Google Natural Language API

Vivek Bansal

(vivekb3@illinois.edu)

Table of Contents

Introduction

Natural Language processing (NLP) is now a household used feature. While, we may not be leveraging all features of NLP, but the common use cases are the Google Home assistant uses it to listen to the commands, transcribe the audio data and process to retrieve the answers. There are various other applications of NLP in the commercial space like auto chat assistants that has become extremely important for customer communication, reducing workload and infrastructure cost. Other feature would include filtering of fake news, hate content, review data etc. Especially, during this time of elections, we can see that Twitter and Facebook accounts are increasingly being scrutinized with the help of technology.

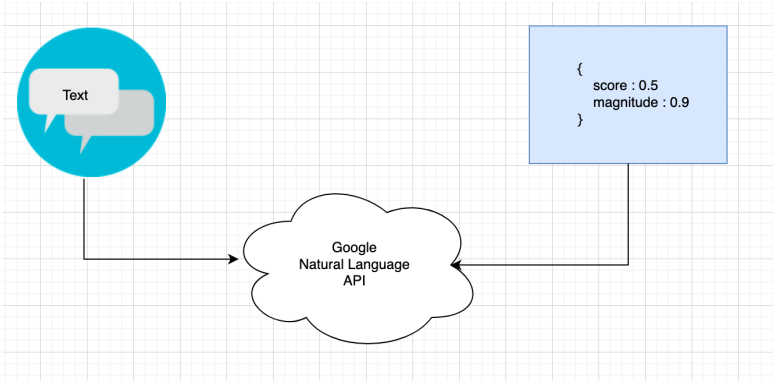
The Cloud Natural Language API attempts to simplify this so that you can use machine learning to process text content without keeping up with all the research papers. Like any machine learning API, the results are best guesses—treat the output as suggestions that may morph over time rather than absolute unquestionable facts. While this is being researched by various companies for commercial use and enhancements, I will focus on Google Natural Language API in this paper.

Google Natural Language API

The Natural Language API is a pretrained machine learning model that can analyze syntax, extract entities, and evaluate the sentiment of text through an easy to use interface. As these models have been trained on enormously large document corpuses, their performance is usually quite good as long as they are used on datasets that do not make use of a very idiosyncratic language.

The biggest advantage of using these pre-trained models via the API is, that no training dataset is needed. The API allows the user to immediately start making predictions, which can be very valuable in situations where little labeled data is available.

Similar to Google Cloud's other machine-learning APIs, the Natural Language API is a stateless API where you send it some input (in this case the input is text), and the API returns some set of annotations about the text.



The Natural Language API has several methods for performing analysis and annotation on your text. Each level of analysis provides valuable information for language understanding.

- Sentiment Analysis

- Entity Analysis
- Entity Sentiment Analysis
- Syntactic analysis
- Content classification

Sentiment Analysis

The syntax analysis service is mostly used early in one's pipeline to create features which are later fed into machine learning models. On the contrary, the sentiment analysis service can be used right out of the box.

Google's sentiment analysis will provide the prevailing emotional opinion within a provided text. The API returns two values: The "score" describes the emotional leaning of the text from -1 (negative) to +1 (positive), with 0 being neutral.

The "magnitude" measures the strength of the emotion.

Let's look at some examples:

Input Sentence	Sentiment Results	Interpretation
The train to London leaves at four o'clock	Score: 0.0 Magnitude: 0.0	A completely neutral statement, which doesn't contain any emotion at all.
This blog post is good.	Score: 0.7 Magnitude: 0.7	A positive sentiment, but not expressed very strongly.
This blog post is good. It was very helpful. The author is amazing.	Score: 0.7 Magnitude: 2.3	The same sentiment, but expressed much stronger.
This blog post is very good. This author is a horrible writer usually, but here he got lucky.	Score: 0.0 Magnitude: 1.6	The magnitude shows us that there are emotions expressed in this text, but the sentiment shows that they are mixed and not clearly positive or negative.

Google's sentiment analysis model is trained on a very large dataset.

Entity Analysis

Entity Analysis is the process of detecting known entities like public figures or landmarks from a given text. Entity detection is very helpful for all kinds of classification and topic modeling tasks. Common nouns such as restaurant, stadium, and so on.) and returns information about those entities. Entity analysis is performed with the `analyzeEntities` method.

The Google Natural Language API provides some basic information about each detected entity and even provides a link to the respective Wikipedia article if it exists. Also, a salience score is calculated. This score for an entity provides information about the importance or centrality of that entity to the entire document text. Scores closer to 0 are less salient, while scores closer to 1.0 are highly salient.

When we send a request to the API with this example sentence: *"Robert DeNiro spoke to Martin Scorsese in Hollywood on Christmas Eve in December 2011."* We receive the following result:

Detected Entity	Additional Information
Robert De Niro	type : PERSON salience : 0.5869118 wikipedia_url : https://en.wikipedia.org/wiki/Robert_De_Niro
Hollywood	type : LOCATION salience : 0.17918482 wikipedia_url : https://en.wikipedia.org/wiki/Hollywood
Martin Scorsese	type : LOCATION salience : 0.17712952 wikipedia_url : https://en.wikipedia.org/wiki/Martin_Scorsese
Christmas Eve	type : PERSON salience : 0.056773853 wikipedia_url : https://en.wikipedia.org/wiki/Christmas
December 2011	type : DATE Year: 2011 Month: 12 salience : 0.0 wikipedia_url : -
2011	type : NUMBER salience : 0.0 wikipedia_url : -

As you can see, all entities are identified and classified correctly, except that 2011 appears twice. Additionally, to the field in the example output, the entity analysis API will also detect organizations, works of art, consumer goods, phone numbers, addresses, and prices.

Entity Sentiment Analysis

Entity sentiment analysis inspects the given text for known entities (proper nouns and common nouns), returns information about those entities, and identifies the prevailing emotional opinion of the entity within the text, especially to determine a writer's attitude toward the entity as

positive, negative, or neutral. Entity analysis is performed with the `analyzeEntitySentiment` method.

If there are models for entity detection and sentiment analysis, it's only natural to go a step further and combine them to detect the prevailing emotions towards the different entities in a text.

While the Sentiment Analysis API finds all displays of emotion in the document and aggregates them, the Entity Sentiment Analysis tries to find the dependencies between different parts of the document and the identified entities and then attributes the emotions in these text segments to the respective entities.

For example, the opinionated text: *"The author is a horrible writer. The reader is very intelligent on the other hand."* leads to the results:

Entity	Sentiment
author	Salience: 0.8773350715637207 Sentiment: magnitude: 1.899999976158142 score: -0.8999999761581421
reader	Salience: 0.08653714507818222 Sentiment: magnitude: 0.8999999761581421 score: 0.8999999761581421

The entity sentiment analysis so far works only for English, Japanese, and Spanish.

Syntactic Analysis

The Natural Language API provides a powerful set of tools for analyzing and parsing text through syntactic analysis. It breaks up the given text into a series of sentences and tokens (generally, word boundaries), providing further analysis on those tokens. To perform syntactic analysis, use the `analyzeSyntax` method.

Syntactic Analysis consists of the following operations:

- Sentence extraction breaks up the stream of text into a series of sentences.
- *Tokenization* breaks the stream of text up into a series of tokens, with each token usually corresponding to a single word.
- The Natural Language API then processes the tokens and, using their locations within sentences, adds syntactic information to the tokens.

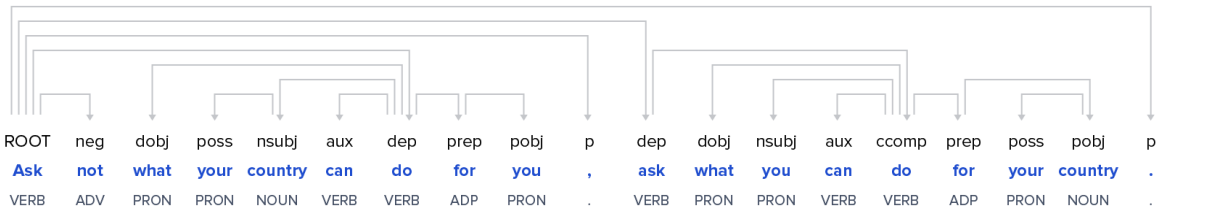
For a given text, Google's syntax analysis will return a breakdown of all words with a rich set of linguistic information for each token. The information can be divided into two parts:

Part of speech: This part contains information about the morphology of each token. For each word, a fine-grained analysis is returned containing its type (noun, verb, etc.), gender, grammatical case, tense, grammatical mood, grammatical voice, and much more.

For example, for the input sentence *"A computer once beat me at chess, but it was no match for me at kickboxing."* (Emo Philips) the part-of-speech analysis is:

A	tag: DET
'computer'	tag: NOUN number: SINGULAR
'once'	tag: ADV
'beat'	tag: VERB mood: INDICATIVE tense: PAST
'me'	tag: PRON case: ACCUSATIVE number: SINGULAR person: FIRST
at	tag: ADP
'chess'	tag: NOUN number: SINGULAR
','	tag: PUNCT
'but'	tag: CONJ
'it'	tag: PRON case: NOMINATIVE gender: NEUTER number: SINGULAR person: THIRD
'was'	tag: VERB mood: INDICATIVE number: SINGULAR person: THIRD tense: PAST
'no'	tag: DET
'match'	tag: NOUN number: SINGULAR
'for'	tag: ADP
'kick'	tag: NOUN number: SINGULAR
'boxing'	tag: NOUN number: SINGULAR
'.'	tag: PUNCT

Dependency trees: The second part of the return is called a dependency tree, which describes the syntactic structure of each sentence. The following diagram of a famous Kennedy quote shows such a dependency tree. For each word, the arrows indicate which words are modified by it.



The commonly used Python libraries nltk and spaCy contain similar functionalities. The quality of the analysis is consistently high across all three options, but the Google Natural Language API is easier to use. The above analysis can be obtained with very few lines of code (see example

further down). However, while spaCy and nltk are open-source and therefore free, the usage of the Google Natural Language API costs money after a certain number of free requests (see cost section).

Apart from English, the syntactic analysis supports ten additional languages: *Chinese (Simplified), Chinese (Traditional), French, German, Italian, Japanese, Korean, Portuguese, Russian, and Spanish*.

Content Classification

Lastly, the Google Natural language API comes with a plug-and-play content classification model. It analyzes text content and returns a content category for the content. Content classification is performed by using the `classifyText` method.

The model is trained to classify the input documents into a large set of categories. The categories are structured hierarchical, e.g. the Category *"Hobbies & Leisure"* has several sub-categories, one of which would be *"Hobbies & Leisure/Outdoors"* which itself has sub-categories like *"Hobbies & Leisure/Outdoors/Fishing."*

This is an example text from a Nikon camera ad:

"The D3500's large 24.2 MP DX-format sensor captures richly detailed photos and Full HD movies—even when you shoot in low light. Combined with the rendering power of your NIKKOR lens, you can start creating artistic portraits with smooth background blur. With ease."

The Google API returns the result:

Category	Confidence
Arts & Entertainment/Visual Art & Design/Photographic & Digital Arts	0.95
Hobbies & Leisure	0.94
Computers & Electronics/Consumer Electronics/Camera & Photo Equipment	0.85

All three of these categories make sense, even though we would intuitively rank the third entry higher than the second one. However, one must consider that this input segment is only a short part of the full camera ad document and the classification model's performance improves with text length.

After trying it out with a lot of documents, I found the results of the classification model meaningful in most cases. Still, as all other models from the Google Natural Language API, the classifier comes as a black-box solution which cannot be modified or even fine-tuned by the API user. Especially in the case of text classification, the vast majority of companies will have their own text-categories that differ from the categories of the Google model and therefore, the

Natural Language API text classification service might not be applicable for the majority of the users.

Another limitation of the classification model is that it only works for English language texts.

Setup the Natural Language

Following steps needs to be performed to enable Google Natural Language API. For more details, please go through Google Cloud documentation for the product.

- Create a project
- Enable billing
- Enable the API
- Set up authentication
- Create a service account and download the private key file
- Use the service account key file in your environment
- Install and initialize the Cloud SDK
- Test the SDK and authentication
- Install the Natural Language client library

How to Use the Natural Language API

The major advantage of the Google Natural Language API is its ease of use. No machine learning skills are required and almost no coding skills. On the Google Cloud website, you can find code snippets for calling the API for a lot of languages.

For example, the Python code to call the sentiment analysis API is as short as:

```
from google.cloud import language_v1

from google.cloud.language_v1 import enums

import six

def sample_analyze_sentiment(content):

    client = language_v1.LanguageServiceClient()

    if isinstance(content, six.binary_type):

        content = content.decode('utf-8')

    type_ = enums.Document.Type.PLAIN_TEXT

    document = {'type': type_, 'content': content}

    response = client.analyze_sentiment(document)

    sentiment = response.document_sentiment

    print('Score: {}'.format(sentiment.score))

    print('Magnitude: {}'.format(sentiment.magnitude))
```

The other API functionalities are called in a similar way, simply by changing `client.analyze_sentiment` to the appropriate function.

Pricing

The Natural Language provides a set of features for analyzing unstructured text. You pay only for the features you use with no upfront commitments. The API supports the following features:

Feature Type	Description
Entity Analysis	Identify entities and label by types such as person, organization, location, events, products and media.
Sentiment Analysis	Understand the overall sentiment expressed in a block of text.
Entity Sentiment Analysis	Understand the sentiment for entities identified in a block of text.
Syntax Analysis	Extract tokens and sentences, identify parts of speech (PoS) and create dependency parse trees for each sentence.
Content Classification	Identify content categories that apply to a block of text.

Pricing units

Your usage of the Natural Language is calculated in terms of "units," where each document sent to the API for analysis is at least one unit. Documents that have more than 1,000 Unicode characters (including whitespace characters and any markup characters such as HTML or XML tags) are considered as multiple units, one unit per 1,000 characters.

For example, if you send three requests to the Natural Language that contain 800, 1,500, and 600 characters respectively, you are charged for four units: one for the first request (800), two for the second request (1,500), and one for the third request (600).

Prices for usage of the Natural Language are computed monthly based on which feature of the API you used, and how many units are evaluated using those features. The table below provides the price per 1,000 units based on the total number of units analyzed during the billing month.

Monthly prices

Feature	0 - 5K	5K+ - 1M	1M+ - 5M	5M+ - 20M
Entity Analysis	Free	$1.00	$0.50	$0.25
Sentiment Analysis	Free	$1.00	$0.50	$0.25
Syntax Analysis	Free	$0.50	$0.25	$0.125
Entity Sentiment Analysis	Free	$2.00	$1.00	$0.50

Feature	0 - 5K	5K+ - 1M	1M+ - 5M	5M+ - 20M
Feature	**0 - 30K**	**30K+ - 250K**	**250K+ - 5M**	**5M+**
Content Classification	Free	$2.00	$0.50	$0.10

If you pay in a currency other than USD, the prices listed in your currency on Cloud Platform SKUs apply.

Conclusion:

The Google Cloud Natural Language API enables you to leverage Google's machine learning technologies and extract beneficial insights from unstructured text.
The API allows you to perform entity recognition, sentiment analysis, syntax analysis, and content classification in 700+ predefined categories. It also allows you to perform text analysis in multiple languages, including English, Portuguese, French, Chinese, and German.
The API can assist you to derive useful information from text documents, blog articles, or other types of content. You can use it to carry out sentiment analysis about your products on web platforms, scrutinize text uploaded to your web portal, or derive meaning from audio conversations. The Google Cloud Natural Language API is priced according to the number of units you consume (every document analyzed by the API constitutes at least one unit) and the feature of the API you use. Processing less than 5,000 units per month is free—for any of the features. Google Cloud NLP API are easy to use and have good documentation.

References:

https://cloud.google.com/natural-language/docs/

https://opensource.com/article/19/7/python-google-natural-language-api