Agent_classification_agent-data-classification

September 9, 2021

[53]:
```
import numpy as np
import matplotlib.pyplot as plt
import pickle
import random
import pandas
import tensorflow as tf


from tensorflow.keras.layers import LSTM, Activation, Dense, Dropout, Input,␣
↪BatchNormalization, Embedding
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.models import load_model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import LearningRateScheduler

from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split

import os
import string
import re

os.environ["TF_FORCE_GPU_ALLOW_GROWTH"]="true"
```

[2]:
```
os.path.sep.join(os.path.__file__.split(os.path.sep)[0:-1])
```

[2]:
```
'/Users/cc.lee/opt/miniconda3/envs/tensorflow/lib/python3.8'
```

0.0.1 Define Constants

[90]:
```
EMBEDDING_FILE_PATH = os.path.sep.join(["..", "embedding", "glove.6B.50d.txt"])
MODEL_OUTPUT_PATH = "./output/2021-06-02/saved_model"
MODEL_OUTPUT_LABEL_BINAIZER_PATH = "./output/2021-06-02/labelBinarizer.pickle"
```

```
MODEL_OUTPUT_AGENT_CHAR_WORD_INDEX = "./output/2021-06-02/lagentWordIndex.
 ↪pickle"
MODEL_OUTPUT_KERAS_MODEL = "./output/2021-06-02/lagent-analysis-keras-model.
 ↪pickle"
COMBINED_DATA="./combined_data/csv_files/agent_original_combined.csv"

classes_dir = os.path.sep.join(["processed_data"])
INIT_LR=1e-3

MAX_VOCAB_SIZE = 10000
MAX_LENGTH = 9
# treat each character as a value of a sequence
MAX_SEQ_LENGTH = 40
```

[63]:
```
classes_dir
```

[63]: 'processed_data'

[64]:
```
classes = []

for dir_path, _, _ in os.walk(classes_dir):
    if dir_path != "processed_data":
        classes.append(dir_path.split(os.path.sep)[-1])
```

testtesttest

0.0.2 Prepare our Training data as a Python Array Object

[65]:
```
contents = []
labels = []

def remove_punctuation(paragraph):
    for punc in string.punctuation:
        paragraph = paragraph.replace(punc,"")
    return paragraph

def preprocess_data(folder_path):
    for i, (dir_path, dir_names, file_names) in enumerate(os.walk(folder_path)):
        if dir_path != "processed_data":
#             print(f"{len(file_names)} files in {dir_path} have been loaded")
            for file_name in file_names:
                file_path = os.path.sep.join([dir_path, file_name])
                category = file_path.split(os.path.sep)[-2]


                with open(file_path, "r", encoding="ISO-8859-1") as io:
#                     content = f.read().strip()
                    content = io.readlines()
```

```
                    content = [remove_punctuation(c.strip()) for c in content]
                    content = [re.sub(r"(\n)+", " ", c) for c in content]
                    content = [c.lower() for c in content]
                    for item in content:
                        if len(item) < MAX_SEQ_LENGTH:
                            contents.append(item)
                            labels.append(category)

preprocess_data(classes_dir)
```

[66]:

```
print(labels)
```

```
['suspEndDate', 'suspEndDate', '_id', '_id', '_id', '_id', '_id', '_id', '_id',
'_id', '_id', '_id', 'tel', 'tel', 'tel', 'tel', 'tel', 'tel', 'tel', 'tel',
'proxyEndDate', 'proxyEndDate', 'proxyEndDate', 'proxyEndDate', 'proxyEndDate',
'proxyEndDate', 'proxyEndDate', 'proxyEndDate', 'proxyEndDate', 'proxyEndDate',
'proxyEndDate', 'proxyEndDate', 'proxyEndDate', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode', 'agentCode',
'agentCode', 'agentCode', 'agentCode', 'userRole', 'userRole', 'userRole',
'userRole', 'role', 'acceptTermVersion', 'proxyStartDate', 'proxyStartDate',
'proxyStartDate', 'proxyStartDate', 'proxyStartDate', 'proxyStartDate',
'proxyStartDate', 'proxyStartDate', 'proxyStartDate', 'proxyStartDate',
'proxyStartDate', 'proxyStartDate', 'proxyStartDate', 'proxy1UserId',
'proxy1UserId', 'proxy1UserId', 'proxy1UserId', 'proxy1UserId', 'proxy1UserId',
'proxy1UserId', 'proxy1UserId', 'proxy1UserId', 'proxy1UserId', 'proxy1UserId',
'proxy1UserId', 'proxy1UserId', 'proxy1UserId', 'mobilePhone', 'mobilePhone',
'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone',
'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone',
'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone',
'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone', 'mobilePhone',
```

'mobilePhone', 'achievements', 'achievements', 'achievements', 'achievements',
'achievements', 'title', 'authorised', 'authorised', 'authorised', 'authorised',
'authorised', 'authorised', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode',
'unitCode', 'unitCode', 'unitCode', 'unitCode', 'unitCode', 'position',
'position', 'position', 'position', 'position', 'position', 'position',
'position', 'position', 'position', 'position', 'position', 'position',
'position', 'position', 'position', 'position', 'position', '_rev', '_rev',
'_rev', '_rev', '_rev', '_rev', '_rev', '_rev', '_rev', '_rev', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date',
'date', 'date', 'date', 'date', 'date', 'date', 'date', 'date', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',

```
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId', 'userId',
'userId', 'userId', 'userId', 'userId', 'userId', 'authorizedHI',
'authorizedHI', 'officePhone', 'officePhone', 'officePhone', 'officePhone',
'officePhone', 'authGroup', 'authGroup', 'authGroup', 'authGroup', 'authGroup',
'authGroup', 'authGroup', 'authGroup', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'fullName',
'fullName', 'fullName', 'fullName', 'fullName', 'fullName', 'userStatus',
'userStatus', 'proxy2UserId', 'proxy2UserId', 'proxy2UserId', 'proxy2UserId',
'proxy2UserId', 'userCat', 'userCat', 'userCat', 'userCat', 'userCat',
'userCat', 'userCat', 'userCat', 'userCat', 'userCat', 'userCat', 'userCat',
'userCat', 'userCat', 'userCat', 'createDate', 'createDate', 'compCode',
'channel', 'channel', 'channel', 'channel', 'manager', 'manager', 'manager',
'manager', 'manager', 'manager', 'manager', 'TYPE', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'directorOrFirm', 'directorOrFirm', 'directorOrFirm', 'directorOrFirm',
'managerCode', 'managerCode', 'managerCode', 'managerCode', 'managerCode',
'managerCode', 'managerCode', 'managerCode', 'mobile', 'mobile', 'mobile',
'mobile', 'mobile', 'mobile', 'mobile', 'mobile', 'authorizedIFAST',
```

```
'authorizedIFAST', 'userType', 'userType', 'userType', 'userType', 'userType',
'authorizedIAGST', 'faAdvisorRole', 'faAdvisorRole', 'faAdvisorRole', 'id',
'id', 'id', 'id', 'id', 'id', 'id', 'id', 'id', 'id', 'suspStartDate',
'suspStartDate', 'authorizedM8A', 'authorizedM8A', 'lstChgDate', 'lstChgDate',
'lstChgDate', 'patchDataSync', 'name', 'name', 'name', 'name', 'name', 'name',
'name', 'name', 'old_userId', 'old_userId', 'old_userId', 'old_userId',
'old_userId', 'old_userId', 'old_userId', 'old_userId', 'old_userId',
'old_userId', 'old_userId', 'old_userId', 'acceptDate', 'acceptDate',
'profileId', 'profileId', 'profileId', 'profileId', 'profileId', 'profileId',
'profileId', 'profileId', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank',
'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank',
'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'rank', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'distribCode', 'distribCode', 'distribCode',
'distribCode', 'distribCode', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code', 'upline2Code',
'upline2Code', 'AGAdvisorRole', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email', 'email',
```

```
'email', 'email', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code', 'upline1Code',
'upline1Code', 'company', 'company', 'company', 'company', 'lastUpdateDate',
'lastUpdateDate', 'authorizedM8', 'authorizedM8']
```

[67]:

```
NUM_CLASSES=len(set(labels))
```

[68]:

```
agent_data_char_vocab = set()

for content in contents:
    agent_data_char_vocab.update(tuple(content))

agent_data_char_vocab
```

[68]:

```
{' ',
 '0',
 '1',
 '2',
 '3',
 '4',
 '5',
 '6',
 '7',
 '8',
 '9',
 'a',
 'b',
 'c',
 'd',
 'e',
 'f',
 'g',
 'h',
 'i',
 'j',
 'k',
 'l',
 'm',
```

```
 'n',
 'o',
 'p',
 'q',
 'r',
 's',
 't',
 'u',
 'v',
 'w',
 'x',
 'y',
 'z'}
```

0.0.3 Understand a bit more About our Dataset

[69]:
```
print(f"we have total of {len(contents)} training data")

nums=np.array([len(content.split()) for content in contents])

max_num_of_words = np.max(nums)
min_num_of_words = np.min(nums)
total_num_of_words = np.sum(nums)
average_num_of_words = total_num_of_words//len(contents)

for threshold in np.arange(0, 50, 10):
    print(f"{len([num for num in nums if num < threshold])} of paragraph has␣
 ↪number of words less than {threshold}")

print(f"max number of words: {max_num_of_words}")
print(f"min number of words: {min_num_of_words}")
print(f"number of words: {total_num_of_words}")
print(f"average number of words: {average_num_of_words}")
```

```
we have total of 1286 training data
0 of paragraph has number of words less than 0
1286 of paragraph has number of words less than 10
1286 of paragraph has number of words less than 20
1286 of paragraph has number of words less than 30
1286 of paragraph has number of words less than 40
max number of words: 8
min number of words: 1
number of words: 2044
average number of words: 1
```

[70]:
```
# self-made tokenization

agent_data_char_vocab.update(('<pad>','<unk>'))
```

```
agent_data_char_vocab_index_word = dict(enumerate(agent_data_char_vocab))
```

[71]:
```
print(agent_data_char_vocab_index_word)
```

```
{0: 'n', 1: '7', 2: 'b', 3: 'j', 4: 'a', 5: 's', 6: 't', 7: 'z', 8: 'f', 9: 'i',
10: '2', 11: 'r', 12: 'u', 13: '0', 14: 'd', 15: '3', 16: 'w', 17: 'q', 18: ' ',
19: '9', 20: '1', 21: '5', 22: '<unk>', 23: 'p', 24: 'm', 25: '4', 26: 'k', 27:
'8', 28: 'g', 29: '<pad>', 30: '6', 31: 'o', 32: 'v', 33: 'l', 34: 'h', 35: 'x',
36: 'c', 37: 'e', 38: 'y'}
```

[72]:
```
# human_vocab = tokenizer.word_index
agent_data_char_vocab_word_index = { v:i for i,v in␣
 ↪agent_data_char_vocab_index_word.items()  }

print(agent_data_char_vocab_word_index)
```

```
{'n': 0, '7': 1, 'b': 2, 'j': 3, 'a': 4, 's': 5, 't': 6, 'z': 7, 'f': 8, 'i': 9,
'2': 10, 'r': 11, 'u': 12, '0': 13, 'd': 14, '3': 15, 'w': 16, 'q': 17, ' ': 18,
'9': 19, '1': 20, '5': 21, '<unk>': 22, 'p': 23, 'm': 24, '4': 25, 'k': 26, '8':
27, 'g': 28, '<pad>': 29, '6': 30, 'o': 31, 'v': 32, 'l': 33, 'h': 34, 'x': 35,
'c': 36, 'e': 37, 'y': 38}
```

[73]:
```
def string_to_onehot(string, max_seq_length, word_index):
    # in a seq-to-seq model batch of one-hot vectors is the expected output of␣
 ↪the final softmax layer
    # vocab play the role as tokenizer.word_index, i.e., word to index
    # in the past I work on tokenizing words, this time we tokenizer every␣
 ↪single characters
    string = string.lower()
    arr = []
    while len(arr) < len(string):
        curr_index = len(arr)
        arr.append(word_index.get(string[curr_index], word_index['<unk>']))

    while len(arr) < max_seq_length:
        arr.append(word_index['<pad>'])

    onehot = np.zeros((max_seq_length, len(word_index)))
    for i in range(max_seq_length):
        onehot[i, arr[i]] = 1

    return onehot, arr

def output_to_date(out, vocab):
    # this time the "vocab" is index_word
    arr = np.argmax(out,axis=-1)
    string = ''
    for i in arr:
```

```
        string += vocab[i]

    return string
```

[74]:
```
X = []

for content in contents:
    X.append(string_to_onehot(
        content,
        MAX_SEQ_LENGTH,
        agent_data_char_vocab_word_index)[0]
    )
```

0.0.4 Split our dataset into training ones and testing/validation ones.

The validation dataset is used to test whether or not our prediction model can "generalize" to data that the model has never seen. It can happen that our trained model performs very well on training dataset but works poorly to new data. This phenomenon is called **over-fitting**.

[75]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, labels, test_size=0.2)

X_train = np.array(X_train)
X_test = np.array(X_test)
Y_train = np.array(Y_train)
Y_test = np.array(Y_test)
```

[76]:
```
labelBinarizer = LabelBinarizer()
labelBinarizer.fit_transform(labels)
Y_train = labelBinarizer.transform(Y_train)
Y_test = labelBinarizer.transform(Y_test)
```

[77]:
```
print(Y_train[0].shape)
```

```
(56,)
```

0.0.5 Example of Transformed Training data that will be fed into LSTM Model

[218]:
```
X_train[0]
```

[218]:
```
array([[0., 0., 0., …, 0., 0., 0.],
       [0., 0., 1., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       …,
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.]])
```

0.0.6 Define Model

Based on the size of training data and after numerical experiement, we finally come up with the following structure. In case when we have more and more data in the future, we need to adjust the retrain the model in order to accept more information.

[78]:
```
def build_model():
    inputs = Input(name='inputs',␣
 ↪shape=(MAX_SEQ_LENGTH,len(agent_data_char_vocab_word_index)))
    x = LSTM(128)(inputs)
    x = Dense(256,name='FC1')(x)
    x = Activation('tanh')(x)
    x = Dense(128,name='FC2')(x)
    x = Activation('relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    x = Dense(NUM_CLASSES, name='out_layer')(x)
    x = Activation('softmax')(x)
    model = Model(inputs=inputs, outputs=x)
    return model

model = build_model()
```

0.0.7 Check the Shape of data Before Training

[79]:
```
print(X_train.shape)
print(Y_train.shape)
print(X_test.shape)
print(Y_test.shape)
```

```
(1028, 40, 39)
(1028, 56)
(258, 40, 39)
(258, 56)
```

0.0.8 Compile the model

[80]:
```
model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(lr=INIT_LR),
    metrics=['acc']
)
```

0.0.9 Define learning rate scheduler

[81]:
```
def poly_decay(epoch):
    if epoch < 30:
        return INIT_LR
    elif epoch in range(30, 60):
        return INIT_LR * 0.1
```

[82]:
```
lr_scheduler = tf.keras.callbacks.LearningRateScheduler(poly_decay)
```

0.0.10 Train the model

[83]:
```
%matplotlib inline
import matplotlib.pyplot as plt

history = model.fit(
    X_train,
    Y_train,
    validation_data = (X_test, Y_test),
    batch_size=64,
    epochs=60,
    callbacks=[lr_scheduler]
)


epoch_list = history.epoch

plt.plot(epoch_list, history.history['loss'], label='Train Loss')
plt.plot(epoch_list, history.history['val_loss'], label='Validation Loss')

plt.ylabel('Loss')
plt.xlabel('Epoch');plt.title('Loss')

plt.legend(loc="best")
plt.grid(color='gray', linestyle='-', linewidth=0.5)
```

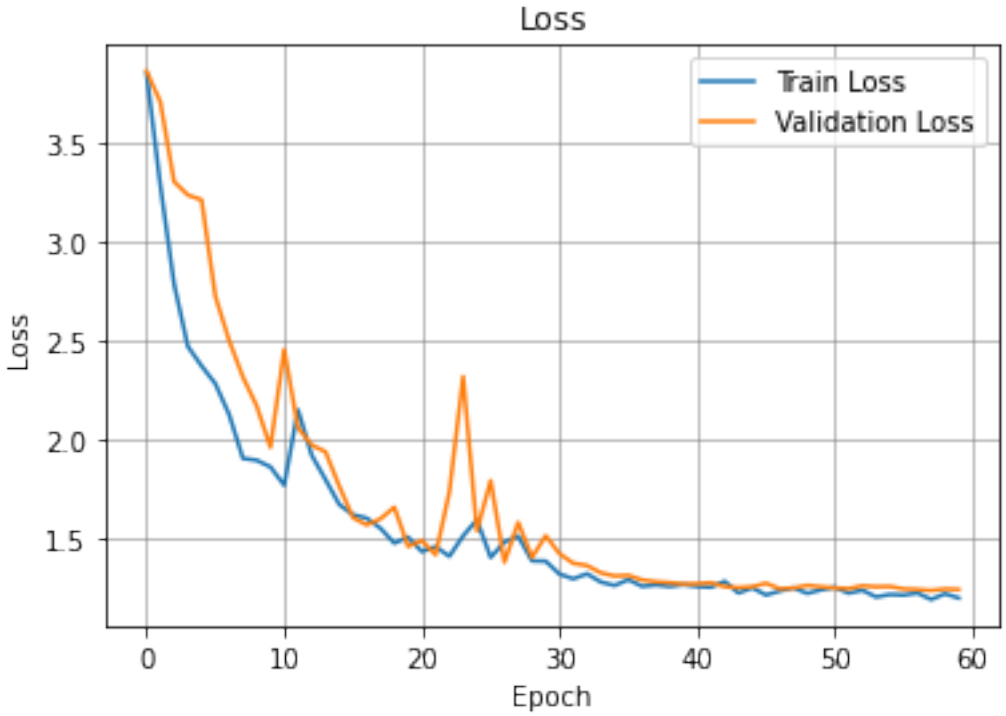
```
Epoch 1/60
17/17 [==============================] - 3s 62ms/step - loss: 3.9498 - acc:
0.1155 - val_loss: 3.8659 - val_acc: 0.0194
Epoch 2/60
17/17 [==============================] - 1s 35ms/step - loss: 3.4202 - acc:
0.2083 - val_loss: 3.7111 - val_acc: 0.2984
Epoch 3/60
17/17 [==============================] - 1s 32ms/step - loss: 2.8783 - acc:
0.3114 - val_loss: 3.3068 - val_acc: 0.2946
Epoch 4/60
17/17 [==============================] - 1s 35ms/step - loss: 2.4473 - acc:
0.4029 - val_loss: 3.2391 - val_acc: 0.2713
```

```
Epoch 5/60
17/17 [==============================] - 1s 32ms/step - loss: 2.4132 - acc:
0.4183 - val_loss: 3.2145 - val_acc: 0.3217
Epoch 6/60
17/17 [==============================] - 1s 32ms/step - loss: 2.3372 - acc:
0.3897 - val_loss: 2.7268 - val_acc: 0.3798
Epoch 7/60
17/17 [==============================] - 1s 33ms/step - loss: 2.1807 - acc:
0.4542 - val_loss: 2.5034 - val_acc: 0.3527
Epoch 8/60
17/17 [==============================] - 1s 38ms/step - loss: 1.9832 - acc:
0.5050 - val_loss: 2.3183 - val_acc: 0.3953
Epoch 9/60
17/17 [==============================] - 1s 35ms/step - loss: 1.8446 - acc:
0.5402 - val_loss: 2.1719 - val_acc: 0.5194
Epoch 10/60
17/17 [==============================] - 1s 32ms/step - loss: 1.8453 - acc:
0.4865 - val_loss: 1.9619 - val_acc: 0.5581
Epoch 11/60
17/17 [==============================] - 1s 35ms/step - loss: 1.7239 - acc:
0.5620 - val_loss: 2.4571 - val_acc: 0.3915
Epoch 12/60
17/17 [==============================] - 1s 32ms/step - loss: 2.2819 - acc:
0.4420 - val_loss: 2.0616 - val_acc: 0.4922
Epoch 13/60
17/17 [==============================] - 1s 34ms/step - loss: 1.9269 - acc:
0.5203 - val_loss: 1.9735 - val_acc: 0.5155
Epoch 14/60
17/17 [==============================] - 1s 44ms/step - loss: 1.7783 - acc:
0.5251 - val_loss: 1.9395 - val_acc: 0.5000
Epoch 15/60
17/17 [==============================] - 1s 32ms/step - loss: 1.7002 - acc:
0.5443 - val_loss: 1.7654 - val_acc: 0.5698
Epoch 16/60
17/17 [==============================] - 1s 39ms/step - loss: 1.6498 - acc:
0.5669 - val_loss: 1.6060 - val_acc: 0.5853
Epoch 17/60
17/17 [==============================] - 1s 33ms/step - loss: 1.6352 - acc:
0.5670 - val_loss: 1.5693 - val_acc: 0.6163
Epoch 18/60
17/17 [==============================] - 1s 36ms/step - loss: 1.5601 - acc:
0.5382 - val_loss: 1.6004 - val_acc: 0.6008
Epoch 19/60
17/17 [==============================] - 1s 37ms/step - loss: 1.5100 - acc:
0.5758 - val_loss: 1.6591 - val_acc: 0.5620
Epoch 20/60
17/17 [==============================] - 1s 33ms/step - loss: 1.5852 - acc:
0.5524 - val_loss: 1.4602 - val_acc: 0.6395
```

```
Epoch 21/60
17/17 [==============================] - 1s 33ms/step - loss: 1.4415 - acc:
0.5910 - val_loss: 1.4900 - val_acc: 0.6395
Epoch 22/60
17/17 [==============================] - 1s 35ms/step - loss: 1.3937 - acc:
0.6064 - val_loss: 1.4178 - val_acc: 0.6473
Epoch 23/60
17/17 [==============================] - 1s 37ms/step - loss: 1.3914 - acc:
0.5931 - val_loss: 1.7366 - val_acc: 0.5310
Epoch 24/60
17/17 [==============================] - 1s 34ms/step - loss: 1.4977 - acc:
0.5547 - val_loss: 2.3203 - val_acc: 0.4535
Epoch 25/60
17/17 [==============================] - 1s 39ms/step - loss: 1.6148 - acc:
0.5477 - val_loss: 1.5392 - val_acc: 0.5775
Epoch 26/60
17/17 [==============================] - 1s 40ms/step - loss: 1.3839 - acc:
0.5996 - val_loss: 1.7935 - val_acc: 0.4922
Epoch 27/60
17/17 [==============================] - 1s 35ms/step - loss: 1.5448 - acc:
0.5624 - val_loss: 1.3805 - val_acc: 0.6550
Epoch 28/60
17/17 [==============================] - 1s 33ms/step - loss: 1.5272 - acc:
0.5945 - val_loss: 1.5817 - val_acc: 0.5891
Epoch 29/60
17/17 [==============================] - 1s 34ms/step - loss: 1.3631 - acc:
0.5984 - val_loss: 1.4052 - val_acc: 0.6279
Epoch 30/60
17/17 [==============================] - 1s 38ms/step - loss: 1.3633 - acc:
0.6017 - val_loss: 1.5148 - val_acc: 0.5659
Epoch 31/60
17/17 [==============================] - 1s 35ms/step - loss: 1.3658 - acc:
0.6056 - val_loss: 1.4237 - val_acc: 0.6202
Epoch 32/60
17/17 [==============================] - 1s 36ms/step - loss: 1.2908 - acc:
0.6247 - val_loss: 1.3749 - val_acc: 0.6434
Epoch 33/60
17/17 [==============================] - 1s 43ms/step - loss: 1.2908 - acc:
0.6173 - val_loss: 1.3636 - val_acc: 0.6512
Epoch 34/60
17/17 [==============================] - 1s 33ms/step - loss: 1.3119 - acc:
0.6093 - val_loss: 1.3269 - val_acc: 0.6705
Epoch 35/60
17/17 [==============================] - 1s 33ms/step - loss: 1.2176 - acc:
0.6283 - val_loss: 1.3107 - val_acc: 0.6705
Epoch 36/60
17/17 [==============================] - 1s 37ms/step - loss: 1.2595 - acc:
0.6417 - val_loss: 1.3137 - val_acc: 0.6667
```

```
Epoch 37/60
17/17 [==============================] - 1s 35ms/step - loss: 1.2107 - acc:
0.6180 - val_loss: 1.2908 - val_acc: 0.6667
Epoch 38/60
17/17 [==============================] - 1s 34ms/step - loss: 1.3184 - acc:
0.5988 - val_loss: 1.2814 - val_acc: 0.6783
Epoch 39/60
17/17 [==============================] - 1s 36ms/step - loss: 1.2386 - acc:
0.6253 - val_loss: 1.2770 - val_acc: 0.6667
Epoch 40/60
17/17 [==============================] - 1s 33ms/step - loss: 1.2351 - acc:
0.6283 - val_loss: 1.2699 - val_acc: 0.6589
Epoch 41/60
17/17 [==============================] - 1s 35ms/step - loss: 1.2445 - acc:
0.6206 - val_loss: 1.2716 - val_acc: 0.6628
Epoch 42/60
17/17 [==============================] - 1s 38ms/step - loss: 1.2802 - acc:
0.6035 - val_loss: 1.2765 - val_acc: 0.6589
Epoch 43/60
17/17 [==============================] - 1s 37ms/step - loss: 1.2530 - acc:
0.6128 - val_loss: 1.2570 - val_acc: 0.6667
Epoch 44/60
17/17 [==============================] - 1s 33ms/step - loss: 1.2955 - acc:
0.5942 - val_loss: 1.2517 - val_acc: 0.6667
Epoch 45/60
17/17 [==============================] - 1s 34ms/step - loss: 1.2411 - acc:
0.6334 - val_loss: 1.2563 - val_acc: 0.6705
Epoch 46/60
17/17 [==============================] - 1s 39ms/step - loss: 1.2100 - acc:
0.6323 - val_loss: 1.2739 - val_acc: 0.6589
Epoch 47/60
17/17 [==============================] - 1s 37ms/step - loss: 1.2500 - acc:
0.6169 - val_loss: 1.2460 - val_acc: 0.6744
Epoch 48/60
17/17 [==============================] - 1s 38ms/step - loss: 1.2648 - acc:
0.6280 - val_loss: 1.2498 - val_acc: 0.6667
Epoch 49/60
17/17 [==============================] - 1s 43ms/step - loss: 1.2608 - acc:
0.6221 - val_loss: 1.2634 - val_acc: 0.6589
Epoch 50/60
17/17 [==============================] - 1s 41ms/step - loss: 1.2381 - acc:
0.6259 - val_loss: 1.2559 - val_acc: 0.6667
Epoch 51/60
17/17 [==============================] - 1s 36ms/step - loss: 1.2894 - acc:
0.6064 - val_loss: 1.2501 - val_acc: 0.6589
Epoch 52/60
17/17 [==============================] - 1s 36ms/step - loss: 1.2389 - acc:
0.6145 - val_loss: 1.2467 - val_acc: 0.6705
```

```
Epoch 53/60
17/17 [==============================] - 1s 34ms/step - loss: 1.2436 - acc:
0.6331 - val_loss: 1.2610 - val_acc: 0.6667
Epoch 54/60
17/17 [==============================] - 1s 35ms/step - loss: 1.2174 - acc:
0.6184 - val_loss: 1.2565 - val_acc: 0.6667
Epoch 55/60
17/17 [==============================] - 1s 33ms/step - loss: 1.2184 - acc:
0.6335 - val_loss: 1.2578 - val_acc: 0.6550
Epoch 56/60
17/17 [==============================] - 1s 33ms/step - loss: 1.2136 - acc:
0.6189 - val_loss: 1.2452 - val_acc: 0.6550
Epoch 57/60
17/17 [==============================] - 1s 36ms/step - loss: 1.2127 - acc:
0.6429 - val_loss: 1.2425 - val_acc: 0.6628
Epoch 58/60
17/17 [==============================] - 1s 36ms/step - loss: 1.2467 - acc:
0.6200 - val_loss: 1.2376 - val_acc: 0.6628
Epoch 59/60
17/17 [==============================] - 1s 34ms/step - loss: 1.2033 - acc:
0.6334 - val_loss: 1.2446 - val_acc: 0.6589
Epoch 60/60
17/17 [==============================] - 1s 34ms/step - loss: 1.1924 - acc:
0.6412 - val_loss: 1.2425 - val_acc: 0.6705
```



0.0.11 Save the Model

[91]:
```
model.save(MODEL_OUTPUT_PATH)

# saving
# with open('./output/tokenizer.pickle', 'wb') as handle:
#     pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

```
WARNING:absl:Found untraced functions such as
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_fn, lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.
WARNING:absl:Found untraced functions such as
lstm_cell_2_layer_call_and_return_conditional_losses, lstm_cell_2_layer_call_fn,
lstm_cell_2_layer_call_fn, lstm_cell_2_layer_call_and_return_conditional_losses,
lstm_cell_2_layer_call_and_return_conditional_losses while saving (showing 5 of
5). These functions will not be directly callable after loading.

INFO:tensorflow:Assets written to: ./output/2021-06-02/saved_model/assets

INFO:tensorflow:Assets written to: ./output/2021-06-02/saved_model/assets
```

[86]:
```
with open(MODEL_OUTPUT_LABEL_BINAIZER_PATH, 'wb') as handle:
    pickle.dump(labelBinarizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
with open(MODEL_OUTPUT_AGENT_CHAR_WORD_INDEX, 'wb') as handle:
    pickle.dump(agent_data_char_vocab_word_index, handle, protocol=pickle.
 ↪HIGHEST_PROTOCOL)
```

0.0.12 Retrieve the Model from Local Storage

[100]:
```
import pickle
import numpy as np
from tensorflow.keras.models import load_model

model = load_model(MODEL_OUTPUT_PATH)

# tokenizer = None
labelBinarizer = None
agentWordIndex = None

# with open('./output/tokenizer.pickle', 'rb') as handle:
#     tokenizer = pickle.load(handle)

with open(MODEL_OUTPUT_LABEL_BINAIZER_PATH, 'rb') as handle:
    labelBinarizer = pickle.load(handle)
with open(MODEL_OUTPUT_AGENT_CHAR_WORD_INDEX, 'rb') as handle:
    agentWordIndex = pickle.load(handle)
```

[101]:
```
def string_to_onehot(string, max_seq_length, word_index):
    # in a seq-to-seq model batch of one-hot vectors is the expected output of␣
 ↪the final softmax layer
    # vocab play the role as tokenizer.word_index, i.e., word to index
    # in the past I work on tokenizing words, this time we tokenizer every␣
 ↪single characters
    string = string.lower()
    arr = []
    while len(arr) < len(string):
        curr_index = len(arr)
        arr.append(word_index.get(string[curr_index], word_index['<unk>']))

    while len(arr) < max_seq_length:
        arr.append(word_index['<pad>'])

    onehot = np.zeros((max_seq_length, len(word_index)))
    for i in range(max_seq_length):
        onehot[i, arr[i]] = 1

    return onehot, arr
```

[102]:
```
labelBinarizer_classes_list = list(labelBinarizer.classes_)
print(labelBinarizer_classes_list)
```

```
['AGAdvisorRole', 'TYPE', '_id', '_rev', 'acceptDate', 'acceptTermVersion',
'achievements', 'agentCode', 'authGroup', 'authorised', 'authorizedHI',
'authorizedIAGST', 'authorizedIFAST', 'authorizedM8', 'authorizedM8A',
'channel', 'compCode', 'company', 'createDate', 'date', 'directorOrFirm',
'distribCode', 'email', 'faAdvisorRole', 'fullName', 'id', 'lastUpdateDate',
'lstChgDate', 'manager', 'managerCode', 'mobile', 'mobilePhone', 'name',
'officePhone', 'old_userId', 'patchDataSync', 'position', 'profileId',
'proxy1UserId', 'proxy2UserId', 'proxyEndDate', 'proxyStartDate', 'rank',
'role', 'suspEndDate', 'suspStartDate', 'tel', 'title', 'unitCode',
'upline1Code', 'upline2Code', 'userCat', 'userId', 'userRole', 'userStatus',
'userType']
```

0.0.13 Play around csv config json

[130]:
```
import json
dict=json.load(open("./csvconfigjson/csvConfig.json", encoding="utf-8"))
field_configs = dict["fields"]
```

[131]:
```
print(field_configs)
```

```
[{'fieldDisplayName': 'Agent Code', 'trainingLabel': 'agentCode', 'fieldId':
'agentCode', 'validators': [{'validationKey': 'regex', 'validationAttributes':
{'regex': '^\\d+'}}, {'validationKey': 'required', 'validationMessage': 'Agent
Code Cannot Be Empty.'}, {'validationKey': 'unique', 'validationMessage': 'Agent
```

```
Code Must Contain Unique Values.'}]}, {'fieldDisplayName': 'English Name',
'trainingLabel': 'fullName', 'fieldId': 'fullName', 'validators':
[{'validationKey': 'required', 'validationMessage': 'English Name Cannot Be
Empty.'}]}, {'fieldDisplayName': 'Primary Email', 'trainingLabel': 'email',
'fieldId': 'email', 'validators': [{'validationKey': 'regex',
'validationAttributes': {'regex': '(?:[a-z0-9!#$%&\'*+/=?^_`{|}~-]+(?:\\.[a-z0-9
!#$%&\'*+/=?^_`{|}~-]+)*|"(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b\\
x5d-\\x7f]|\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]
*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\\[(?:(?:25[0-5]|2[0-4][0-9]|[01
]?[0-9][0-9]?)\\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:
(?:[\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21-\\x5a\\x53-\\x7f]|\\\\[\\x01-\\x09\\x0
b\\x0c\\x0e-\\x7f])+)\\])'}}, {'validationKey': 'required', 'validationMessage':
'Primary Email Cannot Be Empty.'}]}, {'fieldDisplayName': 'Mobile Phone No.',
'trainingLabel': 'mobilePhone', 'fieldId': 'mobilePhone', 'validators':
[{'validationKey': 'regex', 'validationAttributes': {'regex':
'^\\+[1-9]{1}[0-9]{3,14}$'}}, {'validationKey': 'required', 'validationMessage':
'Mobile Phone No. Cannot Be Empty.'}]}, {'fieldDisplayName': 'Office Phone No.',
'trainingLabel': 'officePhone', 'fieldId': 'officePhone', 'validators':
[{'validationKey': 'regex', 'validationAttributes': {'regex':
'^\\+[1-9]{1}[0-9]{3,14}$'}}]}, {'fieldDisplayName': 'Join Date',
'trainingLabel': 'date', 'fieldId': 'joinDate', 'validators': [{'validationKey':
'correlated', 'validationAttributes': {'formula': 'OR( NOT(ISBLANK(agentCode)),
NOT(ISBLANK(email)) )'}, 'validationMessage': 'Join date require Agent or
Email'}, {'validationKey': 'required', 'validationMessage': 'Join Date cannot be
empty.'}, {'validationKey': 'dateFormat', 'validationAttributes': {'dateFormat':
'yyyy-mm-dd'}, 'validationMessage': 'Join Date must be a date in the Format
"yyyy/mm/dd".'}, {'validationKey': 'weekDay', 'validationMessage': 'Join Date
must be a weekday.'}, {'validationKey': 'WeekendDay', 'validationMessage': 'Join
Date must be a weekend day.'}, {'validationKey': 'currentMonth',
'validationMessage': 'Join Date must be within this month.'}, {'validationKey':
'currentYear', 'validationMessage': 'Join Date must be within this year.'},
{'validationKey': 'lastDayOfTheMonth', 'validationMessage': 'Join Date must be
the last day of the month.'}]}]
```

[138]:
```
indexes_of_interest = [
    labelBinarizer_classes_list.index(fieldConfig["trainingLabel"])
    for fieldConfig in field_configs
]
```

[139]:
```
print(indexes_of_interest)
```

```
[7, 24, 22, 31, 33, 19]
```

0.0.14 Define Prediction Method with Human Readable Result

[142]:
```
def predict_agent_field(inputs, threshold=0.5, n_possibilities=10):
    seqs = []
    batch_size = len(inputs)

    for input in inputs:
        seq, _ = string_to_onehot(input, MAX_SEQ_LENGTH, agentWordIndex)
        seqs.append(seq)

    seqs = np.array(seqs)

    probabilities = model.predict(seqs)
    probabilities = probabilities[..., indexes_of_interest]
    print(probabilities[:,0])
    average_probabilities = np.sum(probabilities, axis=0) / batch_size
    sorted_indexes_by_avg_probabilities = np.argsort(average_probabilities)[::
↪-1]
    sorted_avg_probabilities =␣
↪average_probabilities[sorted_indexes_by_avg_probabilities]

    sorted_classes_by_avg_probabilities = [
        labelBinarizer.classes_[indexes_of_interest[index]]
        for index in sorted_indexes_by_avg_probabilities
    ]

    return sorted_classes_by_avg_probabilities, sorted_avg_probabilities
```

[143]:
```
predict_agent_field(
    ["1/22/43",
"Jan 31, 2018",
"Apr 5, 1718",
"Oct 3, 1759",
"December 20, 1597",
"September 15, 1884",
"March 26, 1580",
"October 29, 1982",
"January 2, 1819",
"Sunday, July 14, 1816",
"Wednesday, February 3, 1723",
"Saturday, May 12, 1945",
"5-Dec-11",
"21 Nov 1854",
"28 January 1710",
"27 July 1732",
"20 February 1732",
"19 September 1828",
```

```
"20-Sep-21",
"24 June 1808",
"08 May 1794",
"23 March 1822",
"28 December 1755",
"14 September 1830",
"20/11/1857",
"6/03/1782",
"9/8/2065",
"23/8/2021",
"12/02/1692",
"01/06/1567",
"09/02/1784",
"20/4/1930",
"27/08/1694",
"18/08/1709",
"1852/08/24",
"1609/05/05",
"26/8/1911",
"Wed 23, Aug 1780",
"Sun 11, May 1541",
"Sunday 13, October 1929",
"Sunday 28, December 1552",
"Oct 12, 1903"]
)
```

```
[2.2191876e-01 2.2756198e-05 2.9753917e-05 3.3840668e-05 2.7226075e-05
 3.1168038e-05 2.5116298e-05 2.6339292e-05 2.1876254e-05 3.5434998e-05
 1.1204951e-04 2.6464246e-05 2.9357731e-02 2.8969174e-05 2.5770314e-05
 2.6907634e-05 2.1438609e-05 3.1110547e-05 1.5771473e-02 3.5134257e-05
 4.2270643e-05 2.6725638e-05 2.6021306e-05 3.2382653e-05 4.3957427e-04
 9.4111962e-03 2.5493523e-01 1.7670774e-03 2.5415493e-04 4.3964272e-04
 1.3914005e-03 5.4967953e-03 1.0601621e-03 1.4815441e-03 8.7480864e-04
 5.1782938e-04 1.8286756e-03 1.8974988e-05 1.6984599e-05 4.5862096e-05
 5.1567458e-05 2.2745373e-05]
```

```
[143]: (['date', 'mobilePhone', 'agentCode', 'fullName', 'officePhone', 'email'],
        array([8.1854755e-01, 1.3544405e-02, 1.3042643e-02, 1.0097226e-02,
               7.9686329e-04, 1.4035532e-04], dtype=float32))
```