CHAPTER 43

Google AutoML: Cloud Natural Language Processing

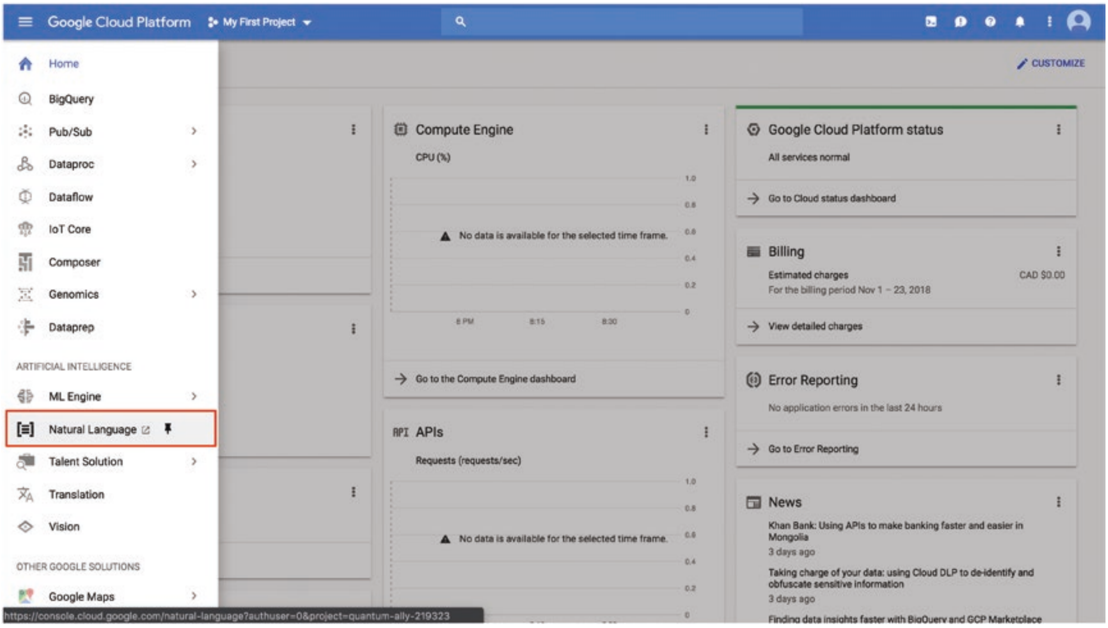
This chapter will build a language toxicity classification model to classify and recognize toxic and non-toxic or clean phrases using Google Cloud AutoML for natural language processing (NLP). The data used in this project is from the Toxic Comment Classification Challenge on Kaggle by Jigsaw and Google. The data is modified to have a sample of 16,000 toxic and 16,000 non-toxic words as inputs to build the model on AutoML NLP.

Enable AutoML NLP on GCP

The following steps will enable AutoML NLP on GCP:

1. Click the triple dash in the top-left corner of the interface and select **Natural Language** under the category ARTIFICIAL INTELLIGENCE as shown in Figure 43-1.

E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*,
https://doi.org/10.1007/978-1-4842-4470-8_43



No data is available for the selected time frame.

All services normal

For the billing period Nov 1 – 23, 2018

No application errors in the last 24 hours

Khan Bank: Using APIs to make banking faster and easier in Mongolia
3 days ago

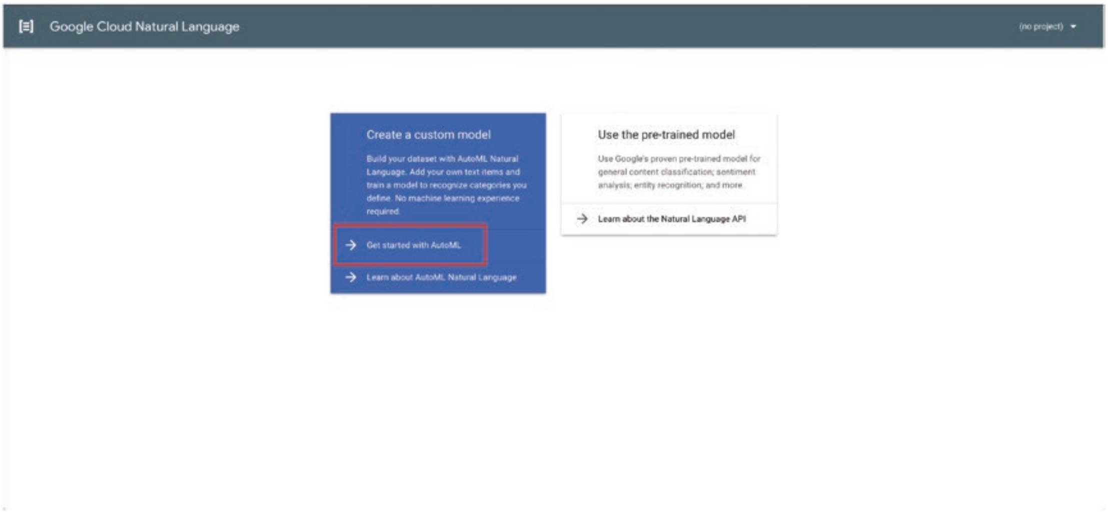
Taking charge of your data: using Cloud DLP to de-identify and obfuscate sensitive information
3 days ago

Finding data insights faster with BigQuery and GCP Marketplace

https://console.cloud.google.com/natural-language?authuser=0&project=quantum-ally-219323

***Figure 43-1.** Open Cloud AutoML for Natural Language*

2. From the screen that follows, click **Get started with AutoML** (see Figure 43-2).

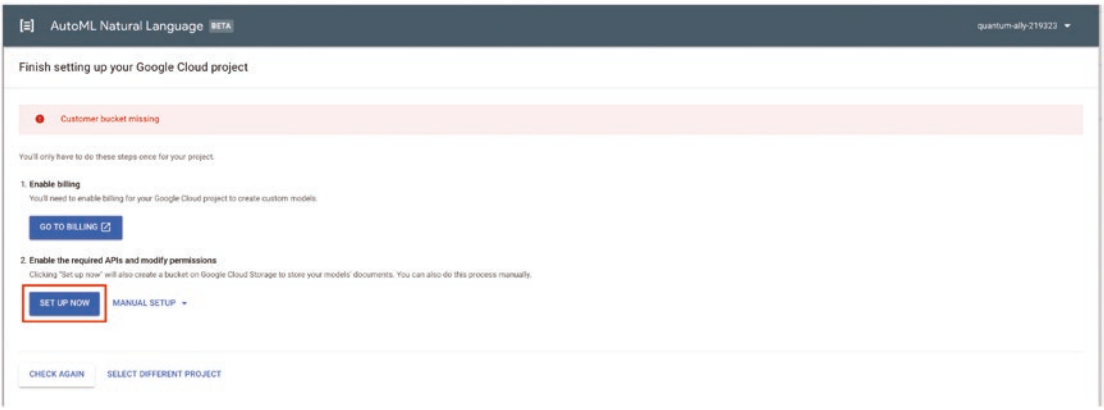

Google Cloud Natural Language
(no project)
Create a custom model
Get started with AutoML
Learn about AutoML Natural Language
Use the pre-trained model
Learn about the Natural Language API

Build your dataset with AutoML Natural Language. Add your own text items and train a model to recognize categories you define. No machine learning experience required.

Use Google's proven pre-trained model for general content classification, sentiment analysis, entity recognition, and more.

***Figure 43-2.** Click Get started with Cloud AutoML NLP*

3. Click **SET UP NOW** to automatically setup the GCP project for working with Cloud AutoML NLP (see Figure 43-3). This process involves activating the API for AutoML and creating a bucket on GCP for storing the data input and output models. We will use this bucket in the next section.



Finish setting up your Google Cloud project

Customer bucket missing

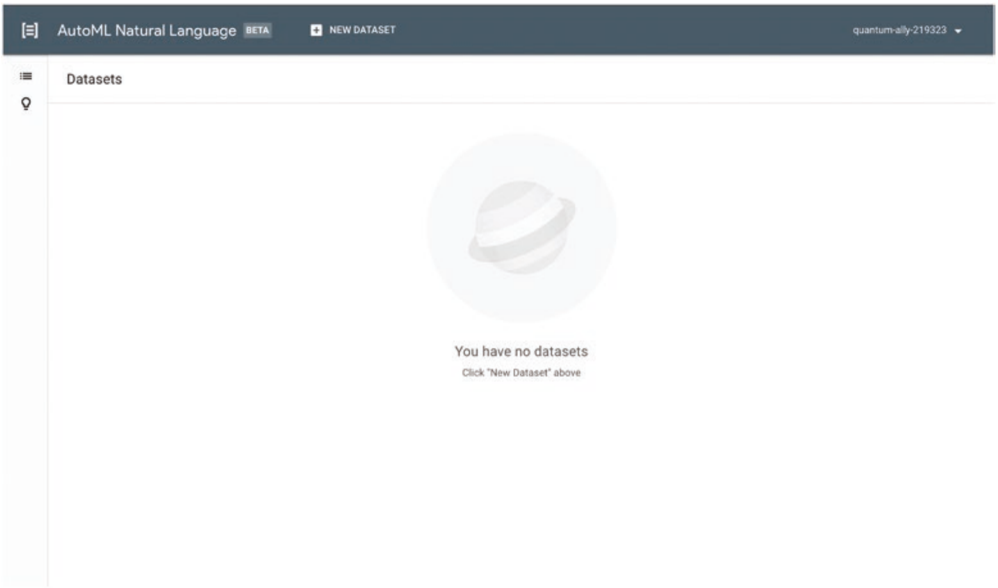
You'll only have to do these steps once for your project.

1. Enable billing
You'll need to enable billing for your Google Cloud project to create custom models.

2. Enable the required APIs and modify permissions
Clicking "Set up now" will also create a bucket on Google Cloud Storage to store your models' documents. You can also do this process manually.

Figure 43-3. Auto-configure Cloud AutoML NLP

4. After configuration, the Cloud AutoML NLP Dashboard is activated (see Figure 43-4).



You have no datasets

Click "New Dataset" above

Figure 43-4. AutoML NLP dashboard

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)
```