

Aim: This document aims to provide run instructions for feature 3 extraction for Milestone 2 only. This doesn't provide a holistic run document for all the features.

Github branch link: <https://github.com/surajsrivathsa/fiction/tree/feature3>

Assumptions: We have some of the below assumptions regarding users system

- 1) User has python(version 3.6 and above) installed, can set up virtual environments on their own and install libraries from requirements.txt in the environment
- 2) User has java(1.8 and above) installed along with IDE and git integration.
- 3) User already has the consolidated feature file generated at least by extracting features 1-21(Old features at least). However, Users can also provide a consolidated list after extracting new features too(features 1 and 2), In that case, one of the parameter_values should be changed in config.properties.
- 4) Users can run the java file which in turn runs a shell script/batch script which has wrapped the python script for feature3.
Java Driver(extract_feature3 function) ---> shell script/batch script ---> Python Driver.
In case One cannot run from java, separate instructions are given below to directly run from the python file.
- 5) If you are on Linux or Windows the Python paths and conda paths should be set either in Linux bash_profile or in the windows environment variables

We have already extracted feature3 and placed it in the file. Also, the code is ready to be run for both English and German. Just that we need to have the books along with their languages in Final_Booklist excel file. **Features_extracted_english_with_features2_3.csv**

Step 1: Setup local repository from a remote GitHub branch feature3(link provided above). Import and build the local project as a Maven project.

Step 2: Create a conda or any other python virtual environment and install libraries as provided in requirements.txt file in linux, In windows the environment thing is not working for now, Hence all the libraries should be installed from the main python environment according to requirements.txt.

Also, download the spacy models into your local environment by executing the below commands. You can download other models too if you wish according to the [website from spacy](#)

```
python -m spacy download en_core_web_sm  
python -m spacy download de_core_news_sm
```

Note: Please note we faced some issues during creating environments and installing packages. Hence in case of such errors Please do a pip install the libraries as per below and download the language models as per above

pip install numpy pandas matplotlib spacy nltk

Step 3: Go to config.properties in Java and do below changes to variables mentioned below

#Python environment name ---> Give your created python environment name. If everything is installed in the base environment then give the environment name as base. Please note that for Windows we are facing some problems with activating environment, Hence this environment variable won't be used in windows scripts. We would be running directly from main python environment for now, in Windows. For Linux/Mac the environment is working fine, Hence go ahead and set the environment where the required dependent libraries are installed

python.environment.name = "nlp_env"

#The consolidated feature file path that consists of extracted features from books. Change it to your feature full file path

feature_file_path = /Users/surajshashidhar/git/fiction/Features_Extracted_English.csv

#The path where extracted html books are there. If you do not have extracted epub then just uncomment the epub-->html function in java driver and start running as usual.

book_file_path = /Users/surajshashidhar/git/fiction/Short_epubs_extracted/

#Full file path of emoticons that has both English and German language emotions

emoticon_file_path =

"/Users/surajshashidhar/Desktop/ovgu/semester_2/XAI_project/researched_code_and_data/all_language_emotions.csv"

#Full file path of the book list(This list was uploaded on the first milestone). Please note that filename can be of any name, but the sheet name of the excel file should be "Final_Booklist", else the program may fail. If the sheet name is to be changed, then it can be changed in constants file in python but not from constants file in Java.

book_list_file_path = "/Users/surajshashidhar/git/fiction/Final_Booklist.xlsx"

#Full file path of new feature file which would have older extracted features with new feature 3. It can be set to a new file or be given the old feature file itself so that it would be overwritten.

new_feature_file_path =

/Users/surajshashidhar/git/fiction/Short_epubs_extracted/new_Features_Extracted.csv"

#Full file path of python driver code

python_code_file_path =

"/Users/surajshashidhar/git/fiction/extract_emotion_features/extract_emotions_driver.py"

#Application constants

#the column number of the feature3 in the feature file. For example, if the old feature file already has features F1, F2....F21 fields, then feature_fields should be set to 22 as feature3 would be populated in F22. If i have extracted features F1 - F23 then i should provide 24 here.

feature_fields = 24

#Dummy column value, can be set to anything or ignored as is. This would be removed in the next milestone.

language = "en"

#Encoding should be utf-8, no changes required here

encoding = "utf-8"

#Hyper parameter to dynamically take book start and bookend based on percentage. If it is 0.2 then we would consider 20% of book start and 20% of bookending for creating feature vectors and calculating similarity. Any value between 0.15-0.25 seems okay.

book_start_percentage = 0.2

book_end_percentage = 0.2

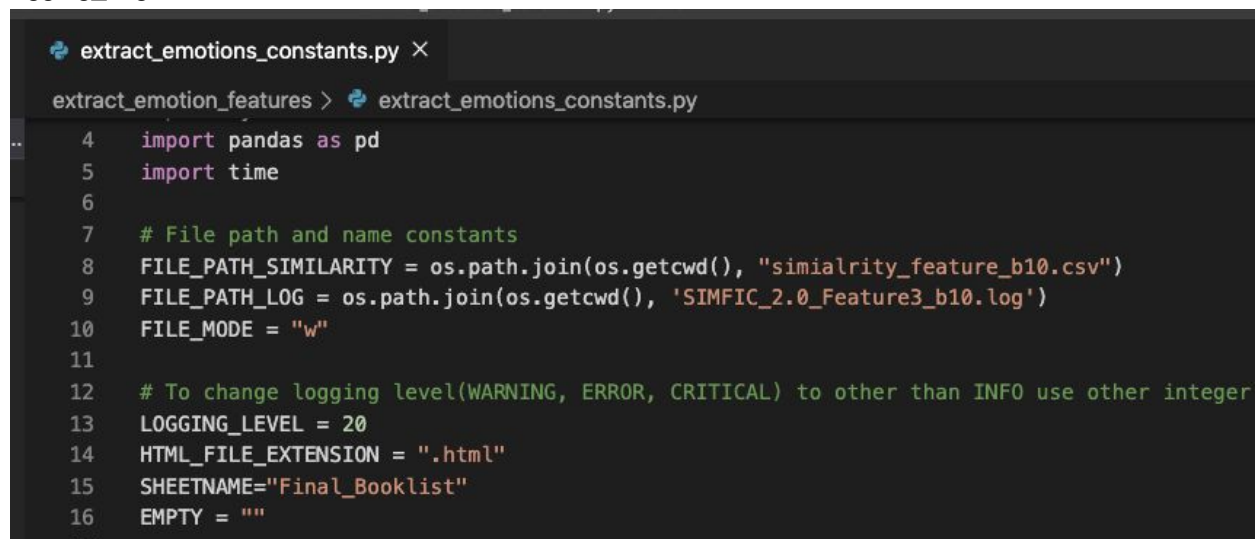
#Type of similarity to be used. We have cosine similarity also available accessible by parameter value "cosine", but L2 worked a bit better than cosine, hence set it to L2

similarity_type = "L2"

Is logging required. We have implemented logging in python where a separate log file would be created for feature 3, if set to true then there would be a bit of slowness in execution due to logging.

To change the log file name, You must do it from python constants file as it is not available in Java constants file or properties file.

logging_flag = "True"



```
extract_emotions_constants.py X
extract_emotion_features > extract_emotions_constants.py
4 import pandas as pd
5 import time
6
7 # File path and name constants
8 FILE_PATH_SIMILARITY = os.path.join(os.getcwd(), "similarity_feature_b10.csv")
9 FILE_PATH_LOG = os.path.join(os.getcwd(), 'SIMFIC_2.0_Feature3_b10.log')
10 FILE_MODE = "w"
11
12 # To change logging level(WARNING, ERROR, CRITICAL) to other than INFO use other integer
13 LOGGING_LEVEL = 20
14 HTML_FILE_EXTENSION = ".html"
15 SHEETNAME="Final_Booklist"
16 EMPTY = ""
```

Below are system scripts that runs python programs to extract feature 3. We just pass the parameters from java ---> shell/batch ---> python

Change it to flag is_windows to Yes for running on Windows and No in Linux

for Linux give the shell script and script type, For windows not required.

script.name = /Users/surajshashidhar/git/fiction/run_python_jobs.sh

script.type = sh

is_windows = Yes

Step 4: Once the properties are set, make sure that all the lines inside the Java Driver file are commented out except the line which calls `extract_feature3()` else the entire pipeline would be run again and all the features F1-F21 would be re extracted.

Step 5: Right-click on Java driver and click on Run as ---> Java application. Wait for some time to see the output on the console. From our experiments, it took 2.5 hours for around 1700 books with percentage considered as 20% and on a 6GB RAM machine.

Alternative steps in case programs don't run from Java. Running Python programs directly.

Step 6: Open `extract_emotions_driver.py` and `extract_emotions_constants.py` file in `extract_emotion_features` folder inside `fiction` folder and navigate until the end of the file to find below piece of text related to **argument parser**. Locate the same parameters mentioned in step 3 and do the changes for **default** value of argument parser mentioned in step 3. Run the python driver file.

```
ap.add_argument( "--feature_file_path", nargs= "?", required=False, help="
feature_file_path", default =
"/Users/surajshashidhar/git/fiction/Features_Extracted_English.csv")
    ap.add_argument("--book_file_path", nargs= "?", required=False,
help="book_file_path", default =
"/Users/surajshashidhar/git/fiction/Batch10_extracted")
    ap.add_argument( "--emoticon_file_path", nargs= "?", required=False, help="
emoticon_file_path",
default="/Users/surajshashidhar/Desktop/ovgu/semester_2/XAI_project/reasearched_code_a
nd_data/all_language_emotions.csv")
    ap.add_argument("--feature_fields", nargs= "?", required=False,
help="feature_fields", default = constants.FEATURE_FIELD)
    ap.add_argument( "--language", nargs= "?", required=False, help=" language",
default = constants.ENGLISH)
    ap.add_argument("--encoding", nargs= "?", required=False, help="encoding", default
= "utf-8")
    ap.add_argument( "--book_start_percentage", nargs= "?", required=False, help="
book_start_percentage", default = constants.DEFAULT_BOOK_START_PERCENTAGE)
    ap.add_argument("--book_end_percentage", nargs= "?", required=False,
help="book_end_percentage", default = constants.DEFAULT_BOOK_END_PERCENTAGE)
    ap.add_argument( "--similarity_type", nargs= "?", required=False, help="
similarity_type", default = constants.L2)
```

```
    ap.add_argument("--new_feature_file_path", nargs= "?", required=False,
help="new_feature_file_path", default =
"/Users/surajshashidhar/git/fiction/Features_Extracted_English_b10.csv")
    ap.add_argument("--book_list_file_path", nargs= "?", required=False,
help="book_list_file_path", default =
"/Users/surajshashidhar/git/fiction/Final_Booklist.xlsx")
    ap.add_argument("--logging_flag", nargs= "?", required=False, help="logging_flag",
default = "True")
```