Project Name: Phone Banking Phase 2

Business Unit:	Phone Banking
Process Name:	Phone Banking Phase 2
Document Type:	Solution Design Document
Version No:	V1.1
Authors:	Arunav Sahay
Date Created:	12 August 2022
Last Amended:	25 September 2022
Project Phase:	Design
Prepared By:	ICICI – Wipro RPA Team

Introduction

Phone Banking team daily receives hundreds of emails from different customers querying about different products, issues, and services. The phone banking team has to manually look into every email and forward it to the desired team or take appropriate action. It is a tedious task to go extract entities from the emails and look for what customer is querying about.

An RPA-AI based solution was henceforth designed to reduce the manual efforts and automate the entire process. An RPA was designed to fetch emails one by one and apply various business logic to identify the email components, product, nature, and entity of the email. This entire automation process was developed in Phase 1, where the bot was able to fetch the customer email and predict the product and intent of the email.

Phase 2 of Phone Banking project was developed to predict sub-intent of the product and intent and extract major entities. There are around 88 sub-intents mapped in phase 2 of the project and 6 entities. The development was done to create an AI model which can classify the sub-intent of the email and extract the correct entity.

General Overview:

The Phase 2 AI model development was done in Python 3.6 language which is compatible with UIPATH Enterprise. This is a Natural Language Processing based project in which customer email is the Natural Language considering English language has been used throughout. Two different NLP models were developed (i) To predict the sub-intent of the email categories and (ii) To identify and extract the entity.

The software used to train and use these models were written in Python 3.6 scripting language. With usage of advanced data science libraries all the development was made possible.

Multiple traditional machine learning algorithms were studied and used in the project for sub-intent classification. Deep Neural Networks were used for extraction of entities.

Scikit-learn library was used for email classification problem. Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

spaCy library was used for Named Entity Extraction. spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. spaCy supports deep learning workflows that allow connecting statistical models trained by popular machine learning libraries like TensorFlow, PyTorch or MXNet. Prebuilt statistical neural network models to perform these tasks are available for 17 languages, including English, Portuguese, Spanish, Russian and Chinese, and there is also a multi-language NER model. Additional support for tokenization for more than 65 languages allows users to train custom models on their own datasets as well.

pandas library was used in both the use cases. A lot of exploratory data analysis and data modelling was done using pandas. pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

Numpy library was used with pandas and other mathematical calculations. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

smote-variants library was used to generate synthetic data for classes having data less than the minimum count to improve the accuracy.

In summary, the existing system used the following libraries/components:

Python 3.6
Numpy
Pandas
Scikit-learn
spaCy
Smote-Variants

Proposed Solution - Statement of Need

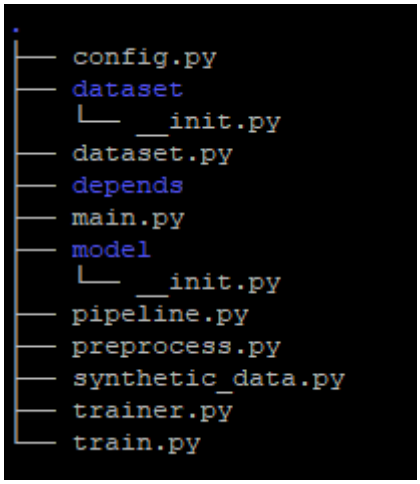
The existing solution classifies the product and entity of the email. The proposed solution for phase 2 classifies the sub-intent after predicting the product and intent. The output from Phase 1 model is passed to the Phase 2 Email Classification API, which appends to the email body and subject and predicts the sub-intent. The same output is passed to Entity Extraction Model API and the entity is predicted with filtered data as decided by business.

Methodology:

Sub-Intent Classification

The Email Classification model takes Email Body, Subject, predicted Product and predicted Intent as input from the UIPATH Bot and processes for sub-intent prediction. But before coming to prediction we will discuss the procedure to make the model and the scripts.

Folder Architecture:



Files:

1. **config.py**. We define certain variables which may impact training of the NLP model.
2. **dataset.py**. Dataset selection and initial dataframe is built from this code.
3. **main.py**. This is the main file which must be present with the same name as it is required in UIPATH AI-FABRIC. For prediction the UIPATH API is hit which calls the *Main.predict()* function passing the email string as an input.
4. **pipeline.py**. All the model definitions, hyper-parameters and text vectorization method is defined in this file.
5. **preprocess.py**. Before training the model or making a prediction, the *preprocess_text()* function is called which performs text processing to remove unwanted data & simplify text data.
6. **synthetic_data.py**. We generate synthetic data from this code. Function *generate_synthetic_text()* generates the synthetic data for classes having data count lower than the threshold set in variable *syn_data_count* in file trainer.py
7. **trainer.py** is the training script which calls all the necessary functions which is used for training. From fetching dataframe from dataset.py script to pre-processing text, synthetic data generation, splitting dataset into train and test sets, model training and evaluation. It is the backbone script for model training.

8. **train.py**. Just like main.py script, it is a necessary file for training model in UIPATH-AIFABRIC. It calls the trainer.py script and saves the trained model.

Folders:

1. **dataset**, houses the csv file which will be used for training. The file uploaded on UIPATH AI-FABRIC is stored in this folder.
2. **depends**, contains all the python libraries which is imported by these scripts.
3. **model**, stores the output model file and the encoder in *.pkl* format, along with the test and train report generated after model training.

Model Folder Contents:

```
17M     classifier_model.pkl
4.0K    encoder_model.pkl
4.0K    test_report_01c7b694a30bb6f29df698e40be6d815.txt
4.0K    test_report_29bcb46a3e17bc98fb97e67cbef2fd1d.txt
4.0K    test_report_7ae9f1d228817d3e90d96262a88043b0.txt
4.0K    train_report_01c7b694a30bb6f29df698e40be6d815.txt
4.0K    train_report_29bcb46a3e17bc98fb97e67cbef2fd1d.txt
4.0K    train_report_7ae9f1d228817d3e90d96262a88043b0.txt
```

classifier_model.pkl is the trained model of Sub-intent. The size of the model may vary of different systems. This screenshot was taken on Ubuntu 18.04.6 LTS system.

encoder_model.pkl file has the class information encoded using joblib library.

test_report_*.txt** file has the following contents

```
0.8242030331166822
              precision    recall  f1-score   support

           0       0.76      0.69      0.72       180
           1       0.84      0.71      0.77        69
           2       0.69      0.81      0.75       155
           3       0.89      0.89      0.89       201
           4       0.97      0.98      0.98       180
           5       0.82      0.85      0.84       180
           6       0.90      0.86      0.88       221
           7       0.74      0.82      0.78       157
           8       0.94      0.92      0.93       169
           9       0.91      0.90      0.90       213
          10       0.79      0.84      0.82       176
          11       0.97      0.95      0.96       177
          12       0.69      0.75      0.72       174
          13       0.58      0.50      0.54       200
          14       0.80      0.80      0.80       178
          15       0.66      0.71      0.69       185
          16       0.59      0.66      0.62       174
```

The first line tells us the overall training accuracy i.e., 0.8242 times 100, which is 82.42%.

The following lines tells us the precision, recall, f1-score and support of each class. Class values are the first column content (0,1,2,3…15,16).

```
              precision    recall  f1-score   support

    accuracy                       0.82      9693
   macro avg       0.83      0.83      0.83      9693
weighted avg       0.83      0.82      0.82      9693
```

The last 3 lines gives us the overall averages accuracy for precision, recall, f1-score and total support (total dataset used for testing). The

train_report_*.txt**, just like *test_report_***.txt* file has the training accuracy and class wise information.

```
0.9891155760749013
              precision    recall  f1-score   support

           0       0.99      0.98      0.98       716
           1       0.99      1.00      0.99       251
           2       0.97      0.99      0.98       630
           3       0.99      0.98      0.99       799
           4       1.00      1.00      1.00       716
           5       0.98      1.00      0.99       716
           6       1.00      0.98      0.99       779
           7       0.99      0.98      0.99       665
           8       1.00      1.00      1.00       727
           9       1.00      0.99      0.99       785
          10       0.98      0.99      0.98       722
          11       1.00      1.00      1.00       642
          12       0.99      0.99      0.99       721
          13       0.99      0.98      0.99       699
```

```
              precision    recall  f1-score   support

    accuracy                       0.99     38771
   macro avg       0.99      0.99      0.99     38771
weighted avg       0.99      0.99      0.99     38771
```

Dataset Overview:

Subcategory Name and the sample count

Imobile_Imobile Technical Issue SR	900
Imobile_Imobile Activation Issue-Probe Handset Details	900
Imobile_Pin Regeneration	900
Statement - CC_Credit Card E-Statement Not Received Request	900
Statement - CC_Billing Cycle Change Enquiry	900
EMI-Debit Card_CANCELLATION AND REFUND RELATED	900
Charges Related - CC_Enquiry On Fuel Surcharge Charges For Credit Card	900
Charges Related - CC_Reversals Processed LPC/Int Overlimit And Other Charges	900
Charges Related - CC_Credit Card Charges – Others	900
Charges Related - CC_Enquiry Of Annual Joining Fee For Credit Card	900
Charges Related - CC_Enquiry On Credit Card Interest Charges	900
Charges Related - CC_Enquiry On Late Payment Charges For Credit Card	900

Charges Related - CC_Enquiry On Autodebit Charges For Credit Card	900
EMI/Mechant EMI_EMI Cancellation Done	900
EMI/Mechant EMI_EMI Conversion Done	900
Credit Card Payment Status_Payment Modes Of Credit Card TAT	900
Credit Card Payment Status_Payment Not Credited	900
Credit Card Payment Status_Auto Debit Status	900
Account Closure_SB account closure request not done	900
Account Closure_Account closure procedure	900
Account Closure_Account closure reason probing	900
Account Closure_Account closure procedure if customer is abroad	900
Internet Banking login - RL_Know USER ID/Activation of USER ID	900
Credit Card Status_Active Status	900
Credit Card Status_Intransit To Re-Direct To Branch Card Delivered At The Branch	900
Credit Card Status_Redispatch Of The Card	900
Credit Card Status_Apin Generation Process	900
Credit Card Status_Courier Dispatch Status	900
Credit Card Status_Replacement Of Card L/S/E/E1/E2	900
Customer details updation - CC_Email ID change Process	900
Customer details updation - CC_Name Mother Maiden Name change Process	900
Customer details updation - CC_Address Change Process	900
Customer details updation - CC_PAN Updation	900
Customer details updation - CC_Mobile change Process	900
UPI/IMPS/NEFT/RTGS_Funds Transfer Status	877
UPI/IMPS/NEFT/RTGS_Upi Imps-Chargeback SR	807
UPI/IMPS/NEFT/RTGS_Wrong Fund Transfer	729
Account Modification_Change Of Communication Address	701
Account Modification_Dormant Account Activation	588
Account Modification_Inactive Account Activation	563
Transaction Status_Transaction Refunded By The Merchant	504
Transaction Status_Transaction Status Declined And Status-CC	499
Transaction Status_Transaction Settled Late	451
Charges Related - RL_Reversal Of ECS Return Charges	430
Charges Related - RL_Mab Savings Account General Inquiry	419
Charges Related - RL_ECS Return Charges Reason For ESC Return	397
Charges Related - RL_Savings Account Atm Pos Decline Charges	382
Charges Related - RL_Other Saving Account Charges	367
Charges Related - RL_Debit Card Annual Fee	364
Charges Related - RL_Cash Transaction Charges Savings Account	333
Charges Related - RL_Reversal Of Debit Card Annual Fee	326
Debit Card - Cheque Deliverables_Cheque Book Dispatch Status	323
Debit Card - Cheque Deliverables_Re Dispatch Of Debit Card Cheque Book	315
Debit Card - Cheque Deliverables_Debit Card Cheque Book Not Received	229
Debit Card - Cheque Deliverables_Priority Delivery	180
Debit Card - Cheque Deliverables_Alternate Channels To Apply For A Debit Card Cheque Book	114

Debit Card - Cheque Deliverables_Re-Issue Or Lost All Channels Request - NRI	108

Training Dataset:

The subject and email Body is appended together to form a new column **“text”.** The subcategory is added in column **“label”.**

text	label
Liabilities Imobile 000WBaGNB2EHUA0U:Level1Complaint / Mobile banking Name: Shalini Manda Ganesh Account No / Application No: 086601518465 Product/Service: Mobile banking Request related to: iMobile Type of request/complaint: iMobile related issues E-mail address: shaliniganesh07@gmail.com Mobile no:9677873793 Telephone no: +91-96-77873793 Level 2 Service Request(SR) No: Complaint/Feedback: Unable to open the imobile app message displayed in the screen we are currently experiencing difficulty in processing your request please try again after some time	Imobile_Imobile Technical Issue SR
Liabilities Imobile 000WBaGNB2EHUBDK:Gold line don't renvel bu imobile aap External Email Warning: Do not click on any attachment or links/URL in this email unless sender is reliable. Dear sir I am not renewal my gold lone by imobile application my registra mobile number is 7073257766 and my lone number is 379305000436 please chek	Imobile_Imobile Technical Issue SR
Liabilities Imobile 000WBaGNB2EHUUHK: External Email Warning: Do not click on any attachment or links/URL in this email unless sender is reliable. Account No:024301531408 Mobile No:8125913079 Can I know How to create upi ID in imobile app for icici bank	Imobile_Imobile Technical Issue SR

Entity Extraction

It takes an email body as an input and passes to the model; it gives list of entities found by model in dictionary format.

Input & Output:
INPUT: Datatype: **String** --> 'subject + emailBody'
OUTPUT: Datatype: **json** --> {'**class1'**:'ent1, ent2', '**class2'**:'ent1'…}

Model Training Process:
Dataset Annotations (Manual/Auto Annotations /Data Augmentation):
File type: **csv**
Column's structure: 1st column – email bodies (Statements), Datatype: str
Next columns – class names, Datatype: list of entities

	A	B	C	D	E	F
1	Statements	Date	Amount	MobileNo	ModeOfPayment	DeliverableType

Statements	Date	Amount	MobileNo	ModeOfPayment	DeliverableType
Name: Riyas Ahamed N Account No / Application No: 4748466858374004 Product/Service: Credit Card Request related to: Others Type of request/complaint: Other issues E-mail address: n.riyasnavas@gmail.com Mobile no:9941405191 Telephone no: +91-- Level 2 Service Request(SR) No: Complaint/Feedback: Dear Icici Team, I am writing this to you for consideration in waiving the charges made on my credit card(4748466858374004) for being late in paying the amount of Rs. 3,038.91. I am going to pay the total outstanding amount on or before my due date of this month. Please revise the extra charges(Rs.3038.91) on my credit card. I am ready to pay the amount tomorrow (30th July22). Please do the needful. I shall be ever grateful if I get an affirmative response for the waiving of bank charges on my credit card. Thanks. Regards, Riyas Ahamed N	["30th July22"]	["Rs. 3,038.91","Rs.3038.91"]	["9941405191"]		

Training Data (dataset.py):

After creation of dataset, that excel is formatted into below format.

[(email_text, {'entities': [start_index, end_index, class_name]}), (), (),….]

email_text: String, Email subject & body which has entity to be extracted.
entities: Keyword(mandatory)
start_index: Index of start character of entity from the email_text.
end_index: Index of end character of entity from the email_text.
class_name: String, entity class name i.e which class it belongs to.

The above formatting is done using script **"dataset.py"**

Entities Name:

Currently deployed Classes are as follows (**Case sensitive**):

Date, Amount, MobileNo, ModeOfPayment, DeliverableType, SRNumber

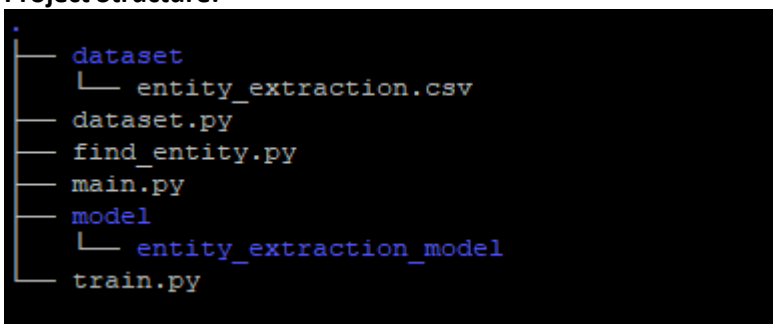
Node: Suggested to keep same class names (including case) in annotated dataset (csv file) as these class names are given as a output to RPA process. **SO PLS DO NOT CHANGE CLASS NAMES IN Dataset File.**

<u>Extraction / Formatting Using Other Methods:</u>
DATA FORMATTING:

	Entity Extraction - Date Formats		
Input	**Output**	**Input description**	**Output description**
01-01-2022	01-01-2022	If date, month & year is mentioned properly	dd-mm-yyyy

Jan 2022/ Jan month	01-01-2022 to 31-01-2022	If only month or mm-yyyy without date	start date of month to end date of month
2021- 2022	01-04-2021 to 31-03-2022	If only year range	start of financial year to end of financial year
2 months	19-05-2022 to 19-07-2022	If duration in months	last 2 months from current date
Last year	19-07-2021 to 19-07-2022	If duration in years	last 1 year from current date
2022		If only year is mentioned without dd-mm	Discard
Note : If output is in dd-mm-yyyy then key label - **"Date"** If output is in range then key label - **"DateRange"**			

Project Structure:



dataset.py - Loading excel file & Creating training data i.e Converting excel to training data
train.py - Created dataset & trains model & stores model in **/model** folder
main.py – Extract entities using model & regex pattern(spacy) & **find_entity** module
find_entity.py – Extracting amount using regex & Cleaning card number + account number + year from date

Classes Prediction:

Date: Results are extracted from model (class date) then formats the date into standard date format i.e **dd-mm-yyyy**
DateRange: Results are extracted from model (class date) then formats the date into standard date format i.e **dd-mm-yyyy to dd-mm-yyyy**
Amount: Results are extracted from model (class Amount) + **find_entity.py** (After cleaning & regex patterns) then formats to only numbers & decimal (rs,inr,k,lacs,etc)
MobileNo: Results are extracted from model (class MobileNo) + added patterns(regex) in spacy then removing mobileno -> length of mobileno < 10 & > 12
ModeOfPayment: Results are extracted from model (class ModeOfPayment)
DeliverableType: Results are extracted from model (class DeliverableType)
SRNumber: Results are extracted by patterns(regex) in spacy

Limitations With Phase 2 Models:

Currently in Sub-Intent Email Classification model, the confidence of the prediction is very vague. It is possible that with correct prediction the accuracy might be very low and with wrong prediction the accuracy might be high. It is to be assumed that the predictions with high accuracy or accuracies above the threshold agreed by the business team are correct and can be used as STP (Straight Through Processing) cases.

With Entity Extraction the following entities have limitations,

Date extraction: Wrong/No prediction when there is no space between date, month & year. E.g.: 12thJune 2020/ 12 jun2022.
Wrong prediction - when 2-digit year is given without date e.g.: jun-22.
Missing out some dates when multiple dates are present in the statement.

Mobile extraction: Wrong predictions: Sometimes extracts account no / credit card number.

Entity Extraction training has been done with more than 20000 data samples. And training this model on UIPATH AI FABRIC took more than 36 hours. With regular annotations, changes in data samples regular training can be time expensive.

No information output – When an email sample doesn't have any information i.e., the body of the email is empty and the subject only has the thread ID, the model gives out any of the sub-intents. This model or the script should be returning a void Flag or default output indicating the predicted output if NULL.

Scope of Improvement:

1. The accuracy of the Sub-Intent model can be worked upon to give a realist score, where correct predictions can have a better score and the wrong predictions can have a lower score.
2. Training time for entity extraction model can be reduced to a few hours. This can be done either by reducing the number of samples for training or improving the server by using GPUs instead of CPUs.
3. Number of improvements in entity extraction training data sample. As the greater number of samples gets trained the accuracy of entity extraction will improve. We need to reduce the training deployments for the same.
4. Merging phase 1 and phase 3 cases with phase 2.
5. Exploration of Deep Neural Network techniques for Email Classification.