**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 python file.

1. 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 file **Features\_extracted\_english\_with\_features2\_3.csv**

**Step 1:** Setup local repository from a remote GitHub branch feature3(link provided above)

**Step 2:** Create a conda or any other python virtual environment and install libraries as provided in requirements.txt file.

**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

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 epubs 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/reasearched\_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

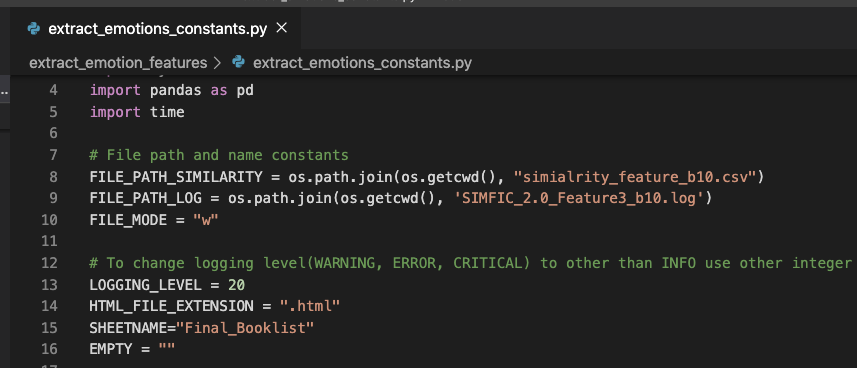
#Type of similarity to be used. We have cosine similarity also available accessible by parameter value “cosine”, but L2 worked a bit better than cosine, hence set it to L2

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"



# 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 batch script when running on windows system

# for windows script.name = "run\_feature3\_batch.bat" , script.type = "bat"

script.name = /Users/surajshashidhar/git/fiction/run\_python\_jobs.sh

script.type = sh

**Step 4:** Once the properties are set, make sure that all the lines inside 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 reextracted.

**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 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 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 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\_and\_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")