Name: Sanjeev Singh

Email: sksingh6@illinois.edu

University of Illinois-Urbana Champaign

CS-410-Fall 2022

Tech Review

Google Natural Language API

The Google Natural Language API is an easy to use interface to a set of powerful NLP models, which have been pre-trained by Google to perform various tasks. 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 Natural Language API comprises five different services:

1. Syntactic analysis
2. Sentiment Analysis
3. Entity Analysis
4. Entity Sentiment Analysis
5. Text Classification

Syntactic Analysis extracts linguistic information, breaking up the given text into a series of sentences and tokens (generally, word boundaries), providing further analysis on those 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.

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

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.

Entity Analysis inspects the given text for known entities (Proper nouns such as public figures, landmarks, and so on. Common nouns such as restaurant, stadium, and so on.) and returns information about those entities. Entity detection is very helpful for all kinds of classification and topic modeling tasks.

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

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

Text Classification

Lastly, the Google Natural language API comes with a plug-and-play text classification model.

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."*

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.

Convenient, but Inflexible

The Google Natural Language API is a very convenient option for quick, out-of-the-box solutions. Very little technical knowledge and no understanding of the underlying machine learning models is required.

The main disadvantage is its inflexibility and the lack of access to the models. The models cannot be tuned to a specific task or dataset.

In a real-world environment, most tasks will probably require a more tailored solution than the standardized Natural Language API functions can provide.

For this scenario, Google AutoML Natural Language is more suitable.

Google AutoML Natural Language

If the Natural Language API is not flexible enough for your business purposes, then AutoML Natural Language might be the right service. AutoML is a new Google Cloud Service (still in beta) that enables the user to create customized machine learning models. In contrast to the Natural Language API, the AutoML models will be trained on the user's data and therefore fit a specific task.

Custom machine learning models for classifying content are useful when the predefined categories that are available from the Natural Language API are too generic or not applicable to your specific use case or knowledge domain.

The AutoML service requires a bit more effort for the user, mainly because you have to provide a dataset to train the model. However, the training and evaluation of the models in completely automated and no machine learning knowledge is required. The whole process can be done without writing any code by using the Google Cloud console. Of course, if you want to automate these steps, there is support for all common programming languages.

What Can Be Done With Google AutoML Natural Language?

The AutoML service covers three use cases. All of these use cases support solely the English language for now.

1. AutoML Text Classification

While the text classifier of the Natural Language API is pre-trained and therefore has a fixed set of text categories, the AutoML text classification builds customized machine learning models, with the categories that you provide in your training dataset.

2. AutoML Sentiment Analysis

As we have seen, the sentiment analysis of the Natural Language API works great in general use cases like movie reviews. Because the sentiment model is trained on a very general corpus, the performance can deteriorate for documents that use a lot of domain-specific language. In these situations, the AutoML Sentiment Analysis allows you to train a sentiment model that is customized to your domain.

3. AutoML Entity Extraction

In many business contexts, there are domain specific entities (legal contracts, medical documents) that the Natural Language API will not be able to identify. If you have a dataset where the entities are marked, you can train a customized model entity extractor with AutoML. If the dataset is sufficiently big, the trained entity extraction model will also be able to detect previously unseen entities.

How to Use AutoML Natural Language

Using the three AutoML is a four-step process and is very similar for all three methodologies:

Dataset Preparation
The dataset has to be in a specific format (CSV or JSON) and needs to be stored in a storage bucket. For classification and sentiment models, the datasets contain just two columns, the text and the label. For the entity extraction model, the dataset needs the text and the locations of all entities in the text.

Model Training
The model training is completely automatic. If no instructions are given otherwise, then AutoML will split the training set automatically into train, test and validation sets. This split can also be decided by the user, but that is the only way to influence the model training. The rest of the training is completely automated in a black-box fashion.

Evaluation
When the training is finished, AutoML will display precision and recall scores as well as a confusion matrix. Unfortunately, there is absolutely no information about the model itself, making it difficult to identify the reasons for bad performing models.

Prediction
Once you are happy with the model's performance, the model can be conveniently deployed with a couple of clicks. The deployment process takes only a few minutes.

Ref: Looking for Meaning - A Google NLP Tutorial | Toptal

Cloud Natural Language documentation | Cloud Natural Language API | Google Cloud