Start Lab 01:07:30

Using the Natural Language API from Google Docs

Lab 45 minutes No cost Intermediate

Rate Lab

GSP126

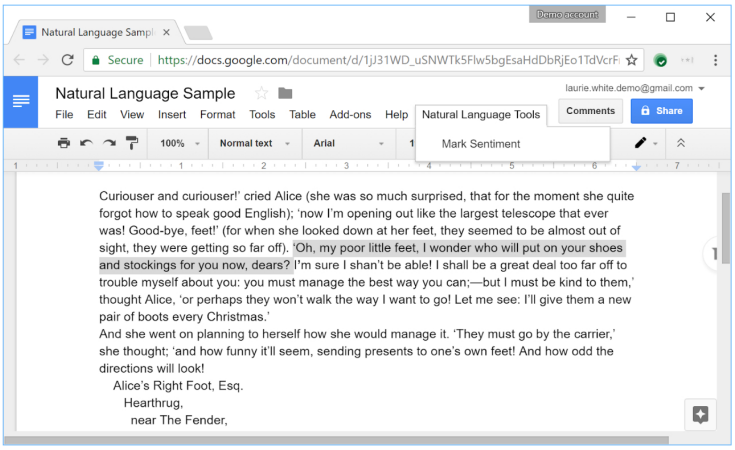
Google Cloud Self-Paced Labs

Overview

The Natural Language API is a pretrained machine learning model that can analyze syntax, extract entities, and evaluate the sentiment of text. You can call the Natural Language API from Google Docs to perform all of these functions.

This lab focuses on calling the Natural Language API from Google Docs. You use the Natural Language API to recognize the sentiment of selected text in a Google Doc and highlight it based on that sentiment.

When you complete this lab, you are able to select text in a document and mark its sentiment, using a menu choice, as shown below.



Text is highlighted in red for negative sentiment, green for positive sentiment, and yellow for neutral sentiment.

What you'll learn

In this lab, you learn how to:

In this lab, you learn how to:

- Call the Natural Language API from Google Docs
- Add menus to Google Docs
- Recognize and work with selected text in Google Docs

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

> **Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

- Time to complete the lab---remember, once you start, you cannot pause a lab.

> **Note:** If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

How to start your lab and sign in to the Google Cloud console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is the **Lab Details** panel with the following:
    - The **Open Google Cloud console** button
    - Time remaining
    - The temporary credentials that you must use for this lab
    - Other information, if needed, to step through this lab
2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

    The lab spins up resources, and then opens another tab that shows the **Sign in** page.

    Tip: Arrange the tabs in separate windows, side-by-side.

    > **Note:** If you see the **Choose an account** dialog, click **Use Another Account**.

3. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

```
"Username"
```

You can also find the **Username** in the **Lab Details** panel.

4. Click **Next**.
5. Copy the **Password** below and paste it into the **Welcome** dialog.

```
"Password"
```

You can also find the **Password** in the **Lab Details** panel.

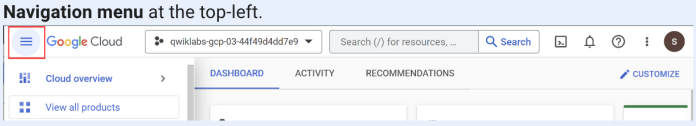
6. Click **Next**.

> **Important:** You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.

> **Note:** Using your own Google Cloud account for this lab may incur extra charges.

7. Click through the subsequent pages:
    - Accept the terms and conditions.
    - Do not add recovery options or two-factor authentication (because this is a temporary account).
    - Do not sign up for free trials.

After a few moments, the Google Cloud console opens in this tab.

> **Note:** To view a menu with a list of Google Cloud products and services, click the **Navigation menu** at the top-left.
>
> 
>

Task 1. Enable the Natural Language API

Before you start, make sure that the Natural Language API is enabled.

1. In the Google Cloud console, select **Navigation menu > APIs & Services > Library**.
2. Search for **Cloud Natural Language API** and click on the API to enable it or to confirm that the API is enabled.

Task 2. Get an API key

Generate an API user key to pass in the request URL.

1. To create an API key, select **Navigation menu > APIs & Services > Credentials**

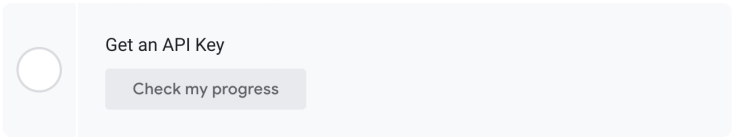
1. To create an API key, select **Navigation menu > APIs & Services > Credentials**.

2. Click **Create credentials** at the top and select **API key**:

3. Copy the API key to a text file or a Google Doc to use in a later step. Click **Close**.

Once you have the API key, you are ready to move into Google Docs.

Click *Check my progress* to verify the objective.



Task 3. Set up your Google Doc

Before you call the Natural Language API, make an Apps Script program to create the menu, link it to a function to mark the text, and extract the text from the user selection.

1. Create a new Google Doc.

2. From within your new document, select the menu item **Extensions > Apps Script**.

3. Delete any code in the script editor and paste in the code below. This code creates a menu item, extracts the text from the current selected text, and highlights the text based on its sentiment. It does not call the Natural Language API yet.

```
/**
* @OnlyCurrentDoc
*
* The above comment directs Apps Script to limit the scope of
file
* access for this add-on. It specifies that this add-on will
only
* attempt to read or modify the files in which the add-on is
used,
* and not all of the user's files. The authorization request
message
* presented to users will reflect this limited scope.
*/

/**
* Creates a menu entry in the Google Docs UI when the document
is
* opened.
*
*/
function onOpen() {
  var ui = DocumentApp.getUi();
  ui.createMenu('Natural Language Tools')
    .addItem('Mark Sentiment', 'markSentiment')
    .addToUi();
}
/**
* Gets the user-selected text and highlights it based on
sentiment
* with green for positive sentiment, red for negative, and
yellow
* for neutral.
*
*/
function markSentiment() {
  var POSITIVE_COLOR = '#00ff00';  // Colors for sentiments
```

```
  var NEGATIVE_COLOR = '#ff0000';
  var NEUTRAL_COLOR = '#ffff00';
  var NEGATIVE_CUTOFF = -0.2;   //  Thresholds for sentiments
  var POSITIVE_CUTOFF = 0.2;

  var selection =
DocumentApp.getActiveDocument().getSelection();
  if (selection) {
    var string = getSelectedText();

    var sentiment = retrieveSentiment(string);

    //  Select the appropriate color
    var color = NEUTRAL_COLOR;
    if (sentiment <= NEGATIVE_CUTOFF) {
      color = NEGATIVE_COLOR;
    }
    if (sentiment >= POSITIVE_CUTOFF) {
      color = POSITIVE_COLOR;
    }

    //  Highlight the text
    var elements = selection.getSelectedElements();
    for (var i = 0; i < elements.length; i++) {
      if (elements[i].isPartial()) {
        var element = elements[i].getElement().editAsText();
        var startIndex = elements[i].getStartOffset();
        var endIndex = elements[i].getEndOffsetInclusive();
        element.setBackgroundColor(startIndex, endIndex, color);

      } else {
        var element = elements[i].getElement().editAsText();
        foundText = elements[i].getElement().editAsText();
        foundText.setBackgroundColor(color);
      }
    }
  }
}
/**
 * Returns a string with the contents of the selected text.
 * If no text is selected, returns an empty string.
 */
function getSelectedText() {
  var selection =
DocumentApp.getActiveDocument().getSelection();
  var string = "";
  if (selection) {
    var elements = selection.getSelectedElements();

    for (var i = 0; i < elements.length; i++) {
      if (elements[i].isPartial()) {
        var element = elements[i].getElement().asText();
        var startIndex = elements[i].getStartOffset();
        var endIndex = elements[i].getEndOffsetInclusive() + 1;
        var text = element.getText().substring(startIndex,
endIndex);
        string = string + text;

      } else {
        var element = elements[i].getElement();
        // Only translate elements that can be edited as text;
skip
        // images and other non-text elements.
        if (element.editAsText) {
          string = string + element.asText().getText();
        }
      }
    }
  }
  return string;
}

/** Given a string, will call the Natural Language API and
```

```
retrieve
  * the sentiment of the string.  The sentiment will be a real
  * number in the range -1 to 1, where -1 is highly negative
  * sentiment and 1 is highly positive.
*/
function retrieveSentiment (line) {
//  TODO:  Call the Natural Language API with the line given
//         and return the sentiment value.
  return 0.0;
}
```

Note: To learn more about Apps Script, refer to the Google Apps Script reference.

4. On the menu bar, click **Save project** (). (The script's name is shown to end users in several places, including the authorization dialog.)

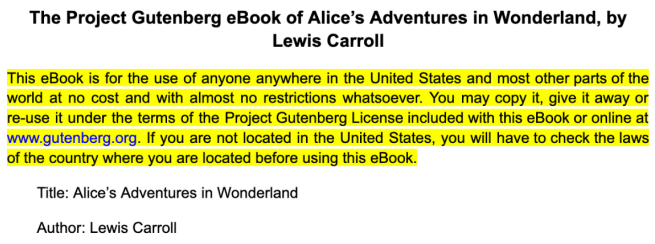
5. Return to your document. Add text to your document. You can use the sample that comes from Alice in Wonderland on Project Gutenberg (copy and paste the `Plain Text UTF-8` version into the document), but feel free to use any text you wish.

6. Reload the document to see the new menu, **Natural Language Tools**, which you created, appear in the Google Docs toolbar.

7. Select text and then the **Mark Sentiment** option from the Natural Language Tools menu. The first time you select this option, you are prompted to authorize the script to run. Click **Continue**, and then confirm your account.

8. **Allow** Natural Language Tools to view and manage documents that this application has been installed in.

9. Once the script is authorized, the selected text is highlighted in yellow, since the stub for sentiment analysis always returns 0.0, which is neutral.



The Project Gutenberg eBook of Alice's Adventures in Wonderland, by Lewis Carroll

This eBook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.org. If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Alice's Adventures in Wonderland

Author: Lewis Carroll

Click *Check my progress* to verify the objective.

Set up your Google Doc

Check my progress

Task 4. Call the Natural Language API

Once your program can extract text from the selection and highlight it, it's time to call the Natural Language API. All of this is done in the body of the `retrieveSentiment` function.

Note: To learn more about the Natural Language API, refer to the Cloud Natural Language API reference.

1. Return to the **Extensions** > **Apps Script** in Google Docs.

2. In the `retrieveSentiment` function, remove the current lines and add a variable to contain your API key, which you saved in the "Get an API key" section:

```
var apiKey = "your key here";
```

3. Create a variable to hold the URL of the Natural Language API with your API key appended to it:

```
 var apiEndpoint =
'https://language.googleapis.com/v1/documents:analyzeSentiment?
key='
+ apiKey;
```

4. Build a structure from the line passed into the function that holds the text of the line, along with its type and language. Currently the only supported language is English.

```
var docDetails = {
  language: 'en-us',
  type: 'PLAIN_TEXT',
  content: line
};
```

5. Build the entire data payload from the document details by adding the encoding type:

```
var nlData = {
  document: docDetails,
  encodingType: 'UTF8'
};
```

6. Create a structure containing the payload and the necessary header information:

```
var nlOptions = {
  method : 'post',
  contentType: 'application/json',
  payload : JSON.stringify(nlData)
};
```

7. Make the call, saving the response:

```
var response = UrlFetchApp.fetch(apiEndpoint, nlOptions);
```

8. The response is returned in JSON format; parse it and extract the score field, if it exists. Return either that field or 0.0.

```
var data = JSON.parse(response);

var sentiment = 0.0;
//  Ensure all pieces were in the returned value
if (data && data.documentSentiment
       && data.documentSentiment.score){
   sentiment = data.documentSentiment.score;
}

return sentiment;
```

The complete code to retrieve the sentiment is below:

```
function retrieveSentiment (line) {
  var apiKey = "your key here";
  var apiEndpoint =
'https://language.googleapis.com/v1/documents:analyzeSentiment?
key='
+ apiKey;

  //  Create a structure with the text, its language, its type,
  //  and its encoding
  var docDetails = {
    language: 'en-us',
    type: 'PLAIN_TEXT',
    content: line
  };

  var nlData = {
    document: docDetails,
    encodingType: 'UTF8'
  };

  //  Package all of the options and the data together for the
call
  var nlOptions = {
    method : 'post',
    contentType: 'application/json',
    payload : JSON.stringify(nlData)
  };

  //  And make the call
  var response = UrlFetchApp.fetch(apiEndpoint, nlOptions);

  var data = JSON.parse(response);

  var sentiment = 0.0;
  //  Ensure all pieces were in the returned value
  if (data && data.documentSentiment
          && data.documentSentiment.score){
     sentiment = data.documentSentiment.score;
  }

  return sentiment;
}
```

9. Save your script, reload the document, and try out the full program. You may need to re-authorize with your credentials to enable the new functionality. Select different sections of your document to see how the sentiment may differ over its parts.

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, “and what is the use of a book,” thought Alice “without pictures or conversations?"

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

10. (Optional) Type and then analyze your own words. For example, type and analyze "I'm mad", and then type and analyze "I'm happy". Experiment to see how the Natural Language API interprets different groups, for example if you analyze "I'm happy. I'm happy. I'm sad.". What happens if you add another "I'm sad."?

Congratulations!

You created a Google Doc and called the Natural Language API to analyze the sentiment of selected portions of the document.

Finish your quest

This self-paced lab is part of the Workspace Integrations quest. A quest is a series of related labs that form a learning path. Completing this quest earns you a badge to recognize your achievement. You can make your badge or badges public and link to them in your online resume or social media account. Enroll in this quest and get immediate completion credit. Refer to the Google Cloud Skills Boost catalog for all available quests.

Take your next lab

Continue your quest with Build a Complete Database Web App with App Maker, or check out The Apps Script CLI - clasp.

Next steps / Learn more

Continue your Google Cloud learning with these suggestions:

- Take more labs. Learn more about the Natural Language API by taking labs, like Entity and Sentiment analysis with the Natural Language API. Or take something completely different like Rent-a-VM to Process Earthquake Data.
- Start a quest. A quest is a series of related labs that form a learning path. Completing a quest earns you a digital badge, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Refer to the Google Cloud Skills Boost catalog for all available quests.

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated June 14, 2023

Lab Last Tested June 14, 2023