Ariusproject manual

Presequites:
We worked with Ubuntu16.04. Everything what is needed is included in install script, so there is no problems with preparing your system for the project.

It should be compatible with other debian distributives, but we didn't test it with other than Linux Mint and Ubuntu.

Installation:

In the root directory of the repository you should run install.sh.
It is automatic, but still needs your attention to confirm installation.

For install QT we use original setup. For Arius you can just skip all the steps, because standard options without registration are good enough for the project.

One of the steps of the installation is downloading Mary TTS voices. You can pick whatever you need, but our default configuring needs EN-us cmu-slt-hsmm voice. Just pick it, install and then quit from the voice downloader.

After installation is finished you should activate virtual environment and download nltk packages for quepy. Just open terminal, write "source env/bin/activate" in the root of the repository and run python.

In the python you should "import nltk" and then use "nltk.download()" to run interface for downloading. You should download "wordnet" and "average_perceptron_tagger" for the work of Arius.

Preparing to the work:

Documents location:
First of all you should locate all information that Arius shows in the folders local_pages and videos. (those folders are located in server/static folder).
The folder "local_pages" is for webpages and pdf documents. The folder "videos" is for webm videos (we highly recommend to transform videos to the webm format as we play it with html5 player).

Right now we use gif images for video template. To change them, add or replaces you should look to the "config.py" and "preview" folder in the "static"

Internet:

It is needed to have connection to the internet for the correct work
Now we use the internet for the voice answers on simple questions. Quepy uses online dbpedia for it. (In the future it can be downloaded and used locally)

Index building:

One of the parts of the processing the question is Elasticsearch. To make it work you should build index of all files. To do this you should run elasticsearch server (it is located in "required_packages/elasticsearch-2.3.5/bin) and then run "esindex_builder.py" in the services folder (you may be needed to activate virtual environment first)

It will make the index for searching for all the needed (look config.py) files in “local_pages” folder.

Tagging:
Except Quepy and Elascticsearch we provide tagging data to show important materials. In the ”server/static” we have two files: “tag_data.json” and “video_data.json”. First is used by core module to generate Arius reaction on the user request. This file contains two important parts: synonyms and tag_data

Tagging new files:

It is the only way to make Arius work with videos, as nothing else in this solution can extract sense from the videos, so it is must to tag video files

To add new tag which can be used for tagging data you should write it key and all words that should be associated with this tag (e.g. **'1':{'key':'Arius', 'equal':['Arius', 'ads project rnd']}**). After adding tag to the "synonyms" you can use it for tagging any type of files, and that files will be showed on the needed request.

Tagging new files:
To add new file with associated tags you need to add it the next format”**'1':{'priority': 0, 'name': Arius, 'path': 'Arius.pdf', 'tags': [['Arius', 1], ['rnd', 0.4], ['showroom', 0.6]}** “

Priority shows importance of the file. It is integer and smaller is it more important the file is.

Name just let us easily identify the file (if the path is not informative)

Path is a relative path (for videos it is path from the folder videos e.g “ **'path': 'arius.webm'** ”if the full path to the file is “ **home/.../ariusproject/server/static/videos/arius.webm** ”.

For other files it is relative path from the folder 'local_pages' where is suggested to save all documents and webpages)

Tags field contains list of lists where we have key of the tag and its relevance to the file. For example if file is about Arius and briefly about the showroom then confidence of the "Arius" tag is bigger than confidence of the "showroom" tag. This number should be float in between 0 and 1

Adding video to the “video_data.json”:

To add new video you also should add information for the server. It is also located in the “static” folder.

There are few fields for description and information which is needed by flask rendering (we show all the video in one template).

It is a must for keys and video_name field to be the relative path to the video from the folder “videos” in the “static”. Also we have only “default.css” file so for all the videos is needed to fill “style” field with “default” for correct rendering.

Running:

As we create virtual environment with installation script and install all needed packages in it, you should at first activate it with “source env/bin/activate” in the root of the project.

Now you are able to run Arius. In the root project we have run.sh script which run all the parts of the Arius.

You can run Arius in three different ways. We use tmux to make debugging easy and comfortable.

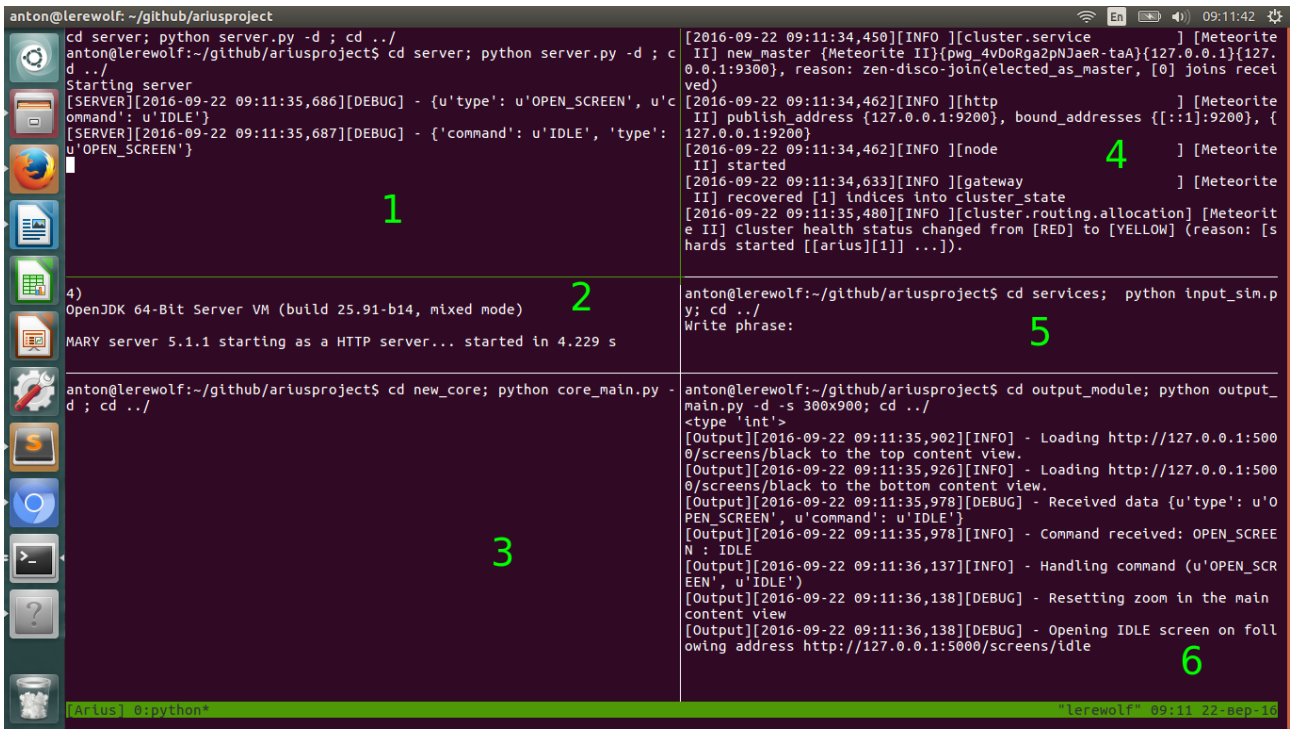
- Debug mode with all information just run “./run.sh -m debug”
- Less informative verbose mode “./run.sh -m verbose”
- Silent work (without visible terminals at all) “./run.sh -m silent”

To make your oLutput size specific use flag “-s widthxheight” (e.g. “./run.sh -m debug -s 500x600”, “./run.sh -m silent -s 100x400”, ...). Otherwise browser will be in the fullscreen mode.

To finish work you need to write “exit” in the “input” terminal.

Tmux terminals layout:
To navigate through terminals you can use **Ctrl**+**B** and **Arrows**

1 – Server terminal
2 – Mary TTS server terminals
3 – Core terminal
4 – Elasticsearch server terminal
5 – Input terminal
6 – Output terminal



Configuration:

All important parts which are configurable are described in the “config.py” file

In the file 'config.py' which is located in the root of the project you can modify addresses, style, voice, default phrases to speak, phrases associated with commands and templates.

It is required to change flask addresses for both clients and servers (e.g. "flask_server_video_addr" and "flask_server_video_addr_client"). Otherwise if won't work correctly.

To get know how to change voice you can attend "127.0.0.1:59125/documentation.html" after starting marytts server (to start it just run “./required_packages/marytts-5.1.1/bin/marytts-server” from the project root) It is good to try voice first with web-interface "127.0.0.1:59125". It is easy to configure and test.