



User Guide

In order to run the application, there are several requirements to be fulfilled beforehand.

For this purpose, a **requirements.txt** file was generated with the aim of helping set up the user's environment of choice (virtual or not). We recommend using Python3.8.

Some non-environment-related aspects that need to be resolved before running the application successfully are that the user has a compatible CLIPS environment for our application output. Newer versions of the CLIPS IDE (<https://sourceforge.net/projects/clipsrules/files/>) seem to not be compatible with some functions used, so we recommend version 6.24.

After installing the IDE, **copy the path and add it to the file metadata/clips_path.txt**, which holds the path to the IDE so it can be opened from within the application.

After opening the application, if the user is using it for the first time, there is a need to generate input files. This can be done by selecting "**Generate CLIPS Files**" from the main menu, then selecting "**Generate Input Files**".

In the "**Generate Input Files**" window, there is a choice for either uploading a file that contains the text to be analyzed or writing it using the interface. At the end, the destination path has to be added, in order to save the input.dat file.

The user can either save and go back to the main menu (if they needed only to generate a new input file and use the functionalities already generated), or save and continue with generating the actual files.

In case they *Save and Continue*, the next dialog that appears is equivalent to the one from "**Generate Functionalities Files**". In this window, you can choose the

starting grammar. This can either be the *Default* one (given by the text found in metadata/default_grammar.txt) or the user can *Upload* a personal text file, in order to generate a grammar from it. Next, the functionalities have to be selected: *Learning or Updating a Grammar* represents the first functionality and *Parsing the Input Sentences* represents the second functionality. Next, the user has to choose the directory in which to save the given files and also the input.dat file generated previously. After hitting either *Save* or *Save and Continue*, the user gets back to the main menu.

The next step is to “**Open CLIPS IDE**” and load the files as usual.

This way of organizing the application helps users that have already generated files to navigate easier and faster the application, in case they only need the functionality files or the input file.

The final results will be saved in the directory of choice and can be checked using any text editor.