

Installation File

To set up the environment for running TAPHSIR, follow the instructions below:

1. Install Python 3.8. Create a new virtual environment, then activate the environment.

Python Installation

If Python is not installed we recommend easy installation via [Anadonda](#)

Virtual environment setup

Using conda/anaconda, you can create a new environment as follows:

```
> conda create -n myenv python=3.8

> conda activate myenv
```

Using the command line, you can create a new virtual environment as follows:

```
> python -m pip install --upgrade virtualenv

> virtualenv -p python3.8 myenv

> source myenv/venv/bin/activate (or myenv/bin/activate)
```

Note: If you use Microsoft Windows, then you need to replace the command for activation with the following one:

```
> myenv\Scripts\activate.bat
```

2. Install the required libraries using the following

instructions:

- Navigate to the project directory where you cloned TAPHSIR on your local machine using *cd*

```
> cd path/to/taphsir
```

- Run the following commands in the command line:

```
> pip install -r requirements.txt

> python -m spacy download en_core_web_sm
```

The first command installs all the required libraries for using TAPHSIR, and the second one downloads `en_core_web_sm`, which we use to employ the natural language processing pipeline in SpaCy.

3. Prepare your input document:

TAPHSIR expects as input a text file (with the extension *.txt*) containing a set of requirements (or sentences).

It is recommended to place the input document in the project directory. However, you have the choice to specify the path to the input document when you run TAPHSIR as explained in README.

4. Run TAPHSIR:

- Navigate to the project directory using *cd* as follows:

```
> cd path/to/taphsir
```

- Run the tool using the following command:

```
> python taphsir.py --doc path/to/doc --mode <mode> --detection <detection_mo
del>
```

where the parameters:

mode takes a numeric value from the following: 1, 2, or 3, indicating whether you wish to run the detection component only, resolution only, or both. The default value for **mode** is 3.

detection_model takes a string value from the following: **LF** for running the ML model that is pre-trained over language features only, **FE** for running the ML model that is pre-trained over feature embeddings only, or **Ensemble** for running the ensemble ML model that is pre-trained over both sets of features. The default value for **detection_model** is **Ensemble**.

Here are some examples:

- For running the resolution component only using our example document:

```
> python taphsir.py --doc Example.txt --mode 2
```

Note that specifying the parameter **detection_model** in this case is not required and has no effect.

- For running the detection component only by applying the pre-trained ML model over language features using our example document:

```
> python taphsir.py --doc Example.txt --mode 1 --detection LF
```

- For running both components (detection and resolution) by applying the pre-trained ensemble ML model on one of your own documents:

```
> python taphsir.py --doc /path/to/file.txt
```

5. Interpret the results of TAPHSIR:

As explained in README, running TAPHSIR results in two Excel files generated in the output folder **./output**. The two files are *detection.xlsx* and *resolution.xlsx*

The files contain the following columns:

Id is a unique identifier for each pronoun in a predefined context. This Id is used mainly to distinguish multiple pronouns in the same context, e.g., *it-1* is first pronoun *it* in the context and *it-2* is the second pronoun, etc.

Context refers to the context generated by TAPHSIR for a particular pronoun, i.e., the sentence where the pronoun occurs and the previous sentence. Further details about the context can be found in our technical

paper.

Pronoun is the pronoun analyzed by TAPHSIR.

The detection file has a **Detected As** column indicating the result, i.e., whether the pronoun is detected as **Ambiguous** or **Unambiguous**.

The resolution file has a **Resolved As** column indicating the result, i.e., the antecedent to which the pronoun refers.

Note: unresolved pronouns will not be displayed in the output file.

Docker

If you are familiar with docker containers, you can use [TAPHSIR image from the docker hub](#).

First you have to download docker from the [official Docker website](#). Make sure that Docker is installed by running the command `docker -v`

The TAPHSIR image has all the software requirements pre-installed. Once docker is installed, you can run the TAPHSIR tool on some requirements document titled *file.txt* using the following instructions:

1. Pull the TAPHSIR image

```
> docker pull ezzini/taphsir
```

2. Run a container and get its ID (make sure to copy the output CONTAINER_ID for later use)

```
> docker container run -d -t ezzini/taphsir bash
```

3. Copy your input text file inside the container. You can skip this step if you wish to use the example file that we provide

```
> docker cp /path/to/file.txt CONTAINER_ID:file.txt
```

4. Open the container command line (run `ls` to make sure your file is there)

```
> docker exec -it CONTAINER_ID bash

> ls
```

5. Run the artifact and `exit` the container once finished

```
> python taphsir.py --doc file.txt --mode <mode> --detection <model>

> exit
```

6. Export the output folder to your local directory

```
> docker cp CONTAINER_ID:output /path/to/your/local/folder/
```

Using docker to run TAPHSIR with our example document can be done as follows:

```
> docker pull ezzini/taphsir

> docker container run -d -t ezzini/taphsir bash
```

2d25d58d5f6057a845aed09df6a040d326cc8c47cf0de087cb5bed4d58e8c183

```
> docker cp users/saad.ezzini/Desktop/Example.txt 2d25d58d5f6057a845aed09df6a040d326cc8c47cf0de087cb5bed4d58e8c183:Example.txt

> docker exec -it 2d25d58d5f6057a845aed09df6a040d326cc8c47cf0de087cb5bed4d58e8c183 bash

> python taphsir.py --doc Example.txt --mode 1 --detection LF

> exit

> docker cp 2d25d58d5f6057a845aed09df6a040d326cc8c47cf0de087cb5bed4d58e8c183:
```

output users/saad.ezzini/Desktop/