Zero Shot Learning for Multilingual Keywords Classification

In this notebook we will be performing the task of assigning label names to multi-lingual keywords by using Zero Shot Learning. The approach has been tested on different datasets and the results generated as keywords with their assigned labl names with their corresponding accuracy score.

This project/notebook consists of several Tasks.

- **Task 1**: Installing the dependencies.
- **Task 2**: Importing the required libraries in the environment.
- **Task 3**: Instantiating the classifier by using huggingface pipeline
- **Task 4**: Forming Class Names to which the keywords will be assigned to
- **Task 5**: Passing the keywords and the class names through the classifier
- **Task 6**: Analysis of the labels assigned to keywords

Task 1: Installing the dependencies

In [1]:

```
!pip install sentencepiece
```

```
Requirement already satisfied (use --upgrade to upgrade): sentencepiece in /opt/cond
a/lib/python3.7/site-packages
You are using pip version 8.1.1, however version 21.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

Task 2: Importing all the required libraries in the environment.

In [2]:

```
#Importing the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow

import sentencepiece
import transformers
from transformers import pipeline

from transformers import AutoTokenizer, AutoModelForSequenceClassification

import nltk

import plotly as py
import plotly.graph_objs as go
import ipywidgets as widgets
from scipy import special
import plotly.express as px

py.offline.init_notebook_mode(connected = True)
import scipy.stats as stats

import warnings
warnings.filterwarnings("ignore")
```

```
2021-10-21 09:26:52.938943: W tensorflow/stream_executor/platform/default/dso_loade
r.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.1
1.0: cannot open shared object file: No such file or directory
2021-10-21 09:26:52.938994: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ign
ore above cudart dlerror if you do not have a GPU set up on your machine.
```

In [4]:

```
#classifier = pipeline("zero-shot-classification",
#                       model="joeddav/xlm-roberta-large-xnli")
```

```
Some weights of the model checkpoint at joeddav/xlm-roberta-large-xnli were not used
when initializing XLMRobertaForSequenceClassification: ['roberta.pooler.dense.bias',
'roberta.pooler.dense.weight']
- This IS expected if you are initializing XLMRobertaForSequenceClassification from
the checkpoint of a model trained on another task or with another architecture (e.g.
initializing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing XLMRobertaForSequenceClassification f
rom the checkpoint of a model that you expect to be exactly identical (initializing
a BertForSequenceClassification model from a BertForSequenceClassification model).
```

Task 3: Instantiating the classifier by using huggingface pipeline

In [3]:

```
classifier = pipeline("zero-shot-classification", model="vicgalle/xlm-roberta-large-
```

In [ ]:

```
df = pd.read_csv("df.csv")
df = df[["Keyword","Label"]]
df.head()
```

df["Label"].value_counts()

In [ ]:

```
df_copy = df.copy()
df_copy = df_copy[["Keyword"]]
df_copy.head()
```

Task 4: Forming Class Names to which the keywords will be assigned to

In [43]:

```
classes = ['Informational',
           'Local',
           'Transactional',
           'Navigational']
```

In [44]:

```
keyword = df_copy['Keyword'][0]
```

In [ ]:

```
result = classifier(keyword, classes, multi_label=False)
result
```

Task 5: Passing the keywords and the class names through the classifier

In [46]:

```
df_copy['labels'] = df_copy.apply(lambda x: classifier(x.Keyword, classes, multi_lab
```

In [47]:

```
df_copy['predicted_label'] = df_copy.apply(lambda row: row['labels']['labels'][0], a
df_copy['score'] = df_copy.apply(lambda row: row['labels']['scores'][0], axis=1)
```

In [ ]:

```
df_copy.head(10)
```

Task 6: Analysis of the labels assigned to keywords

In [ ]:

```
result = pd.merge(df, df_copy, on='Keyword', how='inner')
result = result[['Keyword','Label','predicted_label','score']]
#result = result.groupby('predicted_label').head(20).reset_index(drop=False)
result.head(20)
```

In [ ]:

```
result.loc[result['predicted_label'] == 'Local'].head(7)
#result.loc[result['column_name'] == some_value]
```

In [ ]:

```
result.loc[result['predicted_label'] == 'Navigational'].head(7)
```

In [ ]:

```
result[result.duplicated(['Keyword'], keep=False)].head(10)
```

In [ ]:

```
df.shape[0] - len(df['Keyword'].unique())
```

In [ ]:

```
df.shape
```

In [142…

```
#Number of duplicate values
1615-1253
```

Out[142…

```
362
```

In [ ]:

```
result.loc[result['predicted_label'] == 'Transactional'].head(20)
```

In [ ]:

```
result.tail()
```

In [ ]:

```
keyword_len = []

for index, row in result.iterrows():
    #print(len(row['keyword'].split()))
    keyword_len.append(len(row['Keyword'].split()))

print(f'Average number of words in the keyword are: {np.mean(keyword_len)}' )
```

In [ ]:

```
y = stats.norm.pdf(np.linspace(1,10,50), np.mean(keyword_len), np.std(keyword_len))

plt.hist(keyword_len, bins= range(1,10), density = True)
plt.plot(np.linspace(0,14,50), y, linewidth = 1)
plt.title("Keyword length")
plt.xlabel("Number of words")
plt.ylabel("Probability");
```

In [ ]:

```
from plotly.subplots import make_subplots
import plotly.graph_objects as go


fig = make_subplots(
    rows=2, cols=2,
    subplot_titles=("Distribution of Label in Dataset by SEO team  (Multilingual)",

fig.add_trace(go.Histogram(x=df['Label']),
              row=1, col=1)

fig.add_trace(go.Histogram(x=result['predicted_label']),
              row=1, col=2)

fig.update_layout(height=560, width=1200,
                  title_text="Difference in Keyword Labelling")

fig.show()
```

In [ ]:

```
import pandas as pd
url="df2.url"
c= pd.read_csv(url)
c.head()
```

In [ ]:

```
classes = ["Transactional",
         "Branded",
         "Visual",
         "Research",
         "Answer",
         "Fresh / News",
         "Local",
         "Video"]
```

In [ ]:

```
result_ = classifier(c['Keyword'][1], classes, multi_label=False)
result_
```

In [ ]:

```
c['labels'] = c.apply(lambda x: classifier(x.Keyword, classes, multi_label=False), a
```

In [ ]:

```
c['predicted_label'] = c.apply(lambda row: row['labels']['labels'][0], axis = 1)
c['score'] = c.apply(lambda row: row['labels']['scores'][0], axis=1)
```

In [ ]:

```
c.head()
```

In [ ]:

```

```