```
##############################################################################
### STEP 1: import API
##############################################################################

pip install nltk==3.4.5
```

```
  File "<ipython-input-1-4d8f0ff9c0c3>", line 2
    pip install nltk==3.4.5
        ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
##############################################################################
### STEP 2: donwload libraries
##############################################################################
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
##############################################################################
### STEP 3: Get the frequency and the lines (documents) where appear a term
##############################################################################

## import libreries
import csv
import pandas as pd
import numpy as np
import gspread
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from gspread_dataframe import get_as_dataframe, set_with_dataframe
tokenizer = RegexpTokenizer(r'\w+')

################## Settings to use Google Sheet ##################
from google.colab import auth
```

```
from oauth2client.client import GoogleCredentials
qc = gspread.authorize(GoogleCredentials.get_application_default())

## Setting to analyze texts
Lem = WordNetLemmatizer()
ps = PorterStemmer()
stemmed_words = []

#### Set the URL of the database
agroecology_database = qc.open_by_url('https://docs.google.com/spreadsheets/d/1GI2C
sheet1 = agroecology_database.worksheet('Not Engaged') # Write the name of the shee
agroecology_dataframe = get_as_dataframe(sheet1) # Convert to dataframe
agroecology_dataframe.replace(["None", "NaN", np.nan], "and", inplace=True)

dimension = len(agroecology_dataframe) ## how many records has the database

############## TEXT PRE-PROCESSING ##############
def preprocessing(words):
  tokenizer = RegexpTokenizer(r'\w+')
  words = words.lower() # convert to lowercase
  tokenized_words = tokenizer.tokenize(words) # remove punctuations
  lem_words = [Lem.lemmatize(x) for x in tokenized_words] # convert plural to singu
  stop_words = set(stopwords.words('english')) # set language to english to remove
  filtered_words = [w for w in lem_words if not w in stop_words] # remove stop word
  return filtered_words

############### APPLY preprocessing FUNCTION TO ALL DATA ##############
all_words = []
for x in range(len(agroecology_dataframe.index)): ## for statement go through all r
  ## apply "preprocessing" function to each record and save the preprocesed text in
  all_words = all_words + (preprocessing(agroecology_dataframe["Title"][x])) + (pre

## Remove numbers of all database ##
final_list = [x for x in all_words if not (x.isdigit()
                                           or x[0] == '-' and x[1:].isdigit())]

## Calculate the words frequency ##
frequency = {}
for i in final_list: # go through of each analyzed text
  if i not in frequency.keys(): # if the word i is not as a key, add i as a keyword
    frequency[i] = 0
  frequency[i] = frequency[i]+1 # sum each occurrence of the word

## Convert to dataframe and export to excel ##
df = pd.DataFrame(data=frequency, index=["Frequency","Documents"])
df.columns.names = ['Terms']
df = (df.T)

#### Go throw the database and count the numbers of references where appear each te
for index in frequency:
```

```
  df1 = agroecology_dataframe[agroecology_dataframe['Title'].str.contains(index) |
  counter = len(df1)
  df['Documents'][index] = counter

### Calculate the percentage
total_frequency = df['Frequency'].sum()
df['PF'] = df['Frequency'] / total_frequency
df['PD'] = df['Documents'] / dimension

## Export to Excel
df.to_excel('not engaged frequency and documents 400Random.xlsx')


################################################################################
### STEP 4: Get the root of each term and group by these terms
################################################################################

# import libreries
import csv
import pandas as pd
import numpy as np
import gspread
import nltk
from nltk.stem import PorterStemmer
from gspread_dataframe import get_as_dataframe, set_with_dataframe

################## Settings to use Google Sheet ##################
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
qc = gspread.authorize(GoogleCredentials.get_application_default())
### STEM
ps = PorterStemmer()
stemmed_words = []

#### use the RESULT of previous script, read the Google Sheet file
database = qc.open_by_url('https://docs.google.com/spreadsheets/d/1j4-2wXSYs6heqCpI
sheet1 = database.worksheet('Sheet1') # Write the name of the sheet
dataframe = get_as_dataframe(sheet1) # Convert to dataframe
dataframe['Terms'] = ''
size = len(dataframe)

## Apply stemming to get the root of all words
for x in range(size):
  dataframe['Terms'][x] = ps.stem(dataframe['Words'][x])

## Delete column Terms and group by stemmed words
dataframe = dataframe.drop(columns=['Words'])
dataframe = dataframe.groupby(['Terms']).sum()

## Export to excel
```

```
## Export to excel
dataframe.to_excel('engaged frequency and documents - stemming.xlsx')
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:26: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
```

```
################################################################################
### STEP 5: Calculate the ratio of engaged / not engaged
################################################################################

### Require: the sheet with de percentages
import csv
import pandas as pd
import gspread
from gspread_dataframe import get_as_dataframe, set_with_dataframe
################### Settings to use Google Sheet ###################
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
qc = gspread.authorize(GoogleCredentials.get_application_default())
############## SET THE URL OF THE DATABASE ######################
# First dataset (engaged)
dataset = qc.open_by_url('https://docs.google.com/spreadsheets/d/1Ds1X_Y1GmWv1cR74a
sheet1 = dataset.worksheet('engaged') # Write the name of the sheet
df1 = get_as_dataframe(sheet1) # e document
sheet2 = dataset.worksheet('not engaged') # Write the name of the sheet
df2 = get_as_dataframe(sheet2) # ne document


#Create dataframes
df = pd.merge(df1, df2, on='Terms', how='outer')

df['RatioF'] = df['PEF']/df['PNEF']
df['RatioD'] = df['PED']/df['PNED']
df['Average'] = (df['RatioF']+df['RatioD'])/2
df

df.to_excel("ratio comparison 400Random.xlsx", index=False)
```

Colab paid products - Cancel contracts here