Preparing the Training Dataset

Let's step through preparing the dataset for building a custom language classification model with Cloud AutoML NLP:

1. The training input can either be a document in (.txt) format or as an in-line text in a (.csv) file. Multiple texts can be grouped as a compressed (.zip) file.
2. For this project, text files are placed in sub-folders with their grouped output labels as the folder names. This is later used to create a CSV file containing the data file path and their labels. For example:
   - [files]
     - [toxic]
     - [clean]
3. Next, a CSV must be generated that points to the paths of the images and their corresponding label. Just like Cloud Vision, Cloud NLP uses the CSV file to point to the location of the training documents or words and their corresponding labels. The CSV file is placed in the same GCS bucket created AutoML NLP was configured. In our case, this bucket is named 'gs://quantum-ally-219323-lcm'. The following code segment prepares the data and produces a CSV file.

```
import numpy as np
import pandas as pd
import re
import pathlib
import os

# read the Toxic Comment Classification training dataset
data = pd.read_csv('./data/train.csv')

# add clean column label
data['clean'] = (1 - data.iloc[:, 2:].sum(axis=1) >= 1).
astype(int)
```

```
# merge all other non-clean comments to toxic
data.loc[data['clean'] == 0, ['toxic']] = 1

# select dataframe of clean examples
data_clean = data[data['clean'] == 1].sample(n=20000)
# select dataframe of toxic examples
data_toxic = data[data['toxic'] == 1].sample(n=16000)

# join into one dataframe
data = pd.concat([data_clean, data_toxic])

# remove unused columns
data.drop(['severe_toxic', 'obscene', 'threat', 'insult',
'identity_hate'], axis=1, inplace=True)

# create text documents and place them in their folder classes.
for index, row in data.iterrows():
    comment_text = re.sub(r'[^\w\s]',",row['comment_text']).
    rstrip().lstrip().strip()
    classes = "
    if (row['toxic'] == 1):
        classes = 'toxic'
    else:
        classes = 'clean'

    pathlib.Path("./file/{}".format(classes)).mkdir(parents=True,
    exist_ok=True)
    with open("./file/{}/text_{}.txt".format(classes,index), "w")
    as text_file:
        text_file.write(comment_text)

data_path = []
directory = 'file/'

# create data csv
for subdir, dirs, files in os.walk(directory):
    for file in files:
        filepath = subdir + os.sep + file
```

```
            if filepath.endswith(".txt"):
                entry = ['{}/{}'.format('gs://quantum-ally-219323-
                lcm',filepath), os.path.basename(subdir)]
                data_path.append(entry)

# convert to Pandas DataFrame
data_pd = pd.DataFrame(np.array(data_path))

# export data to csv
data_pd.to_csv("data.csv", header=None, index=None)
```

4. The preceding code will result in a CSV looking like the following sample:

```
gs://quantum-ally-219323-lcm/file/clean/text_100055.txt,clean
gs://quantum-ally-219323-lcm/file/clean/text_100059.txt,clean
gs://quantum-ally-219323-lcm/file/clean/text_100077.txt,clean
...
gs://quantum-ally-219323-lcm/file/toxic/text_141122.txt,toxic
gs://quantum-ally-219323-lcm/file/toxic/text_141138.txt,toxic
gs://quantum-ally-219323-lcm/file/toxic/text_141143.txt,toxic
```

   The first part is the image path or URI, while the other is the document label.

5. When preparing the text dataset, it is useful to have a '**None_of_the_above'** class. This class will contain documents that do not belong to any of the predicted classes. Adding this class can have an overall effect on the model accuracy.

6. Navigate to the folder chapter and copy the image files to the GCS bucket. The flag **-m** initiates parallel uploads to speed up upload time of large document sizes to GCP.

```
gsutil -m cp -r file gs://quantum-ally-219323-lcm
```

7. Copy the CSV data file containing the document paths and their labels to the GCS bucket.

```
gsutil cp data.csv gs://quantum-ally-219323-lcm/file/
```

Building a Custom Language Classification Model on Cloud AutoML NLP

This section will walk through creating a document dataset and building a custom language classification model on AutoML Vision.

1. From the Cloud AutoML NLP dashboard, click **NEW DATASET** as shown in Figure 43-5.

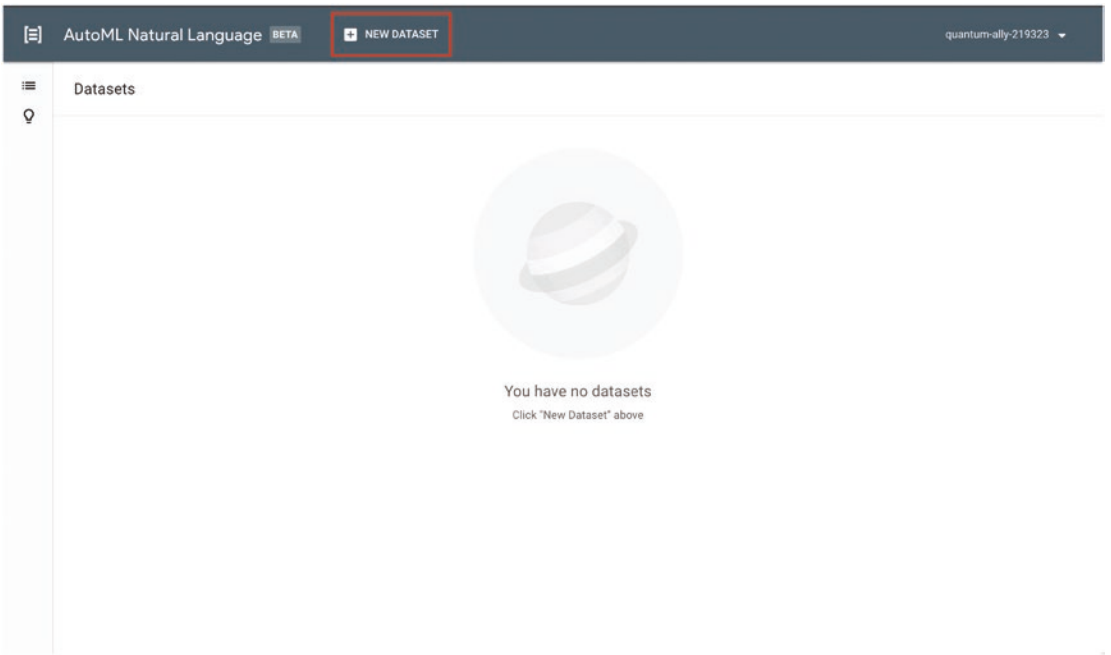

Figure 43-5. *New Dataset on AutoML NLP*

2. To create a Dataset on Cloud AutoML NLP, set the following parameters as shown in Figure 43-6: