

GOOGLE NATURAL LANGUAGE API AND SENTIMENT ANALYSIS



Here we discuss Natural Language processing using the Google Natural Language API. Our goal is to do sentiment analysis.

Definition

Sentiment analysis means seeing whether what someone writes is positive or negative. Business can use this to look at Twitter, Yelp, or whoever offers a API and then change their marketing, practices, or even reach out to the person who has complained about their brand and offer to fix what has irked them so.

To do sentiment analysis you could write your own code or use any of the many cloud APIs from different vendors and pay for the service. Writing it yourself would save you money. You just need to understand concepts like **bag of words** and master the [NLP APIs](#) in a deep learning ML library like Torch.

Sign up for a Google Cloud free trial [here](#) and enable the Google Natural Language Processing (NLP) API. This gives you \$300 credit that you can use in 365 days. Don't worry, they promise they will not bill your credit card without asking you first. So you should be able to use it for free for this tutorial and other tutorials that we will write on Google Cloud.

Setup

You need to install the Google cloud utilities on your system.

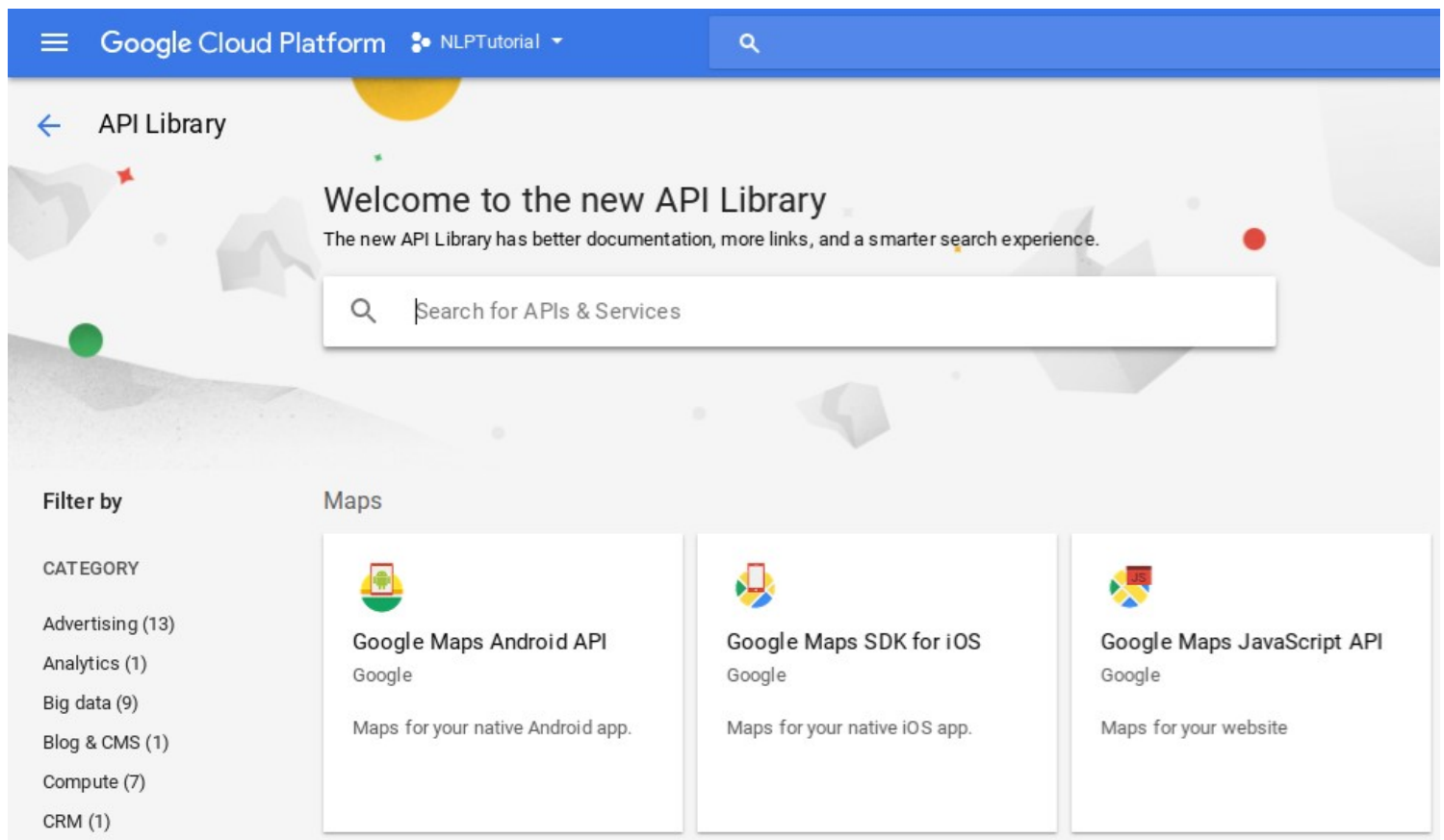
Download the Google cloud utilities and SDK following [these instructions](#) to set it up, but **STOP** at the **gcloud init** step since we are using the NLPTutorial, which is already partially set up.

Enable API for project

Go to the [Google console](#) and pick the **NLPTutorial** project, which should already be there. if it is not you have not signed up and enables the NLP trial.

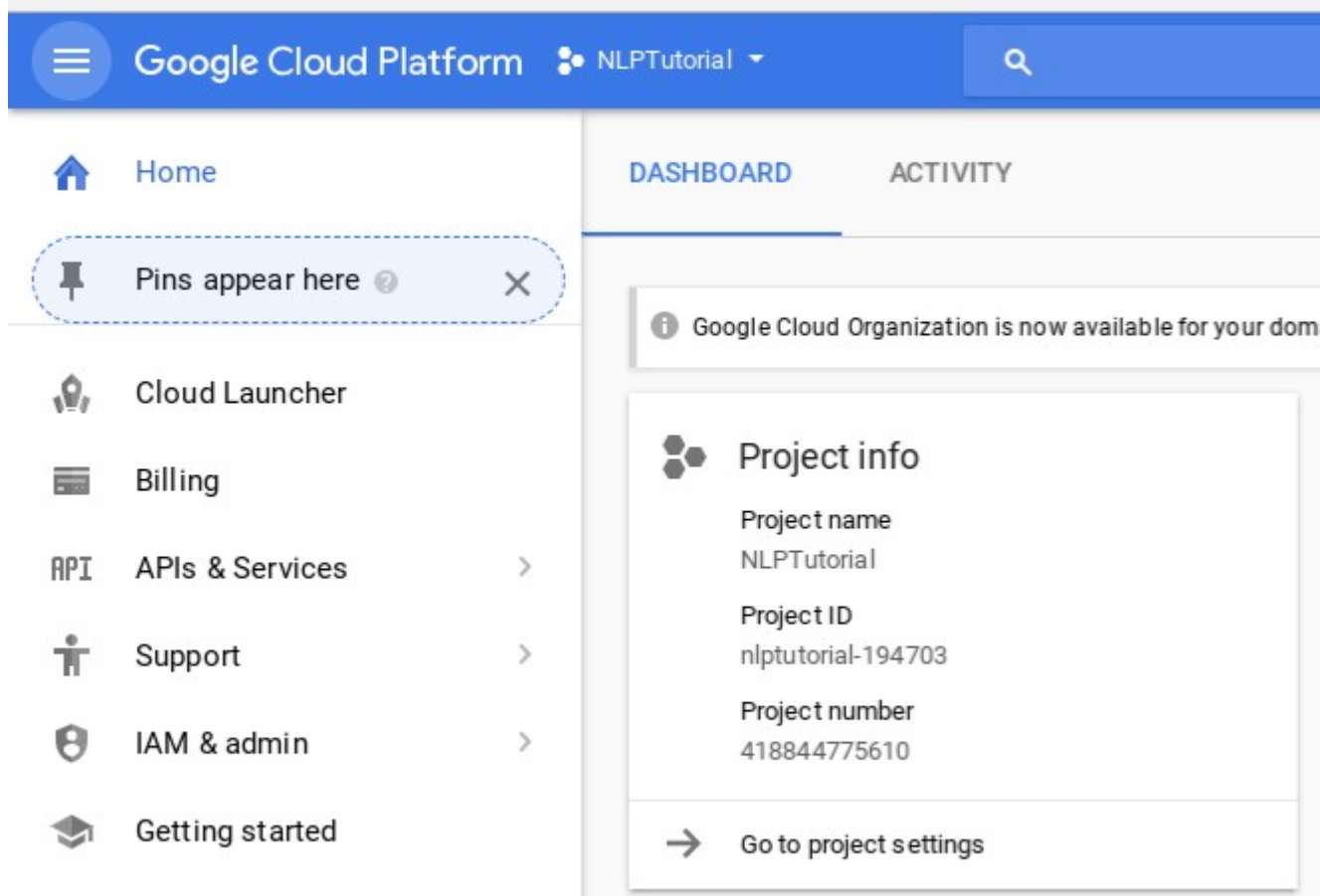
Go to **Enable APIs and Services**.

Enable both **Natural Language** processing and Google Service Mangement APIs by typing the first few letters in this screen and picking each:



If you cannot find that screen go to the URL directly:

[https://console.developers.google.com/apis/library/cloudresourcemanager.googleapis.com/?project=\(name it shows on as Project id\)](https://console.developers.google.com/apis/library/cloudresourcemanager.googleapis.com/?project=(name it shows on as Project id)) like this:



If the API is setup correctly, the screen should look like this:

The screenshot shows the 'APIs & Services' dashboard. The left sidebar has links for Dashboard, Library, and Credentials. The main content area has a 'Dashboard' tab and a '+ ENABLE APIS AND SERVICES' button. Below this is a table listing various APIs with their status, requests, errors, error ratio, and latency.

API	Requests	Errors	Error ratio	Latency, median	Latency, 98%	
Google Cloud Natural Language API	2	0	0%	66 ms	257 ms	Disable
Google Service Management API	1	1	100%	393 ms	519 ms	Disable
BigQuery API	—	—	—	—	—	Disable
Google Cloud APIs	—	—	—	—	—	Disable
Google Cloud Datastore API	—	—	—	—	—	Disable
Google Cloud Resource Manager API	—	—	—	—	—	Disable
Google Cloud SQL	—	—	—	—	—	Disable
Google Cloud Storage	—	—	—	—	—	Disable
Google Cloud Storage JSON API	—	—	—	—	—	Disable
Stackdriver Debugger API	—	—	—	—	—	Disable
Stackdriver Logging API	—	—	—	—	—	Disable
Stackdriver Monitoring API	—	—	—	—	—	Disable
Stackdriver Trace API	—	—	—	—	—	Disable

Download a service account key in JSON format following **these directions**.

Set the **GOOGLE_APPLICATION_CREDENTIALS** environment variable to point to it:

```
export GOOGLE_APPLICATION_CREDENTIALS=/home/walker/Documents/nlp/google.json
```

Run Sentiment Analysis

Now we will test this by writing a complaint about a restaurant we visited. Type:

```
gcloud ml language analyze-sentiment --content="The service in your restaurant is terrible. My wife and I are never coming back."
```

As you can see the **sentiment** is negative (< 0). To understand how negative you need to study what **magnitude and score** mean.

```
{
  "documentSentiment": {
    "magnitude": 1.2,
    "score": -0.6
  },
  "language": "en",
  "sentences":
}
```

Now write something positive:

```
gcloud ml language analyze-sentiment --content="We love your restaurant and will recommend it to all our friends."
```

Now the sentiment is positive (> 0).

```
{
  "documentSentiment": {
    "magnitude": 0.9,
    "score": 0.9
  },
  "language": "en",
  "sentences":
}
```

Next you could try to the same thing using Python following [the instructions](#) provided by Google.