Technology Review

Google Natural Language API

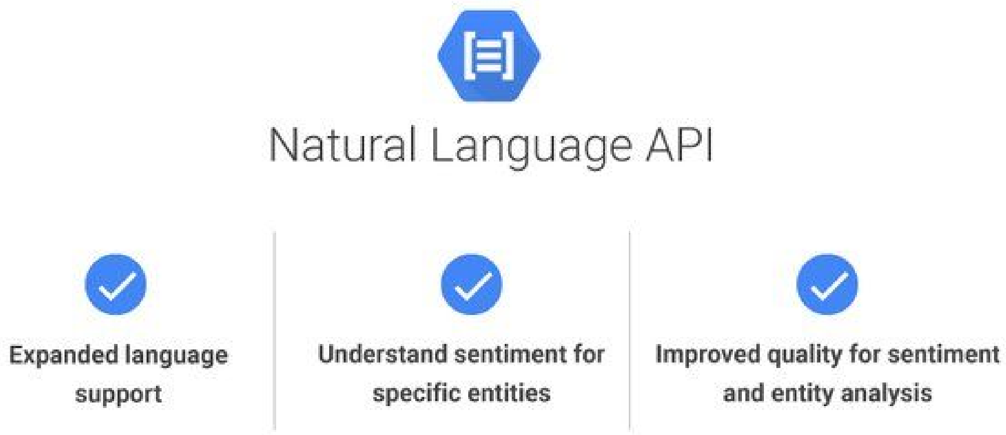
Author

Saad Rasheed
Srashee2

Table of Contents

Introduction

This technology review is aimed at reviewing Google’s Natural Language API. This allows users to extract information from unstructured data such as information about people, places and events. The framework also includes the following frameworks; AutoML Natural Language, Natural Language API and Healthcare Natural Language AI. In this review we will be exploring this API to see what it has to offer.


Natural Language API
Expanded language support
Understand sentiment for specific entities
Improved quality for sentiment and entity analysis

Google Natural Language API

The Google Natural Language API includes different ways to access the API. When you first look into it you are introduced to the Google Cloud Console, a web based UI “to provision, configure, manage and monitor systems that use Google Cloud products”[1]. The great thing about this API is that you don’t need to train your own model as this is a SAAS. You are able to have a powerful trained modeling tool just by accessing this API.

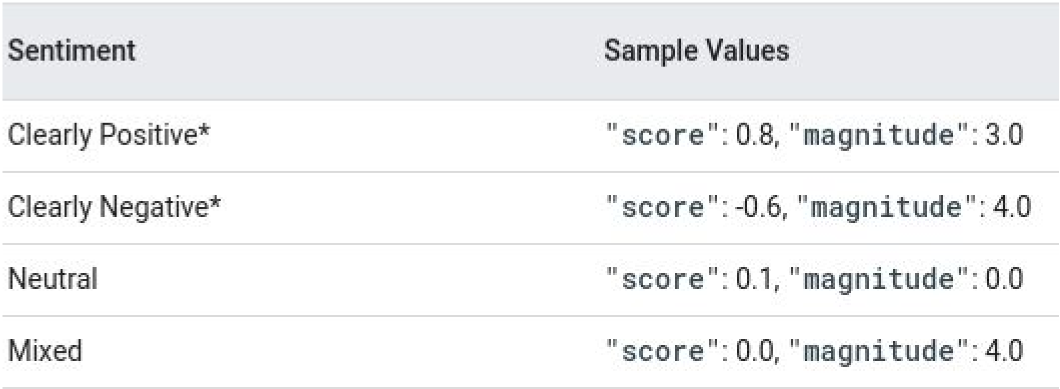
Let’s look at some different features this framework has to offer.

Analyzing Sentiment

This allows a user to directly access an API and provide a document for sentiment analysis. More specifically sentiment analysis allows us to get subjective information and emotional states of text data. This is a very powerful tool because it allows anyone who’s willing to use this framework to have a very quick and powerful way to check sentiments of text.

Take for example a business that has a lot of feedback from reviews but aren’t unable to manually parse each review due to the amount of text. This business could quickly use this API and provide their reviews to run sentiment analysis on the data.

Interpreting the sentiment analysis data is very easy, please see the chart below provided by the documentation.

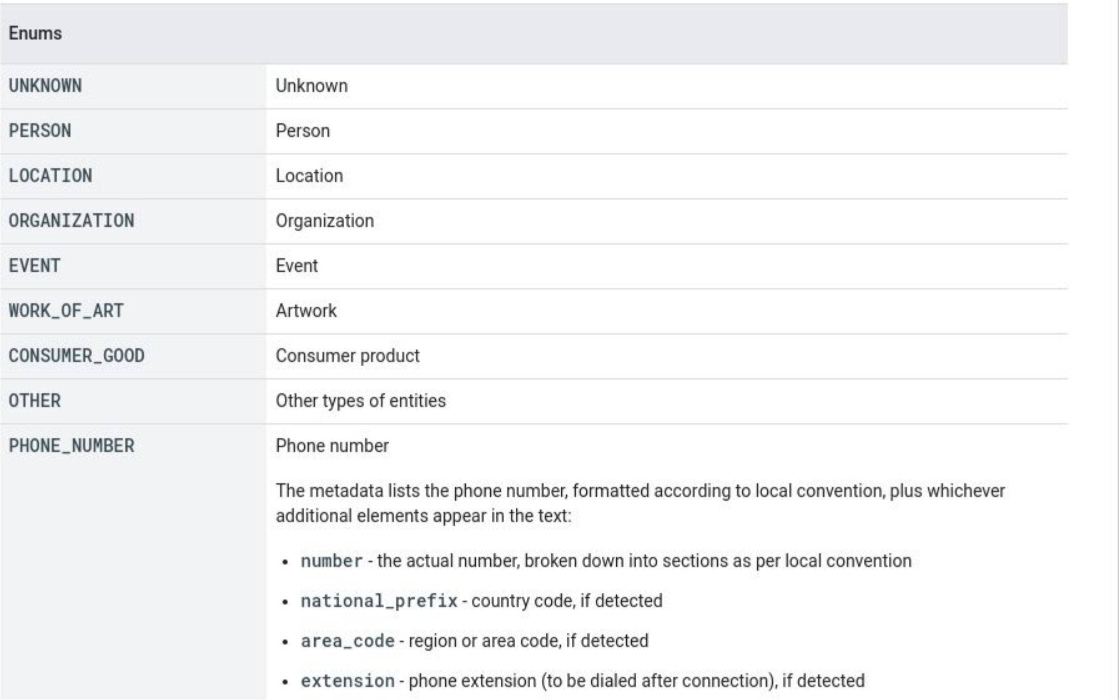


Sentiment	Sample Values
Clearly Positive*	"score": 0.8, "magnitude": 3.0
Clearly Negative*	"score": -0.6, "magnitude": 4.0
Neutral	"score": 0.1, "magnitude": 0.0
Mixed	"score": 0.0, "magnitude": 4.0

[1] https://cloud.google.com/natural-language/docs/setup

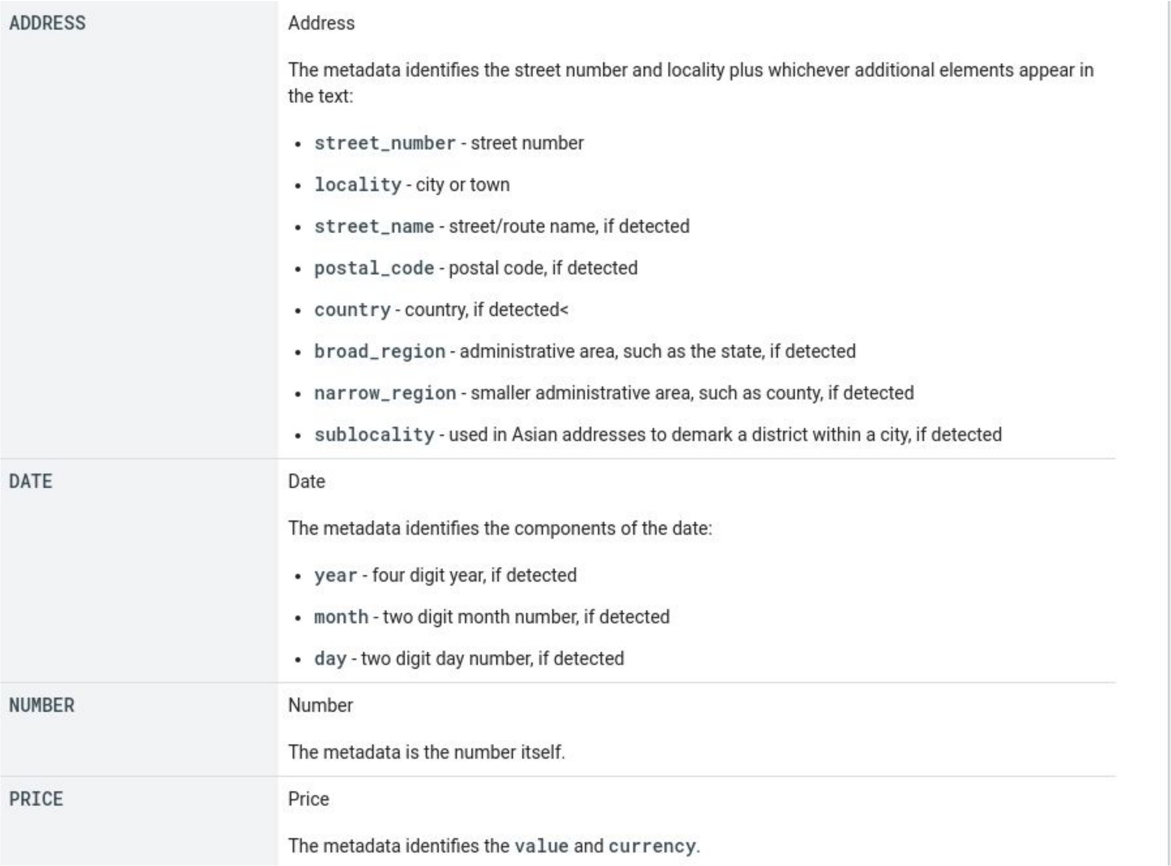
Analyzing Entities

This is also a very interesting feature provided by the Google Natural Language API. This feature does text processing and finds entities in the text such as public figures, landmarks, etc.



Enums	
`UNKNOWN`	Unknown
`PERSON`	Person
`LOCATION`	Location
`ORGANIZATION`	Organization
`EVENT`	Event
`WORK_OF_ART`	Artwork
`CONSUMER_GOOD`	Consumer product
`OTHER`	Other types of entities
`PHONE_NUMBER`	Phone number The metadata lists the phone number, formatted according to local convention, plus whichever additional elements appear in the text: • `number` - the actual number, broken down into sections as per local convention • `national_prefix` - country code, if detected • `area_code` - region or area code, if detected • `extension` - phone extension (to be dialed after connection), if detected

[2]



`ADDRESS`	Address The metadata identifies the street number and locality plus whichever additional elements appear in the text: • `street_number` - street number • `locality` - city or town • `street_name` - street/route name, if detected • `postal_code` - postal code, if detected • `country` - country, if detected< • `broad_region` - administrative area, such as the state, if detected • `narrow_region` - smaller administrative area, such as county, if detected • `sublocality` - used in Asian addresses to demark a district within a city, if detected
`DATE`	Date The metadata identifies the components of the date: • `year` - four digit year, if detected • `month` - two digit month number, if detected • `day` - two digit day number, if detected
`NUMBER`	Number The metadata is the number itself.
`PRICE`	Price The metadata identifies the `value` and `currency`.

The figures above show the entity data returned by the API. As you can see there's quite a bit of data and metadata associated with this API.

Analyzing Syntax

This feature allows us to modularize text into sentences and words. This feature specifically is in the field of morphology or "the internal structure of words". In the course we've learned about part of speech tagging which falls into this category as well. This feature includes things such as: tags, numbers, person, gender, case, mood, etc.

This feature compared to all the other features is probably the most difficult to implement and understand. This is because languages all have different syntactic structures and dependency trees. This feature use "a common set of dependencies that apply across the supported languages[3]"

Analyzing Entity Sentiment

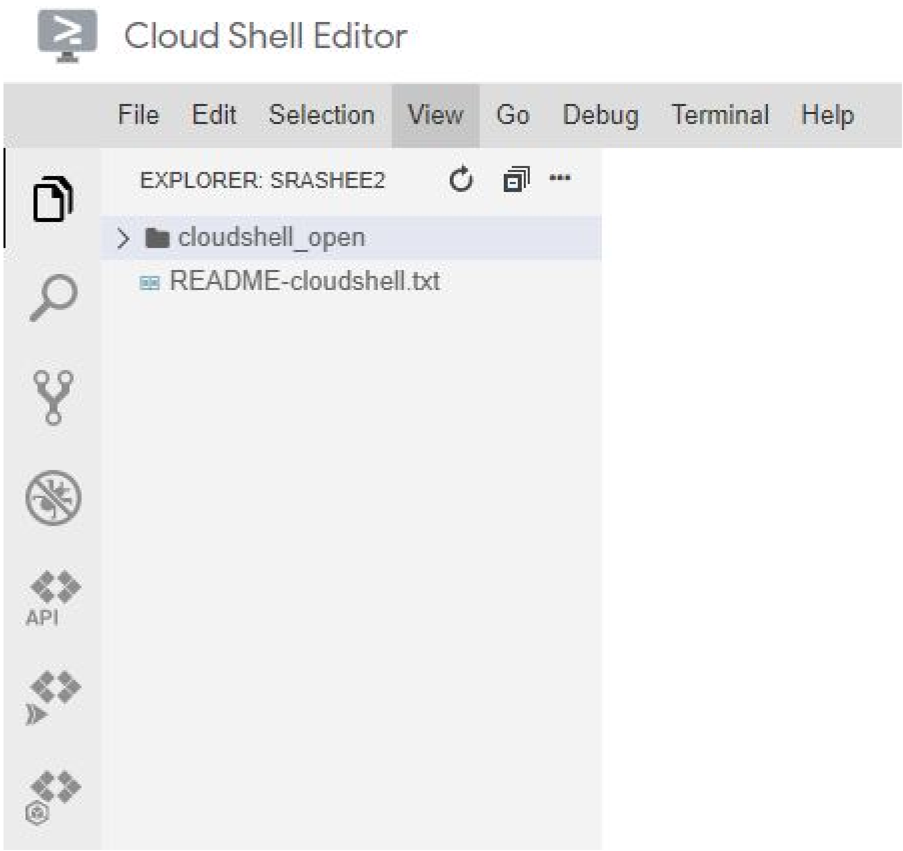
As the name suggests this combines the two features already provided with the API. This feature tries to gauge the sentiments on the extracted entities. Similarly with the ranking system of the other features, sentiments are represented numerically and those are associated with said entities. The same sentiment analysis figure above applies for this as well.

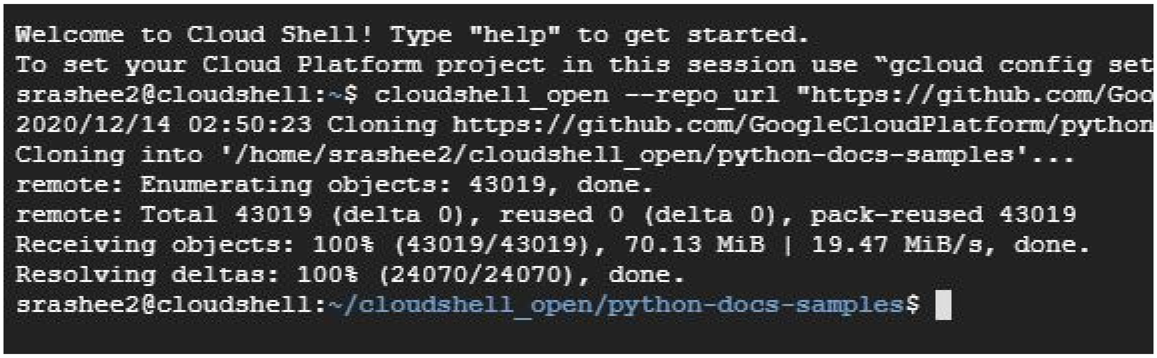
[3] https://cloud.google.com/natural-language/docs/morphology

Analysis

Cloud Shell Editor

In this section we dive into the features mentioned above and use the Google Cloud Shell editor. As seen below you can see Google’s Cloud Shell Editor. It’s a very convenient editing tool and gives you a shell in the browser!





```
Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use “gcloud config set
srashee2@cloudshell:~$ cloudshell_open --repo_url "https://github.com/Goo
2020/12/14 02:50:23 Cloning https://github.com/GoogleCloudPlatform/python
Cloning into '/home/srashee2/cloudshell_open/python-docs-samples'...
remote: Enumerating objects: 43019, done.
remote: Total 43019 (delta 0), reused 0 (delta 0), pack-reused 43019
Receiving objects: 100% (43019/43019), 70.13 MiB | 19.47 MiB/s, done.
Resolving deltas: 100% (24070/24070), done.
srashee2@cloudshell:~/cloudshell_open/python-docs-samples$ 
```

Inside the Cloud Shell Editor I will explore samples and write my own python scripts to test out the features of this framework. To begin I cloned down the Python Client for Google Cloud Natural Language[4]. There are a few steps to get this setup which can be found in the readme of the github. The following must be done to use this library.

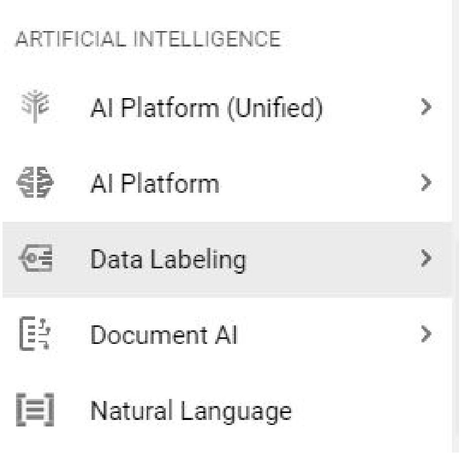
1. Select or create a cloud platform project
2. Enable billing for your project
3. Enable the Google Cloud Language API
4. Setup Authentication

Afterwards we can begin to use the framework. I then ran all the python examples listed under the client github page. I manually edited the document content to add other words to check that sentiment analysis is working properly. As most of the code and examples are very well established and completed, I haven't shared the code snippets here.

Overall Usage and Comments

The overall experience using this API wasn't too difficult but as with most of the cloud application stuff it's heavy on the startup side. To begin using the services required a free trial on my part, creating a project, enabling billing and the API and having proper authentication. This was also true for using the google cloud shell which should be a way for people to explore the framework without having to set up everything.

The billing and interface are very well designed and intuitive. Google offers quite a bit of services so it may be daunting at first to look at their google cloud console, but in general I think they have done a good job with organization.



[4] https://github.com/googleapis/python-language

Conclusion

The Google Natural Language API is an extremely powerful framework that allows anyone to quickly get a text processing framework without needing training data or needing to train a model yourself. We've explored quite a few features of the framework