In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.layers import Dropout
from keras.models import load_model
# from nltk.corpus import stopwords
from sklearn.feature_extraction.stop_words import ENGLISH_STOP_WORDS
import re
from nltk.corpus import stopwords
from nltk import word_tokenize
STOPWORDS = set(stopwords.words('english'))
from bs4 import BeautifulSoup
import plotly.graph_objs as go
import chart_studio.plotly as py
import cufflinks
from IPython.core.interactiveshell import InteractiveShell
import plotly.figure_factory as ff
InteractiveShell.ast_node_interactivity = 'all'
from plotly.offline import iplot
cufflinks.go_offline()
cufflinks.set_config_file(world_readable=True, theme='pearl')
```

```
Using TensorFlow backend.
```

In [2]:
```
df = pd.read_csv("F:\\Hackathon\\servicenow5.csv")

df['text'] = df['description'] + df['short_description']
```

In [3]:
```
df[["u_portfolio","opened_at","business_service","short_description","descript
ion"]].describe()
```

Out[3]:

	u_portfolio	opened_at	business_service	short_description	description
count	63	63	1	63	63
unique	5	59	1	62	63
top	IBM IT Helpdesk	8/7/2020 20:48	{'link': 'https://<instance_name>.service-now....	Reset my password	Tried to connect to SAP and all I see is a "Co...
freq	32	3	1	2	1

In [4]:

```
df[["u_portfolio","opened_at","business_service","short_description","description"]].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 63 entries, 0 to 62
Data columns (total 5 columns):
u_portfolio          63 non-null object
opened_at            63 non-null object
business_service     1 non-null object
short_description    63 non-null object
description          63 non-null object
dtypes: object(5)
memory usage: 2.5+ KB
```

In [5]:

```
df.u_portfolio.value_counts()
```

Out[5]:

```
IBM IT Helpdesk           32
PRODUCT Development       16
IBM INFRA Helpdesk        11
IBM Operation Helpdesk     2
IBM PAYROLL Helpdesk       2
Name: u_portfolio, dtype: int64
```

In [6]:

```
import matplotlib.pyplot as plt
import seaborn as sns
# Eliminate categories with fewer than 100 tickets
classifier = "u_portfolio"
ticket_threshold = 100
df_classifiers = df[df.groupby(classifier)[classifier].transform(len) > 0]
# Print number of relevant categories & shape
print("Categories: " + str(df_classifiers[classifier].nunique()))
# # Plot the classifiers
fig = plt.figure(figsize=(10,6))
sns.barplot(df_classifiers[classifier].value_counts().index, df_classifiers[cl
assifier].value_counts())
plt.xticks(rotation=20)
plt.show()
```

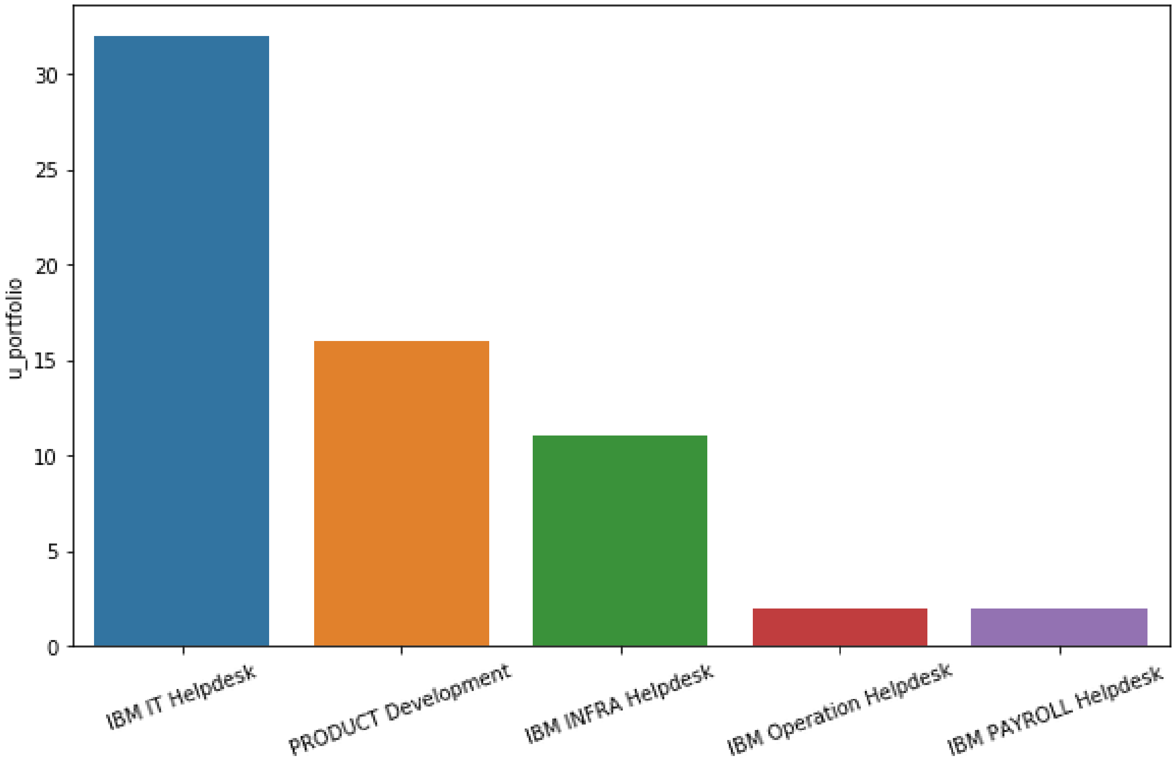
```
Categories: 5
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11ed0a88>
```

Out[6]:

```
(array([0, 1, 2, 3, 4]), <a list of 5 Text xticklabel objects>)
```



In [7]:

```
def print_plot(index):
    example = df[df.index == index][['text', 'u_portfolio']].values[0]
    if len(example) > 0:
        print(example[0])
        print('PortFolio:', example[1])
print_plot(62)
```

```
Need access to the common drive for sharing files which can be accessed by al
l members. Please provide access.Need access to the common drive.
PortFolio: PRODUCT Development
```

In [8]:

```
df = df.reset_index(drop=True)
REPLACE_BY_SPACE_RE = re.compile('[/(){}\[\]\|@,;]')
BAD_SYMBOLS_RE = re.compile('[^0-9a-z #+_]')
STOPWORDS = set(stopwords.words('english'))

STOPLIST = set(stopwords.words('english') + list(ENGLISH_STOP_WORDS))

def clean_text(text):
    """
        text: a string

        return: modified initial string
    """
    text = text.lower() # lowercase text
    text = REPLACE_BY_SPACE_RE.sub(' ', text) # replace REPLACE_BY_SPACE_RE symbols by space in text. substitute the matched string in REPLACE_BY_SPACE_RE with space.
    text = BAD_SYMBOLS_RE.sub('', text) # remove symbols which are in BAD_SYMBOLS_RE from text. substitute the matched string in BAD_SYMBOLS_RE with nothing.
    text = text.replace('x', '')
#    text = re.sub(r'\W+', '', text)
    text = ' '.join(word for word in text.split() if word not in STOPWORDS) # remove stopwors from text
    return text
df['text'] = df['text'].apply(clean_text)
df['text'] = df['text'].str.replace('\d+', '')
```

In [9]:

```
print_plot(10)
```

```
hard drive making loud grinding noise last two daysseem issue hard drive
PortFolio: IBM INFRA Helpdesk
```

In [10]:

```
# The maximum number of words to be used. (most frequent)
MAX_NB_WORDS = 50000
# Max number of words in each complaint.
MAX_SEQUENCE_LENGTH = 250
# This is fixed.
EMBEDDING_DIM = 100
tokenizer = Tokenizer(num_words=MAX_NB_WORDS, filters='!"#$%&()*+,-./:;<=>?@[\]^_`{|}~', lower=True)
tokenizer.fit_on_texts(df['text'].values)
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
```

```
Found 432 unique tokens.
```

In [11]:

```
X = tokenizer.texts_to_sequences(df['text'].values)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', X.shape)
```

```
Shape of data tensor: (63, 250)
```

In [12]:
```
Y = pd.get_dummies(df['u_portfolio']).values
print('Shape of label tensor:', Y.shape)
```

```
Shape of label tensor: (63, 5)
```

In [13]:

```
Y
```

```
Out[13]: array([[0, 1, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 1, 0, 0],
                [0, 1, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 0, 1],
                [0, 1, 0, 0, 0],
                [0, 0, 1, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 1, 0, 0, 0],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 0, 1],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 1, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 0, 1],
                [0, 0, 0, 0, 1],
                [1, 0, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 1, 0, 0, 0],
                [0, 0, 0, 1, 0],
                [0, 0, 0, 0, 1]], dtype=uint8)
```

In [14]:
```
A = pd.DataFrame(df['u_portfolio'])
```

In [15]:
```
A['u_portfolio'].value_counts()
```

Out[15]:
```
IBM IT Helpdesk            32
PRODUCT Development        16
IBM INFRA Helpdesk         11
IBM Operation Helpdesk      2
IBM PAYROLL Helpdesk        2
Name: u_portfolio, dtype: int64
```

In [17]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.3, random_state = 42)
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)

model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X.shape[1]))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(25, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(5, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])



epochs = 10
batch_size = 15

history = model.fit(X_train, Y_train, epochs=epochs, batch_size=batch_size,validation_split=0.1,callbacks=[EarlyStopping(monitor='val_loss', patience=3, min_delta=0.0001)])

print(history)
```

```
(44, 250) (44, 5)
(19, 250) (19, 5)
WARNING:tensorflow:From C:\Users\Asus\Anaconda3\envs\nlp_course\lib\site-pack
ages\tensorflow\python\ops\resource_variable_ops.py:435: colocate_with (from
tensorflow.python.framework.ops) is deprecated and will be removed in a futur
e version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\Asus\Anaconda3\envs\nlp_course\lib\site-pack
ages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.pytho
n.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 39 samples, validate on 5 samples
Epoch 1/10
39/39 [==============================] - 15s 387ms/step - loss: 1.6028 - accu
racy: 0.3077 - val_loss: 1.5983 - val_accuracy: 0.4000
Epoch 2/10
39/39 [==============================] - 9s 238ms/step - loss: 1.5731 - accur
acy: 0.6667 - val_loss: 1.5825 - val_accuracy: 0.4000
Epoch 3/10
39/39 [==============================] - 10s 245ms/step - loss: 1.5480 - accu
racy: 0.7179 - val_loss: 1.5616 - val_accuracy: 0.4000
Epoch 4/10
39/39 [==============================] - 7s 187ms/step - loss: 1.5184 - accur
acy: 0.7692 - val_loss: 1.5338 - val_accuracy: 0.6000
Epoch 5/10
39/39 [==============================] - 6s 164ms/step - loss: 1.4639 - accur
acy: 0.8462 - val_loss: 1.4921 - val_accuracy: 0.6000
Epoch 6/10
39/39 [==============================] - 6s 165ms/step - loss: 1.4316 - accur
acy: 0.7436 - val_loss: 1.4288 - val_accuracy: 0.6000
Epoch 7/10
39/39 [==============================] - 7s 173ms/step - loss: 1.3727 - accur
acy: 0.6667 - val_loss: 1.3450 - val_accuracy: 0.6000
Epoch 8/10
39/39 [==============================] - 7s 169ms/step - loss: 1.2847 - accur
acy: 0.6923 - val_loss: 1.2565 - val_accuracy: 0.6000
Epoch 9/10
39/39 [==============================] - 6s 160ms/step - loss: 1.2607 - accur
acy: 0.6923 - val_loss: 1.1829 - val_accuracy: 0.6000
Epoch 10/10
39/39 [==============================] - 7s 185ms/step - loss: 1.1587 - accur
acy: 0.7179 - val_loss: 1.1346 - val_accuracy: 0.6000
<keras.callbacks.callbacks.History object at 0x0000000013725888>
```

In [ ]:

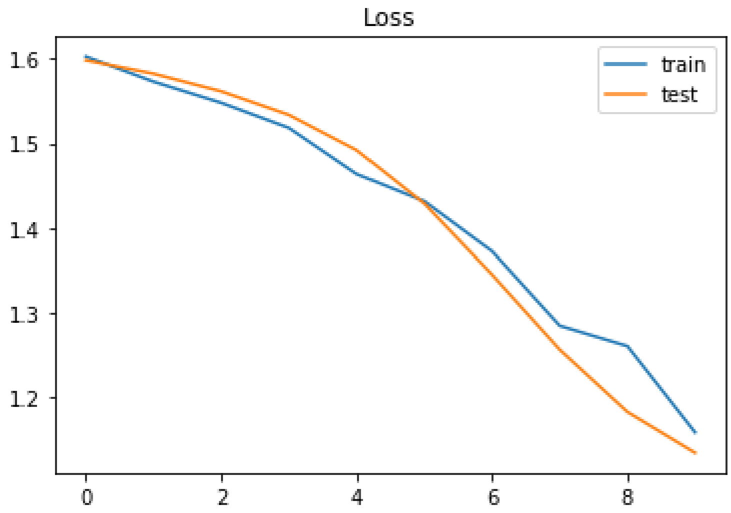
In [18]:

```
accr = model.evaluate(X_test,Y_test)
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1
]))
```

```
19/19 [==============================] - 1s 37ms/step
Test set
  Loss: 1.156
  Accuracy: 0.632
```

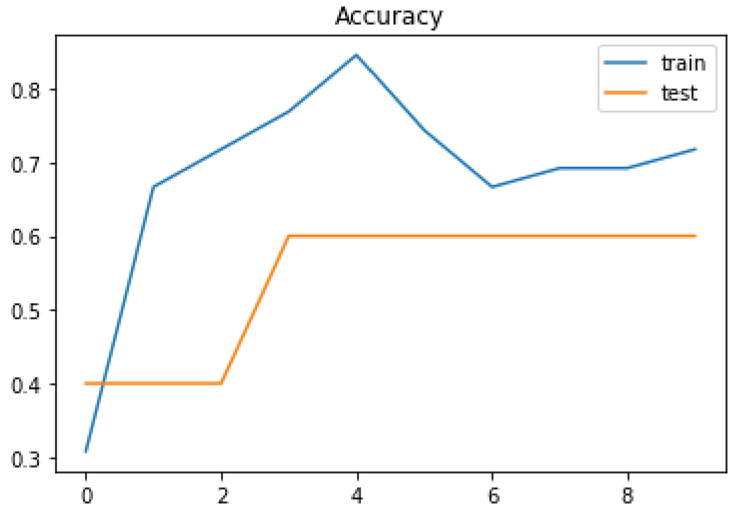
In [19]:

```
plt.title('Loss')
plt.plot(history.history['loss'], label='train')
plt.plot(history.history['val_loss'], label='test')
plt.legend()
plt.show();
```



In [20]:

```
plt.title('Accuracy')
plt.plot(history.history['accuracy'], label='train')
plt.plot(history.history['val_accuracy'], label='test')
plt.legend()
plt.show();
```



In [27]:

```
new_complaint = ['Currently running 10GR1 and need to upgrade to 10GR2.Need Oracle 10GR2 installed']
seq = tokenizer.texts_to_sequences(new_complaint)
padded = pad_sequences(seq, maxlen=MAX_SEQUENCE_LENGTH)
pred = model.predict(padded)
labels = ['IBM INFRA Helpdesk','IBM IT Helpdesk', 'IBM Operation Helpdesk', 'IBM PAYROLL Helpdesk', 'PRODUCT Development']
print(pred, labels[np.argmax(pred)])
```

```
[[0.1790213  0.41892007 0.04473705 0.04996657 0.30735496]] IBM IT Helpdesk
```

In [22]:

```
df.u_portfolio.value_counts()
```

Out[22]:

```
IBM IT Helpdesk           32
PRODUCT Development       16
IBM INFRA Helpdesk        11
IBM Operation Helpdesk     2
IBM PAYROLL Helpdesk       2
Name: u_portfolio, dtype: int64
```

In [23]:

```
pred
```

Out[23]:

```
array([[0.17336534, 0.42993852, 0.04501655, 0.05140622, 0.3002734 ]],
      dtype=float32)
```