QUD Anno Challenge
Annotation Tool QUDA
- Installation and Usage Guide -

Christoph Hesse, Maurice Langner, Arndt Riester

October 2022

1 Introduction

This document covers three areas of interest to users of the annotation tool QUDA: (i) How to install and run the tool under various operating systems, (ii) the set of annotation labels available in the tool, and (iii) the file format of QUD trees annotated with QUDA.

2 Install and run QUDA

Some general points before we get into the operating system specifics: QUDA runs in a web browser. We recommend Mozilla Firefox (there are known issues with other browsers, e.g., Opera or Safari) to ensure full feature support. In order to work in the browser, you need to run a local server. For this you need to have Python 3 (or later) installed, which you can download for free, e.g., at the Apple or Microsoft App Store.

It also means that every time you want to use QUDA you need to start this server using the shell script `run_QUDA_on_[your operating system].sh`. After you are done using QUDA (make sure you have saved your annotations), close Firefox and remember to also terminate the server. All files necessary to install QUDA (`install_QUDA_on_[your operating system].sh`) and to run the server (`run_QUDA_on_[your operating system].sh`) are contained in the Github repository. Download the Github repository `https://github.com/MMLangner/QUDA` and unpack it. Both the install and run scripts need to be executed from the console of your operating system (i.e., a way of interacting with you system without a graphical user interface). This console goes by different names in the different operating systems: in Windows it is called the *command line*, in MacOS *Terminal*, and in Linux the *console*. The operating-system-specific quirks of working with the console are explained in the following sections.

3 Quick guide

1. Make sure you have the latest version of Mozilla Firefox and Python 3 installed.
2. Download `https://github.com/MMLangner/QUDA` by clicking on the file `QUDA_local_<version>.zip` and *Download*.
3. Unpack the ZIP file. You should now have a `QUDA-local_<version>` folder with a subfolder `qudviewer`, which contains the installation and run scripts for your operating system.
4. Execute the installation script for your operating system contained in `QUDA_local_<version>`.

5. Execute the run script for your operating system contained in `QUDA_local_<version>`.

6. Open Firefox and enter the address `http://127.0.0.1:8000/apps/viewer/` to open QUDA. (In Windows, Firefox might be started automatically.)

4 Windows

Install

1. Make sure you have Python 3 installed. If not, please download it from `https://www.python.org/downloads/windows/` or the Microsoft Store (Version > 3.8) and install it.
2. Use the file menu to get to the decompressed git repository. Go to the subfolder containing `install_QUDA_on_WIN.bat`, which should be `QUDA_local_<version>\QUDA_local_<version>` or just `QUDA_local_<version>`.
3. Run the shell script `install_QUDA_on_WIN.bat` by double-clicking it. This should open a cmd prompt, that gives you information on the installation status. In case Python is not installed, it will echo "Python is not installed on your system." Otherwise, if installation succeeds, the prompt will respond with "Requirements installed. Please use QUDA start file to start app."

Run

1. Go to the subfolder containing `run_QUDA_on_WIN.bat`, which should be `QUDA_local_<version>\QUDA_local_<version>` or just `QUDA_local_<version>`.
2. Run the shell script `run_QUDA_on_WIN.bat` by double-clicking it. This should open a new cmd prompt, which tells you that the small local server was started and the app is accessible via the localhost (127.0.0.1:8000/). Please open Firefox and navigate to `http://127.0.0.1:8000/apps/viewer` to access QUDA.
3. If you want to stop QUDA, save your data, close Firefox, move to the cmd prompt window and click CTRL + c to quit the QUDA app. If the cmd prompt does not close automatically, you can close it by pressing the x-button in the upper corner.

5 MacOS

Install

1. Open the *Terminal* app from the Dashboard (F4 key), subfolder *Other* or use search. In the *Finder*, it is located under `Applications/Utilities`.
2. Make sure you have Python 3 installed. You can check by typing `python3 --version` in the Terminal.
3. In Finder, go to the unpacked git repository. Go to the subfolder containing `install_QUDA_on_MacOS.sh`, which should be `QUDA-local_<version>`.
4. Run the shell script `install_QUDA_on_MacOS.sh`, by drag-and-dropping it from the Finder window to the Terminal window: type `sh` in Terminal, then drag-and-drop `install_QUDA_on_MacOS.sh` over. You should see the script's full file path. Hit enter to run the script.

Run

1. Run `run_QUDA_on_MacOS.sh` using the same drag-and-drop technique: type `sh` in the Terminal and drag `run_QUDA_on_MacOS.sh` over from the Finder; hit enter. You might get some warnings in the Terminal about unapplied migrations. You can ignore these. The last line in the Terminal

should say `Quit the server with CONTROL-C.` This means the local server needed for QUDA is now running in the background. (Note that you don't see anything until you start your browser.)
2. Open Mozilla Firefox and navigate to `http://127.0.0.1:8000/apps/viewer/` to use QUDA.
3. In order to close QUDA, save your annotations, close Firefox, and in the Terminal, press Control + c. The Terminal then quits the run script `run_QUDA_on_MacOS.sh` and returns to the prompt showing your MacOS username. The server is now terminated and you can close the Terminal app.

6 Linux

Install

1. If pip3 is not installed on your system, use `sudo apt-get install python3-pip` (or use pacman for arch or a different package manager, depending on your distribution).
2. Use pip3 to install the neccessary dependencies, namely **django**, **django_downloadview** and **numpy**, with syntax `pip3 install [...]`.

Run

1. Navigate to the decompressed git repository and its subdirectory containing the manage.py file (`~/qudviewer/manage.py`)
2. Execute the command `python3 manage.py runserver`
3. Open firefox and navigate to `http://127.0.0.1:8000/apps/viewer/`
4. In order to quit the QUDA tool, save your progress, close the browser and terminate the local server with CTRL+c in the terminal window.

7 Annotation labels

There are three levels of annotation: (1) QUDs, (2) discourse units, and (3) segmentation of discourse units (e.g., focus structure). The minimal requirements for QUD trees require discourse units corresponding to the annotated text and QUDs for the annotated QUDs. Segmentation of discourse units is optional or as needed (e.g., for annotating focus structure).

QUDs

QUDA allows to distinguish between implicit and overt QUDs. By default, any newly created QUD is implicit. Use the dropdown menu to change it from `QUD` (for implicit QUDs) to `QUD-OVERT`.

Discourse Units

Similar to QUDs, discourse units, `UNIT` can also be labeled using their dropdown menu as overt or implicit. This can be useful for ellipsis or implicit assertions.

Segmentation

Discourse units can be segmented further using segments' dropdown menu. The following labels are available for segments:

- Newly created segments have the default label `None`. Please assign an actual label.
- `F`: focus segment
- `DT`: discourse topic

- `BG`: background material
- `CT`: contrastive topic
- `T`: sentence topic
- `NAI`: not-at-issue content
- `DM`: discourse marker
- `CMT`: comment constituent
- `OVERT`: can be used to mark material containing overt questions.

8 XML Specification

Annotations made in QUDA are saved in XML format. This section shows an example XML file. If you wish to use another tool for annotation, you are encouraged to ensure that your XML files comply with the XML specification here.

8.1 Example XML file (QUD tree only)

```
<ROOT>

<QUD string="What is the way things are?" classification="QUD">
    <QUD string="What about Peter?" classification="QUD">
        <UNIT>Peter is a famous actor.</UNIT>
        <QUD string="What evidence is there that Peter is famous?" classification="QUD">
            <QUD string="What about his stage plays?" classification="QUD">
                <UNIT>His stage plays always attract large audiences</UNIT>
            </QUD>
            <QUD string="What about his feature films?" classification="QUD">
                <UNIT>and his feature films are also usually big box office hits.</UNIT>
            </QUD>
        </QUD>
        <QUD string="What is the secret of his success?" classification="QUD-OVERT">
            <QUD string="Is it simply his good looks?" classification="QUD-OVERT"/>
            <UNIT>Some say he just connects brilliantly with his audience.</UNIT>
        </QUD>
    </QUD>
</QUD>

</ROOT>
```

8.2 Example XML file with information structure

In case you would like to include information-structural markup, this is an extended version of the same piece of text:

```
<ROOT>

<QUD string="What is the way things are?" classification="QUD">
    <QUD string="What about Peter?" classification="QUD">
        <UNIT>Peter is a famous actor.
            <T>Peter</T>
            <F>is a famous actor</F>
        </UNIT>
        <QUD string="What evidence is there that Peter is famous?" classification="QUD">
            <QUD string="What about his stage plays?" classification="QUD">
                <UNIT>
                His stage plays always attract large audiences
                <CT>His stage plays</CT>
                <F>always attract large audiences</F>
                </UNIT>
            </QUD>
            <QUD string="What about his feature films?" classification="QUD">
                <UNIT>
                and his feature films are also usually big box office hits.
                <DM>and</DM>
                <CT>his feature films</CT>
                <F>are also usually big box office hits</F>
                </UNIT>
            </QUD>
        </QUD>
        <QUD string="What is the secret of his success?" classification="QUD-OVERT">
            <QUD string="Is it simply his good looks?" classification="QUD-OVERT"/>
            <UNIT>
            Some say he just connects brilliantly with his audience.
            <NAI>Some say</NAI>
            <T>he</T>
            <F>just connects brilliantly with his audience</F>
            </UNIT>
        </QUD>
    </QUD>
</QUD>

</ROOT>
```