fewer relevant records. When to stop screening is left to you. See *Progress and results* for more information on progress monitoring and information on when to stop.

Note: If you are in doubt about your decision, take your time as you are the oracle. Based on your input, a new model will be trained in the background.

11.1.1 Screening in Exploration mode

In Exploration mode, a blue bar is displayed on top of the record. The blue bar indicates whether the record is relevant or irrelevant.

☰ **ASReview LAB**

← Projects

Your project

Example of Exploration mode

📊 Analytics

📋 **Review**

🕘 History

☰ **ASR**eview LAB

← Projects

IN REVIEW

Your project
Example of Exploration mode

📊 Analytics

📋 **Review**

🕐 History

11.2 Autosave

Your decisions are saved automatically into your ASReview project file. There is no need to press any buttons to save your work anywhere in ASReview LAB.

11.3 Change decisions

In some cases, you might want to change your previous decision. The screening interface of ASReview LAB offers two options to change your decision.

11.3.1 Undo last decision

You can return to your previous decision during screening. You can disable this option in the Settings menu.

1. *Start ASReview LAB*.

2. Open or *Create a project*.

3. Label the record displayed in the screen as relevant or irrelevant.

4. Click on **Undo** (At the bottom right)

5. Click on **Keep (ir)relevant** or **Convert to (ir)relevant**

6. Continue screening.

11.3.2 Screening history

An overview of your decisions made during screening can be found on the **History** page. You can change decisions on this page.

1. *Start ASReview LAB*.

2. Open or *Create a project*.

3. Click on History in the menu on the left.

Changing decisions on the history page

4. To change a label of a record, click the heart icon. The next iteration of the model will take the new label into account.

11.4 Full Text

If a column with Digital Object Identifiers (DOI) or URLs is available in the metadata of your dataset, ASReview LAB will display the DOI and URL during screening. Most of the time, DOIs point to the full-text of a publication. See *datasets* for more information on including DOI and URL values to your datasets.

(i) Labeled as irrelevant in the dataset

World Trade Center disa
A longitudinal study of d

doi   🔗

Purpose: The current study examine
spent working on the site, dust clou

11.5 Keyboard shortcuts

ASReview LAB supports the use of keyboard shortcuts during screening. The table below lists the available keyboard shortcuts.

Note: Keyboard shortcuts are only available when the **Undo** feature has been enabled in the Settings (bottom left).

You can press a key (or a combination of keys) to label a record as relevant or irrelevant, or to return to the previous decision during screening. By default, keyboard shortcuts are disabled.

Action	Shortcut
Label record as relevant	**r** or **Shift + r**
Label record as irrelevant	**i** or **Shift + i**
Return to previous decision	**u** or **Shift + u**

11.6 Display

11.6.1 Dark mode

ASReview LAB offers the option to customize the screening appearance and functionality.

1. *Start ASReview LAB*.

2. Click on *Settings* (bottom left).

3. Go to *Display* and toggle the dark mode

Note: Your preference is saved in the browser.

11.6.2 Font size

You can make the text on the review screen smaller or larger.

1. *Start ASReview LAB*.

2. Click on *Settings* (bottom left).

3. Go to *Display* and click on *Font size*.

4. Slide the slider to the desired font size.

Progress and results

During screening, you might want to keep track of your progress and know when to stop. This section provides documentation on useful tools for these purposes.

12.1 Analytics

ASReview LAB offers some insightful statistics, a progress chart, and recall chart to keep track of your screening process and help you to decide when to stop screening.

To open the statistics panel:

1. *Start ASReview LAB*.

2. Open or *Create a project*.

3. Click on Analytics on in the left menu.

12.1.1 Summary statistics

The summary statistics are counts of the records in your dataset.

- Total records: the total number of records in your dataset.

- Labeled records: the number of records that you labeled as relevant or irrelevant, including those you added as prior knowledge.

- Relevant records: the number of records that you labeled as relevant, including those you added as prior knowledge.

- Irrelevant records: the number of records that you labeled as irrelevant, including those you added as prior knowledge.

- Irrelevant since last relevant: the number of irrelevant records you have seen since the last relevant record. The larger this number (relatively), the more likely you have seen the relevant records.

Analytics

12.1.2 Charts

The charts on the analytics page can be useful to monitor your progress. There is a *Progress* and a *Recall* chart. You will probably see a decline in the number of relevant items you find. This helps to decide when to stop. The charts do not include prior knowledge and are most relevant after you have screened a couple of hundreds of records.

Progress chart

The progress chart plots the number of relevant records in the last 10 records that you reviewed in ASReview LAB. For example, when you reviewed 100 records, you labeled 3 relevant records between the 91st and 100th reviewed records.

Recall chart

The recall chart plots the number of relevant records against the number of records that you reviewed in ASReview LAB. Relevant by ASReview LAB refers to the relevant records that you labeled with the assistance of the active learning model. Random relevant refers to the relevant records that you might find if you manually reviewed all the records without the assistance of the active learning model.

12.2 Stop screening

The blogpost *ASReview Class 101* and the How to stop screening? discussion provide tips on when to stop with screening.

12.3 Mark project as finished

When you decide to stop screening, you can mark the project as finished. You can undo this at any time. To mark your project as finished:

1. *Start ASReview LAB*.

2. Go to the *Projects dashboard* (http://localhost:5000/projects)

3. Hover the project you want to mark as finished and click on *Options*.

4. Click on *Mark as finished*.

The button to continue screening is now disabled. This can be undone by clicking again on *Mark as in review*.

12.4 Export results

You can export the results of your screening to a RIS, CSV, TSV or Excel file. A file contains all imported data including your decisions. The file is ordered as follows:

1. All relevant records you have seen in the order they were shown during the screening process.

2. All records not seen during the screening and ordered from most to least relevant according to the last iteration of the model.

3. All non-relevant records are presented in the order these are shown during the screening proces.

To download your results follow these steps:

1. *Start ASReview LAB*.

2. Open or *Create a project*.

3. Click on *Export* in the menu on the left.

4. Select *Dataset*.

5. Select the file type for prefer: i.e. Excel, RIS, TSV, or CSV file.

6. Save the file to your device.

Note: A RIS file can only be exported if a RIS file is imported.

The following variables will be added to your dataset:

- The column titled **included** contains the labels as provided by the user: `0` = not relevant, `1` = relevant and if missing it means the record is not seen during the screening process.

- The column titled **asreview_ranking** contains an identifier to preserve the rank ordering as described above.

ASReview LAB

← Projects

Your project

simulate_van_de_schoot_2017

Analytics

History

Export

Export

Export

Manage projects

ASReview LAB offers the options to import and export projects. This can be useful for sharing results, archiving projects, and for backup purposes.

13.1 Import Project

To import a project:

1. *Start ASReview LAB*.

2. Go to the *Projects dashboard* (http://localhost:5000/projects)

3. Click on the *Import project* icon on the top right.

4. Click on *Select file* and select a project from your device (with `.asreview` extension).

5. Open the project from the *Projects dashboard*.

13.2 Export Project

The ASReview project file (extension `.asreview`) can be exported from ASReview LAB. The file contains the dataset, review history, notes, and model configuration. It can be imported into ASReview LAB on a different device, which allows other users to replicate the project, or continue the systematic review.

To export your project:

1. *Start ASReview LAB*.

2. Go to the *Projects dashboard* (http://localhost:5000/projects)

3. Hover the project you want to export and click on the *Export* icon.

4. Click on *Select file* and click on *Project*.

5. Click on *Export*

You will be asked where to save the ASReview file (extension *.asreview*).

13.3 Delete Project

To permanently delete a project, including ALL files:

1. *Start ASReview LAB*.

2. Go to the *Projects dashboard* (http://localhost:5000/projects)

3. Hover the project you want to export and click on *Options*.

4. Click on *Delete forever*.

5. This action cannot be made undone, ASReview LAB will ask you to confirm by typing in the project title.

Extensions

ASReview has extensive support for extensions. They can extend the functionality of ASReview LAB, and the *Command Line Interface*. There are *officially supported extensions* and *community* maintained extensions.

Looking to develop your own extension? See *Create extensions* for detailed instructions.

14.1 Installation

Most extensions are installable from PyPI (the same way ASReview LAB is installed) or GitHub. It is preferred to follow the installation instructions provided by the extension.

The following example shows the installation of ASReview Insights, an extension for plotting and computing metrics for simulations in ASReview.

```
pip install asreview-visualization
```

Extension (only) published on Github can be installed directly from the repository. Replace *{USER_NAME}* and *{REPO_NAME}* by the corresponding values of the extension.

```
pip install git@github.com:{USER_NAME}/{REPO_NAME}.git
```

14.2 Supported Extensions

The following extensions are officially supported and maintained by the maintainers of ASReview LAB. They are extensively tested and integrate well with ASReview LAB.

- **ASReview Datatools**

    - ASReview-datatools: Tool for describing, cleaning (input) data, and converting file formats via the command line.

- **ASReview Insights**

- – ASReview-insights: Advanced insights to ASReview simulations like performance plots and metrics.
- **ASReview Wordcloud**
  - – ASReview-wordcloud: Create wordclouds to visualize the contents of datasets.

14.3 List of extensions for ASReview LAB

The List of extensions for ASReview LAB on the Discussion platform gives an overview of known extensions to ASReview LAB and other useful tools in the AI-aided systematic review pipeline. These extensions can extend the software with new models, subcommands, and datasets.

Overview

15.1 What is a simulation?

A simulation involves mimicking the screening process with a certain model. As it is already known which records are labeled relevant, the software can automatically reenact the screening process as if a human was labeling the records.

15.2 Why run a simulation?

Simulating with ASReview LAB has multiple purposes. First, the performance of one or multiple models can be measured by different metrics (see *Analyzing results*). A convenient one is that you can investigate the amount of work you could have saved by using active learning compared to your manual screening process.

Suppose you don't know which model to choose for a new (unlabeled) dataset. In that case, you can experiment with the best performing combination of the classifier, feature extraction, query strategy, and balancing and test the performance on a labeled dataset with similar characteristics.

You could also use the simulation mode to benchmark your own model against existing models for different available datasets. ASReview LAB allows for adding new models via a template.

You can also find 'odd' relevant records in a 'classical' search. Such records are typically found isolated from most other records and might be worth closer inspection

15.3 Datasets for simulation

Simulations require *fully labeled datasets* (labels: 0 = irrelevant, 1 = relevant). Such a dataset can be the result of an earlier study. ASReview offers also fully labeled datasets via the benchmark platform. These datasets are available via the user interface in the *Data* step of the setup and in the command line with the prefix *benchmark:* (e.g. *benchmark:van_de_schoot_2017*).

> **Warning:** When you import your data, make sure to remove duplicates and to retrieve as many abstracts as possible (See Importance-of-abstracts blog for help). With clean data you benefit most from what *active learning* has to offer.

15.4 Simulating with ASReview LAB

ASReview LAB offers three different solutions to run simulations:

- With the *webapp (the frontend)*
- With the *command line interface*
- With the *Python API*

15.5 Simulate with webapp

To run a simulation in the ASReview webapp, create a project as described in *Create a project*. Most of the steps of the setup are identical or straightworward. In this section, some of the differences are highlighted.

In the step on *Project Information*, select the "Simulation" mode (see figure below).

In the step *Data*, import a *fully labeled dataset* or use one of the benchmark datasets.

Selecting prior knowledge is relatively easy. In case you know relevant records to start with, use the search function. In case you don't, use the *Random* option. Toggle the button "Relevant" on top to see some random irrelevant records. Label some relevant and some irrelevant records.

The step *Warm up* is differs slightly from the Oracle and Exploration mode. This step start the simulation, after some seconds, it will return "Got it". This means, the simulation runs further in the background. You are returned to the Analytics page.

This page now has a refresh button on the top right. If the simulation is not finished yet, you can refresh the page or use the refresh button to follow the progress. After a while, the Elas mascotte on the left will hold a sign with "finished". Your simulation is now finished and you can study the results in the analytics page.

15.6 Analyzing results

After a simulation, the results are stored in the ASReview project file (extension *.asreview*). This file contains a large number of variables and logs on the simulation. The data can be extracted from the project file via the API or with one of the available extensions. See *these examples on the Project API* for more information about opening the project file. An easier solution would be to use one of the extensions. ASReview Insights is a useful example.

The extension ASReview Insights offers useful tools, like plotting functions and metrics, to analyze results of a simulation.

Install ASReview Insights directly from PyPi:

```
pip install asreview-insights
```

Detailed documention can found on the ASReview Insights GitHub page.

The following command returns the recall at any moment during the simulation:

Create a new project

1

Project information

Project information

Mode

Simulation

Simulate a review on a c

Oracle

Review your dataset with

Exploration

Dataset

Add a dataset from

○ File   ○ URL   ◉

The benchmark datasets w
your dataset to the benchm

Datasets used in the va

van de Schoot et al. (2

The GRoLTS-Checklist

DOI: 10.1080/10705511.

≡ **ASR**eview LAB

▦ **Projects**

Dataset

Add a dataset from

○ File   ○ URL   ◉

The benchmark datasets w
your dataset to the benchm

Datasets used in the val

van de Schoot et al. (2

The GRoLTS-Checklist

DOI: 10.1080/10705511.

License: CC-BY Attributic

ASReview LAB

Projects

Create a new project

Project information

```
asreview plot recall MY_SIMULATION.asreview
```

Simulation via command line

ASReview LAB comes with an extensive simulation interface via the command line.

16.1 Getting started

The simulation command line tool can be accessed directly like:

```
asreview simulate MY_DATASET.csv --state_file MY_SIMULATION.asreview
```

This performs a simulation with the default active learning model, where `MY_DATASET.csv` is the path to the fully labeled dataset you want to simulate. The result of the simulation is stored, after a succesful simulation, at `MY_SIMULATION.asreview` where `MY_SIMULATION` is the filename you prefer.

Note: For instructions on preparing your fully labeled data, see *Prepare your data*.

16.2 Simulation options

ASReview LAB provides an extensive simulation interface via the command line. An overview of the options are found on the *ASReview command line interface for simulation* page. This section highlights some of the most used options. When no additional arguments are specified in the `asreview simulate` command, default settings are used.

- To make your simulations reproducible you can use the `--seed` and `--init_seed` options. 'init_seed' controls the starting set of papers to train the model on, while the 'seed' controls the seed of the random number generation that is used after initialization.

- By default, the model initializes with one relevant and one irrelevant record. You can set the number of priors by `--n_prior_included` and `--n_prior_excluded`. However, if you want to initialize your model with a specific set of starting papers, you can use `--prior_idx` to select the indices of the papers you want to start the simulation with.

- The `--n_instances` argument controls the number of records that have to be labeled before the model is retrained, and is set at 1 by default. If you want to reduce the number of training iterations, for example to limit the size of your state file and the time to simulate, you can increase `--n_instances`.

- You can select a classifier with the `-m` flag, which is set to be Naive Bayes by default. Names for implemented classifiers are listed on the *Classifiers* table.

- Implemented query strategies are listed on the *Query Strategies* table and can be set with the `-q` option.

- For feature extraction, supply the `-e` flag. Default is TF-IDF, more details on the table for *Feature Extraction*.

- The last element that can be changed is the *Balance Strategies*, and is changed with the `-b` flag. Default is double balance.

Simulate with Python API

The API is still under development and can change at any time without warning.

For more control over the workings of the ASReview software, the ASReview Python API can be used. For example, it is possible to use custom models or implement different sampling strategies. This example shows how to simulate a review with the ASReview API and store the results in an ASReview project file.

Please keep in mind that the ASReview API is experimental at the moment. Improvements and simplifications are planned.

```
[1]: from pathlib import Path

     from asreview import ASReviewData, ASReviewProject
     from asreview.review import ReviewSimulate
```

Create a temporary folder for the results and examples in this document.

```
[2]: project_path = Path("tmp_data")
     project_path.mkdir(exist_ok=True)
```

Create an ASReviewProject to store the results

```
[3]: # Create a project object and folder
     project = ASReviewProject.create(
         project_path=project_path / "api_simulation",
         project_id="api_example",
         project_mode="simulate",
         project_name="api_example"
     )
```

Add a dataset to the project folder in the folder `data` (can also be stored somewhere else, but it is advised to use the data folder). In the following example, a dataset is downloaded from the benchmark platform with CURL (macOS, Unix systems).

```
[4]: %%bash
     curl https://raw.githubusercontent.com/asreview/systematic-review-datasets/master/
     ↪datasets/van_de_Schoot_2017/output/van_de_Schoot_2017.csv > tmp_data/api_simulation/
     ↪data/van_de_Schoot_2017.csv
```

```
  % Total     % Received % Xferd  Average Speed   Time    Time     Time  Current
                                   Dload  Upload   Total   Spent    Left  Speed
100  9.9M  100  9.9M    0      0  13.2M      0 --:--:-- --:--:-- --:--:-- 13.2M
```

Add the reference to the dataset to the project.

```
[5]: project.add_dataset("van_de_Schoot_2017.csv")
```

Setup the models.

```
[6]: from asreview.models.classifiers import NaiveBayesClassifier
     from asreview.models.query import MaxQuery
     from asreview.models.balance import DoubleBalance
     from asreview.models.feature_extraction import Tfidf


     # Select models to use
     train_model = NaiveBayesClassifier()
     query_model = MaxQuery()
     balance_model = DoubleBalance()
     feature_model = Tfidf()
```

Run the simulation with the `ReviewSimulate` class.

```
[7]: data_obj = ASReviewData.from_file(Path("tmp_data", "api_simulation", "data", "van_de_
     ↪Schoot_2017.csv"))
```

```
[8]: # Initialize the simulation reviewer
     reviewer = ReviewSimulate(
         as_data=             data_obj,
         model=               train_model,
         query_model=         query_model,
         balance_model=       balance_model,
         feature_model=       feature_model,
         n_instances=         10,
         project=             project,
         n_prior_included=    1,
         n_prior_excluded=    1,
     )
```

```
[9]: # Start the review process
     project.update_review(status="review")
     try:
         reviewer.review()
         project.mark_review_finished()
     except Exception as err:
         project.update_review(status="error")
         raise err
```

Export the project to a location of choice, in this case `tmp_data/api_example.asreview`.

```
[10]: # Finish and export the project
      project.export(Path("tmp_data", "api_example.asreview"))
```

The following code removes the temporary folder that was created:

```
[11]: import shutil
      shutil.rmtree(project_path)
```

Overview

The development section is meant for users that need advanced functions of ASReview LAB and for developers. It contains technical information on the usage, instructions for developing extensions, and an extensive API reference.

18.1 ASReview architecture

ASReview provides users an API to interact directly with the underlying ASReview machinery. This provides researchers an interface to study the behavior of algorithms and develop custom workflows. The following overview shows the available interfaces for interacting with the ASReview software:

- Layer 5: ASReview CLOUD

    - ASReview is currently in development. For information on ASReview CLOUD, be sure visit our communication channels.

- Layer 4: *ASReview LAB*

    - ASReview LAB is the user friendly webapp and all underlying interfaces. Documentation on LAB can be found in the *ASReview LAB section*.

- Layer 3: REST API

    - The REST API uses a Flask REST API to provide a method to let the React webapp communicate with the backend and algorithms. The REST API is not documented and should be considered 'internal use only'.

- Layer 2: *Command Line*

    - The Command Line is an interface used to open ASReview LAB, run simulations, and run *Subcommand extensions* for ASReview. This development section documents all available command line options for both ASReview LAB and simulation mode.

- Layer 1: *API Reference*

    - The ASReview API is a low level Python interface for ASReview. This interface requires detailed knowledge about the workings of the software. This reference contains extensive documentation on all functions, classes, and modules found in ASReview.

ASReview
CLOUD*
*in development

ASReview

LAB

– An outline for usage can be found in *Simulate with Python API* and *Access data from ASReview file*.

18.2 Extensions

The Create an extension section documents the creation of model, subcommand, and dataset extensions for ASReview. More information on extensions can be found in the extension *Extensions*.

Command Line

ASReview provides a powerful command line interface for running tasks like simulations. For a list of available commands, type `asreview --help`.

19.1 LAB

asreview lab launches the ASReview LAB software (the webapp).

```
asreview lab [options]
```

--ip IP
   The IP address the server will listen on.

--port PORT
   The port the server will listen on.

--port-retries NUMBER_RETRIES
   The number of additional ports to try if the specified port is not available.

--no-browser NO_BROWSER
   Do not open ASReview LAB in a browser after startup.

--certfile CERTFILE_FULL_PATH
   The full path to an SSL/TLS certificate file.

--keyfile KEYFILE_FULL_PATH
   The full path to a private key file for usage with SSL/TLS.

--embedding EMBEDDING_FP
   File path of embedding matrix. Required for LSTM models.

--clean-project CLEAN_PROJECT
   Safe cleanup of temporary files in project.

--clean-all-projects CLEAN_ALL_PROJECTS
   Safe cleanup of temporary files in all projects.

--seed SEED
    Seed for the model (classifiers, balance strategies, feature extraction techniques, and query strategies). Use an integer between 0 and 2^32 - 1.

-h, --help
    Show help message and exit.

19.2 Simulate

asreview simulate measures the performance of the software on existing systematic reviews. The software shows how many papers you could have potentially skipped during the systematic review. You can use *your labeled dataset*

```
asreview simulate [options] [dataset [dataset ...]]
```

or one of the *benchmark-datasets* (see index.csv for dataset IDs).

```
asreview simulate [options] benchmark: [dataset_id]
```

Examples:

```
asreview simulate YOUR_DATA.csv --state_file myreview.asreview
```

```
asreview simulate benchmark:van_de_Schoot_2017 --state_file myreview.asreview
```

dataset
    A dataset to simulate

-m, --model MODEL
    The prediction model for Active Learning. Default: nb. (See available options below: *Classifiers*)

-q, --query_strategy QUERY_STRATEGY
    The query strategy for Active Learning. Default: max. (See available options below: *Query strategies*)

-b, --balance_strategy BALANCE_STRATEGY
    Data rebalancing strategy. Helps against imbalanced datasets with few inclusions and many exclusions. Default: double. (See available options below: *Balance strategies*)

-e, --feature_extraction FEATURE_EXTRACTION
    Feature extraction method. Some combinations of feature extraction method and prediction model are not available. Default: tfidf. (See available options below: *Feature extraction*)

--embedding EMBEDDING_FP
    File path of embedding matrix. Required for LSTM models.

--config_file CONFIG_FILE
    Configuration file with model settings and parameter values.

--seed SEED
    Seed for the model (classifiers, balance strategies, feature extraction techniques, and query strategies). Use an integer between 0 and 2^32 - 1.

--n_prior_included N_PRIOR_INCLUDED
    The number of prior included papers. Only used when prior_idx is not given. Default 1.

--n_prior_excluded N_PRIOR_EXCLUDED
    The number of prior excluded papers. Only used when prior_idx is not given. Default 1.

--prior_idx [PRIOR_IDX [PRIOR_IDX ...]]
  Prior indices by rownumber (0 is first rownumber).

--prior_record_id [PRIOR_RECORD_ID [PRIOR_RECORD_ID ...]]
  Prior indices by record_id.

--state_file STATE_FILE, **-s** STATE_FILE
  Location to ASReview project file of simulation.

--init_seed INIT_SEED
  Seed for setting the prior indices if the prior_idx option is not used. If the option prior_idx is used with one or more index, this option is ignored.

--n_instances N_INSTANCES
  Number of papers queried each query.Default 1.

--stop_if STOP_IF
  The number of label actions to simulate. Default, 'min' will stop simulating when all relevant records are found. Use -1 to simulate all labels actions.

-w WRITE_INTERVAL, **--write_interval** WRITE_INTERVAL
  The simulation data will be written away after each set of thismany labeled records. By default only writes away data at the endof the simulation to make it as fast as possible.

--verbose VERBOSE, **-v** VERBOSE
  Verbosity

-h, --help
  Show help message and exit.

Note: Some classifiers (models) and feature extraction algorithms require additional dependecies. Use `pip install asreview[all]` to install all additional dependencies at once.

19.2.1 Feature Extraction

Name	Reference	Requires
tfidf	`asreview.models.feature_extraction.Tfidf`	
doc2vec	`asreview.models.feature_extraction.Doc2Vec`	gensim
embedding-idf	`asreview.models.feature_extraction.EmbeddingIdf`	
embedding-lstm	`asreview.models.feature_extraction.EmbeddingLSTM`	
sbert	`asreview.models.feature_extraction.SBERT`	sentence_transformers

19.2.2 Classifiers

Name	Reference	Requires
nb	`asreview.models.classifiers.NaiveBayesClassifier`	
svm	`asreview.models.classifiers.SVMClassifier`	
logistic	`asreview.models.classifiers.LogisticClassifier`	
rf	`asreview.models.classifiers.RandomForestClassifier`	
nn-2-layer	`asreview.models.classifiers.NN2LayerClassifier`	tensorflow
lstm-base	`asreview.models.classifiers.LSTMBaseClassifier`	tensorflow
lstm-pool	`asreview.models.classifiers.LSTMPoolClassifier`	tensorflow

19.2.3 Query Strategies

Name	Reference	Requires
max	`asreview.models.query.MaxQuery`	
random	`asreview.models.query.RandomQuery`	
uncertainty	`asreview.models.query.UncertaintyQuery`	
cluster	`asreview.models.query.ClusterQuery`	
max_random	`asreview.models.query.MaxRandomQuery`	
max_uncertainty	`asreview.models.query.MaxUncertaintyQuery`	

19.2.4 Balance Strategies

Name	Reference	Requires
simple	`asreview.models.balance.SimpleBalance`	
double	`asreview.models.balance.DoubleBalance`	
undersample	`asreview.models.balance.UndersampleBalance`	

19.3 Algorithms

`asreview algorithms` provides an overview of all available active learning model elements (classifiers, query strategies, balance strategies, and feature extraction algorithms) in ASReview.

```
asreview algorithms
```

Note: `asreview algorithms` included models added via extensions. See *Create extensions* for more information on extending ASReview with new models via extensions.

Access data from ASReview file

The API is still under development and can change at any time without warning.

Data generated using ASReview LAB is stored in an ASReview project file. Via the ASReview Python API, there are two ways to access the data in the ASReview (extension .asreview) file: Via the *project-API* and the state-API. The project API is for retrieving general project settings, the imported dataset, the feature matrix, etc. The state API retrieves data related directly to the reviewing process, such as the labels, the time of labeling, and the classifier used.

20.1 Example Data

To illustrate the ASReview Python API, the benchmark dataset van_de_Schoot_2017 is used. The project file example.asreview can be obtained by running asreview simulate benchmark: van_de_Schoot_2017 -s example.asreview.

The ASReview Python API can be used for project files obtained via the Oracle, Exploration, and Simulation mode.

20.2 Python Imports

```python
[1]: import shutil
     from pathlib import Path

     import pandas as pd
     from asreview import open_state
     from asreview import ASReviewProject
     from asreview import ASReviewData
```

20.3 Project API

The ASReview project file is a zipped folder. To unzip the folder and store its contents in a temporary directory, use the following code:

```
[2]: project_path = Path("tmp_data")
     project_path.mkdir()
     project = ASReviewProject.load('example.asreview', project_path)
```

The returned `project` instance is of type ASReviewProject.

To inspect the project details, use the following code:

```
[3]: project.config
```

```
[3]: {'version': '1.0rc1+46.gb310417.dirty',
      'id': 'example',
      'mode': 'simulate',
      'name': 'example',
      'description': 'Simulation created via ASReview via command line interface',
      'authors': None,
      'created_at_unix': 1654172382,
      'datetimeCreated': '2022-06-02 14:19:42.822517',
      'reviews': [{'id': '23a4a370cba1433db3842b20a504ec82',
        'start_time': '2022-06-02 14:19:43.214958',
        'status': 'finished',
        'end_time': '2022-06-02 14:19:51.587138'}],
      'feature_matrices': [{'id': 'tfidf', 'filename': 'tfidf_feature_matrix.npz'}],
      'dataset_path': 'van_de_Schoot_2017.csv'}
```

The imported dataset is located at `/tmp_data/{project_name}/data/{dataset_filename}`, and can be inspected using the following code:

```
[4]: dataset_fp = Path(project_path, project.config['id'], "data", project.config['dataset_
     →path'])
     dataset = ASReviewData.from_file(dataset_fp)
     dataset.to_dataframe().head()
```

```
[4]:                                                        title  \
     record_id
     0                 Manual for ASEBA School-Age Forms & Profiles
     1            Queensland Trauma Registry: A summary of paedi...
     2            Posttraumatic Stress Disorder: Scientific and ...
     3                             SOCIAL CLASS AND MENTAL ILLNESS
     4            Computerised test generation for cross-nationa...

                                                     abstract keywords  \
     record_id
     0
     1
     2            This comprehensive overview of research and cl...
     3
     4            "'Computerised Test Generation for Cross-Natio...

                                                authors    year  date  \
     record_id
     0                  Achenbach, T. M., Rescorla, L. A.  2001.0  2001
     1                  Dallow, N., Lang, J., Bellamy, N.  2007.0  2007
```

```
2                Ford, J. D., Grasso, D. J., Elhai, J. D., Cour...  2015.0   NaN
3                      Hollingshead, A. B., Redlich, F. C.  1958.0   NaN
4                                        Irvine, S. H.  2014.0   NaN


           doi label_included  label_abstract_screening  duplicate_record_id
record_id
0          NaN              0                         0                  NaN
1          NaN              0                         0                  NaN
2          NaN              0                         0                  NaN
3          NaN              0                         0                  NaN
4          NaN              0                         0                  NaN
```

To obtain the content of the feature matrix, for example, the first row of the matrix, use the following code (note the matrix is in a sparse matrix format):

```python
[5]: feature_extraction_id = project.feature_matrices[0]['id']
     feature_matrix = project.get_feature_matrix(feature_extraction_id)
     print(feature_matrix[0])
```

```
  (0, 20452)    0.35937211648312967
  (0, 18297)    0.26158369118434677
  (0, 13842)    0.3248271421716685
  (0, 9739)     0.38355660008860293
  (0, 3231)     0.7059309068495663
  (0, 2384)     0.22684547910949254
```

20.4 State API

The data stored during the review process can be accessed as a pandas DataFrame using the following code:

```python
[6]: with open_state('example.asreview') as state:
         df = state.get_dataset()
```

The returned `state` instance is of type SQLiteState.

```python
[7]: df.to_csv('example_state.csv', index=False)
     df.head()
```

```
[7]:    record_id  label classifier query_strategy balance_strategy  \
    0       2455      1       None          prior             None
    1       2677      0       None          prior             None
    2       2859      0         nb            max           double
    3       3406      0         nb            max           double
    4       1366      0         nb            max           double

      feature_extraction  training_set                 labeling_time notes
    0               None            -1  2022-06-02 14:19:45.069368  None
    1               None            -1  2022-06-02 14:19:45.069368  None
    2              tfidf             2  2022-06-02 14:19:45.163866  None
    3              tfidf             3  2022-06-02 14:19:45.179505  None
    4              tfidf             4  2022-06-02 14:19:45.195130  None
```
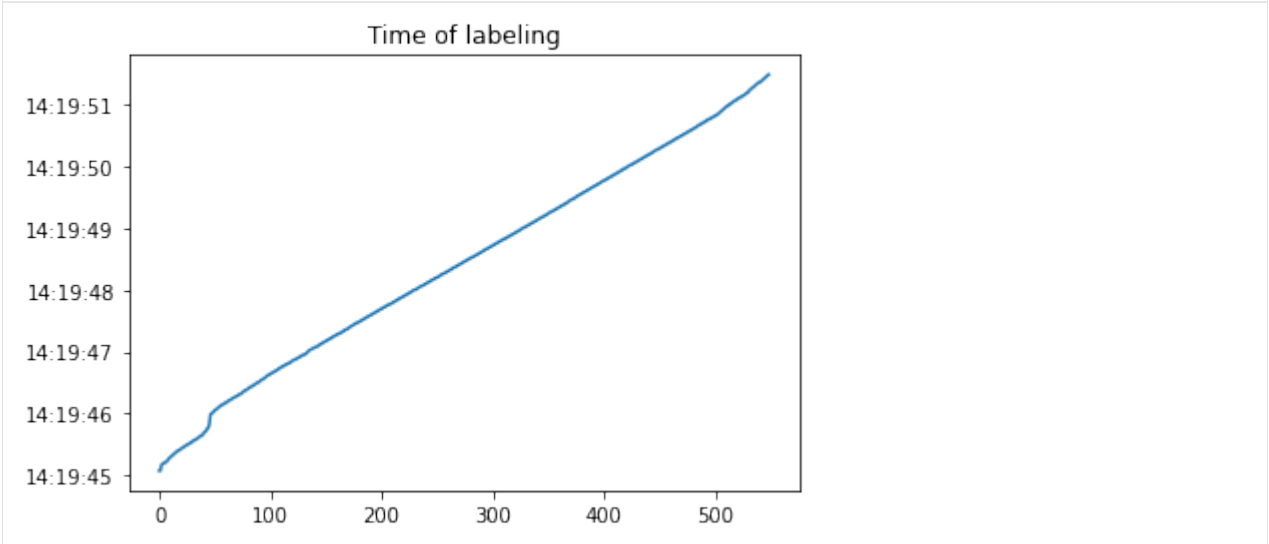
There are also multiple functions to get one specific type of data directly. For example, to obtain the labeling times and plot them in a graph, use the following code:

```
[8]: with open_state('example.asreview') as state:
         labeling_times = state.get_labeling_times()
     pd.to_datetime(labeling_times).plot(title="Time of labeling")
```

```
[8]: <AxesSubplot:title={'center':'Time of labeling'}>
```


By default, the records that are part of the prior knowledge are included in the results. To obtain the labels use the following code:

```
[9]: with open_state('example.asreview') as state:
         labels = state.get_labels(priors=False)
     labels
```

```
[9]: 0      0
     1      0
     2      0
     3      0
     4      0
           ..
     542    0
     543    0
     544    0
     545    0
     546    1
     Name: label, Length: 547, dtype: int64
```

To obtain the data corresponding to a specific record identifier, use the following code:

```
[10]: with open_state('example.asreview') as state:
          record_data = state.get_data_by_record_id(5176)
      record_data
```

```
[10]:    record_id  label classifier query_strategy balance_strategy  \
      0       5176      0         nb            max           double

         feature_extraction  training_set              labeling_time notes
      0               tfidf             5  2022-06-02 14:19:45.195130  None
```

20.5 Cleanup

The following code removes the temporary folder that was created:

```
[11]: shutil.rmtree(project_path)
```

Create extensions

ASReview extensions enable you to integrate your programs with the ASReview framework seamlessly, by using the Python API. These extensions fall into three different categories, and interact with the API in different ways.

1. *Model extensions*
2. *Subcommand extensions*
3. *Dataset extensions*

The extensibility of the framework is provided by the entrypoints of setuptools. You will need to create a package and install it (for example with pip).

Did you develop a useful extension to ASReview and want to list it on *List of extensions for ASReview LAB*? Create a Pull Request or open an issue on GitHub.

For more information on the ASReview API for creating an extension, a technical reference for development is found under the *API reference*. This technical reference contains functions for use in your extension, and an overview of all classes to extend on.

21.1 Model Extensions

An extension of a `asreview.models.base.BaseModel` type class.

Model extensions extent the ASReview software with new classifiers, query strategies, balance strategies, or feature extraction techniques. These extensions extend one of the model base classes (`asreview.models.balance.base`, `asreview.models.classifiers.base`, `asreview.models.feature_extraction.base`, `asreview.models.query.base`).

The easiest way to extend ASReview with a model is by using the . Create a copy of the template and add the new algorithm to a new model file. It is advised to use the following structure of the package:

```
├── README.md
├── asreviewcontrib
│   └── models
```
(continues on next page)

```
│       ├── classifiers
│       │   ├── __init__.py
│       │   └── example_model.py
│       ├── feature_extraction
│       │   ├── __init__.py
│       │   └── example_feature_extraction.py
│       ├── balance
│       │   ├── __init__.py
│       │   └── example_balance_strategies.py
│       └── query
│           ├── __init__.py
│           └── example_query_strategies.py
├── setup.py
└── tests
```
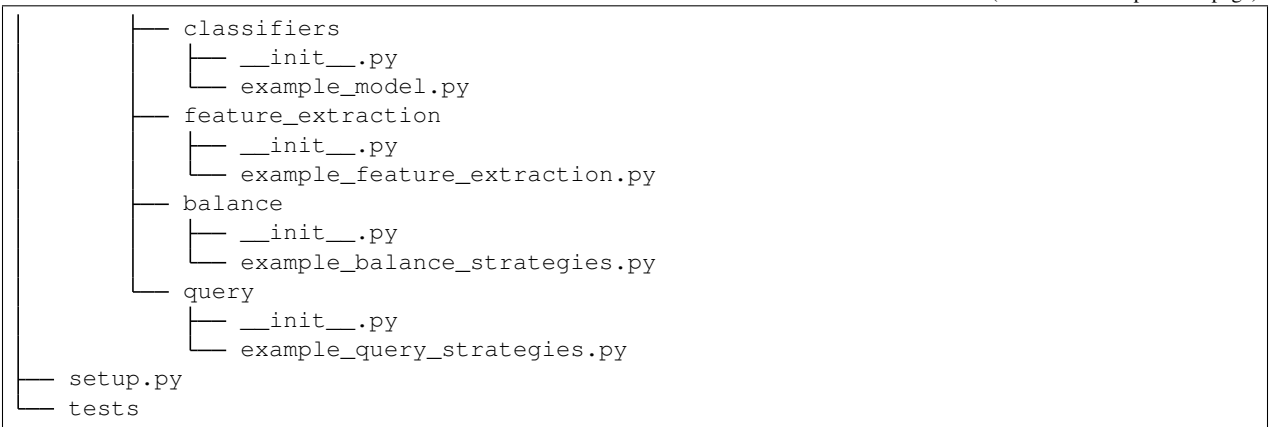
The next step is to add metadata to the setup.py file. Edit the `name` of the package and point the `entry_points` to the models.

```
entry_points={
    'asreview.models.classifiers': [
        'example = asreviewcontrib.models.classifiers.example_model:ExampleClassifier
↪',
    ],
    'asreview.models.feature_extraction': [
        # define feature_extraction algorithms
    ],
    'asreview.models.balance': [
        # define balance_strategy algorithms
    ],
    'asreview.models.query': [
        # define query_strategy algorithms
    ]
},
```

This code registers the model with name `example`.

21.2 Subcommand Extensions

An extension of the `asreview.entry_points.base.BaseEntryPoint` class.

Subcommand extensions are programs that create a new entry point for ASReview. From this entry point the Python API can be used in many ways (like `plot` or `simulate`).

Extensions in ASReview are Python packages and can extend the subcommands of asreview (see `asreview -h`).

An example of a subcommand extension is the Visualization Extension

The easiest way to create a new subcommand is by defining a class that can be used as a new entry point for ASReview. This class should inherit from `asreview.entry_points.base.BaseEntryPoint`. Add the functionality to the class method `execute`.

```
from asreview.entry_points import BaseEntryPoint

class ExampleEntryPoint(BaseEntryPoint):
```

```
    description = "Description of example extension"
    extension_name = "asreview-example"  # Name of the extension
    version = "1.0"  # Version of the extension in x.y(.z) format.

    def execute(self, argv):
        pass  # Implement your functionality here.
```

It is strongly recommended to define the attributes `description`, `extension_name`, and `version`.

The class method `execute` accepts a positional arugument (`argv` in this example). First create the functionality you would like to be able to use in any directory. The argument `argv` are the command line arguments left after removing asreview and the entry point.

It is advised to place the newly defined class `ExampleEntryPoints` in the following package structure: `asreviewcontrib.{extension_name}.{your_modules}`. For example:

```
├── README.md
├── asreviewcontrib
│   └── example
│       ├── __init__.py
│       ├── entrypoint.py
│       └── example_utils.py
├── setup.py
└── tests
```

Create a `setup.py` in the root of the package, and set the keyword argument *entry_points* of `setup()` under `asreview.entry_points`, for example:

```
entry_points={
    "asreview.entry_points": [
        "example = asreviewcontrib.example.entrypoint:ExampleEntryPoint",
    ]
}
```

After installing this package, ASReview is extended with the `asreview example` subcommand. See `asreview -h` for this option.

21.3 Dataset Extensions

An extension of the `asreview.datasets.BaseDataSet` class.

Dataset extensions integrate new datasets for use in ASReview. Adding datasets via extension provides quick access to the dataset via Command Line Interface or in ASReview LAB.

It is advised to place the new dataset `your_dataset` in the following package structure:

```
├── README.md
├── asreviewcontrib
│   └── dataset_name
│       ├── __init__.py
│       └── your_dataset.py
├── data
│   └── your_dataset.csv
├── setup.py
└── tests
```

For minimal functionality, `your_dataset.py` should extent `asreview.datasets.BaseDataSet` and `asreview.datasets.BaseDataGroup`.

A working template to clone and use can be found at Template for extending ASReview with a new dataset.

Further functionality can be extensions of any other class in `asreview.datasets`.

API Reference

22.1 Data and datasets

22.1.1 Read data

ASReviewData
load_data

22.1.2 Statistics

data.statistics.abstract_length
data.statistics.n_duplicates
data.statistics.n_irrelevant
data.statistics.n_keywords
data.statistics.n_missing_abstract
data.statistics.n_missing_title
data.statistics.n_records
data.statistics.n_relevant
data.statistics.n_unlabeled
data.statistics.title_length

22.1.3 Datasets

Available datasets

asreview.datasets.BenchmarkDataGroup

Table 3 – continued from previous page
asreview.datasets. NaturePublicationDataGroup

Dataset managers

asreview.datasets.BaseDataSet
asreview.datasets.BaseDataGroup
asreview.datasets.DatasetManager

22.2 Reviewer

review.BaseReview
review.ReviewSimulate

22.3 Models

Base class

models.base.BaseModel

22.3.1 `asreview.models.balance`

Classes

balance.base.BaseBalance
balance.SimpleBalance
balance.DoubleBalance
balance.TripleBalance
balance.UndersampleBalance

Functions

balance.get_balance_model
balance.get_balance_class
balance.list_balance_strategies

22.3.2 `asreview.models.classifiers`

Classes

classifiers.base.BaseTrainClassifier
classifiers.NaiveBayesClassifier
classifiers.RandomForestClassifier
classifiers.SVMClassifier

Table 9 – continued from previous page

classifiers.LogisticClassifier
classifiers.LSTMBaseClassifier
classifiers.LSTMPoolClassifier
classifiers.NN2LayerClassifier

Functions

classifiers.get_classifier
classifiers.get_classifier_class
classifiers.list_classifiers

22.3.3 `asreview.models.feature_extraction`

Classes

feature_extraction.base. BaseFeatureExtraction
feature_extraction.Tfidf
feature_extraction.Doc2Vec
feature_extraction.EmbeddingIdf
feature_extraction.EmbeddingLSTM
feature_extraction.SBERT

Functions

feature_extraction.get_feature_model
feature_extraction.get_feature_class
feature_extraction. list_feature_extraction

22.3.4 `asreview.models.query`

Classes

query.base.BaseQueryStrategy
query.base.ProbaQueryStrategy
query.MaxQuery
query.MixedQuery
query.MaxRandomQuery
query.MaxUncertaintyQuery
query.UncertaintyQuery
query.RandomQuery
query.ClusterQuery

Functions

query.get_query_model
query.get_query_class
query.list_query_strategies

22.4 Projects and States

Load, interact, and extract information from project files and states (the "diary" of the review).

——

22.4.1 ASReviewProject

ASReviewProject

22.4.2 State

open_state
state.SQLiteState

22.4.3 Utils

project.get_project_path
project.project_from_id
project.list_asreview_projects
project.is_project
project.is_v0_project

22.5 Readers and writers

This module contains the input and output functionality. You can install them as extensions.

asreview.list_readers
asreview.list_writers

io.CSVReader
io.CSVWriter
io.ExcelReader
io.ExcelWriter
io.PaperRecord
io.RISReader
io.RISWriter
io.TSVWriter

22.6 Misc

Classes

`asreview.settings.ASReviewSettings`

Functions

`asreview.search.fuzzy_find`
`asreview.asreview_path`
`asreview.get_data_home`

22.7 Entry points

Entry points for ASReview LAB.

`entry_points.BaseEntryPoint`
`entry_points.AlgorithmsEntryPoint`
`entry_points.LABEntryPoint`
`entry_points.SimulateEntryPoint`
`entry_points.StateInspectEntryPoint`

Indices and tables

- genindex
- modindex

Index

Symbols

-certfile CERTFILE_FULL_PATH
    asreview-lab command line option, 15,
        77
-clean-all-projects CLEAN_ALL_PROJECTS
    asreview-lab command line option, 16,
        77
-clean-project CLEAN_PROJECT
    asreview-lab command line option, 16,
        77
-config_file CONFIG_FILE
    asreview-simulate command line
        option, 78
-embedding EMBEDDING_FP
    asreview-lab command line option, 16,
        77
    asreview-simulate command line
        option, 78
-init_seed INIT_SEED
    asreview-simulate command line
        option, 79
-ip IP
    asreview-lab command line option, 15,
        77
-keyfile KEYFILE_FULL_PATH
    asreview-lab command line option, 16,
        77
-n_instances N_INSTANCES
    asreview-simulate command line
        option, 79
-n_prior_excluded N_PRIOR_EXCLUDED
    asreview-simulate command line
        option, 78
-n_prior_included N_PRIOR_INCLUDED
    asreview-simulate command line
        option, 78
-no-browser NO_BROWSER
    asreview-lab command line option, 15,
        77

-port PORT
    asreview-lab command line option, 15,
        77
-port-retries NUMBER_RETRIES
    asreview-lab command line option, 15,
        77
-prior_idx [PRIOR_IDX [PRIOR_IDX ...]]
    asreview-simulate command line
        option, 78
-prior_record_id [PRIOR_RECORD_ID
        [PRIOR_RECORD_ID ...]]
    asreview-simulate command line
        option, 79
-seed SEED
    asreview-lab command line option, 16,
        77
    asreview-simulate command line
        option, 78
-state_file STATE_FILE, -s STATE_FILE
    asreview-simulate command line
        option, 79
-stop_if STOP_IF
    asreview-simulate command line
        option, 79
-verbose VERBOSE, -v VERBOSE
    asreview-simulate command line
        option, 79
-b, -balance_strategy BALANCE_STRATEGY
    asreview-simulate command line
        option, 78
-e, -feature_extraction
        FEATURE_EXTRACTION
    asreview-simulate command line
        option, 78
-h, -help
    asreview-lab command line option, 15,
        78
    asreview-simulate command line
        option, 79
-m, -model MODEL