

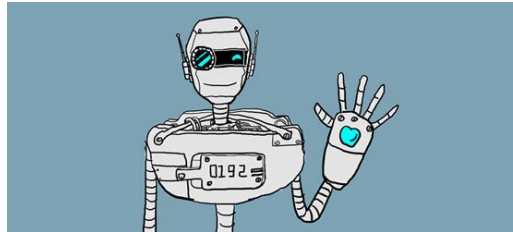


Raheel Shaikh

Follow

Oct 20, 2018 · 5 min read

- Gentle Start to Natural Language Processing using Python



## What is NLP ?

Natural language processing (NLP) is about developing applications and services that are able to understand human languages. Some Practical examples of NLP are speech recognition for eg: google voice search, understanding what the content is about or sentiment analysis etc.

## Benefits of NLP

As all of you know, there are millions of gigabytes every day are generated by blogs, social websites, and web pages.

There are many companies gathering all of these data for understanding users and their passions and give these reports to the companies to adjust their plans.

Suppose a person loves traveling and is regularly searching for a holiday destination, the searches made by the user is used to provide him with relative advertisements by online hotel and flight booking apps.

You know what, search engines are not the only implementation of natural language processing (NLP) and there are a lot of awesome implementations out there.

## NLP Implementations

These are some of the successful implementations of Natural Language Processing (NLP):

- **Search engines** like Google, Yahoo, etc. Google search engine understands that you are a tech guy so it shows you results related to you.
- **Social websites feed** like the Facebook news feed. The news feed algorithm understands your interests using natural language processing and shows you related Ads and posts more likely than other posts.
- **Speech engines** like Apple Siri.
- **Spam filters** like Google spam filters. It's not just about the usual **spam filtering**, now spam filters understand what's inside the email content and see if it's a spam or not.

How do I Start with NLP using Python?

Natural language toolkit (NLTK) is the most popular library for natural language processing (NLP) which was written in Python and has a big community behind it.

NLTK also is very easy to learn, actually, it's the easiest natural language processing (NLP) library that you'll use.

In this NLP Tutorial, we will use Python NLTK library.

Before I start installing NLTK, I assume that you know some [Python basics](#) to get started.

## Install nltk

If you are using Windows or Linux or Mac, you can install NLTK [using pip](#):

### \$ pip install nltk

You can use NLTK on Python 2.7, 3.4, and 3.5 at the time of writing this post.

Alternatively, you can [install it from source](#) from this [tar](#).

To check if NLTK has installed correctly, you can open python terminal and type the following:

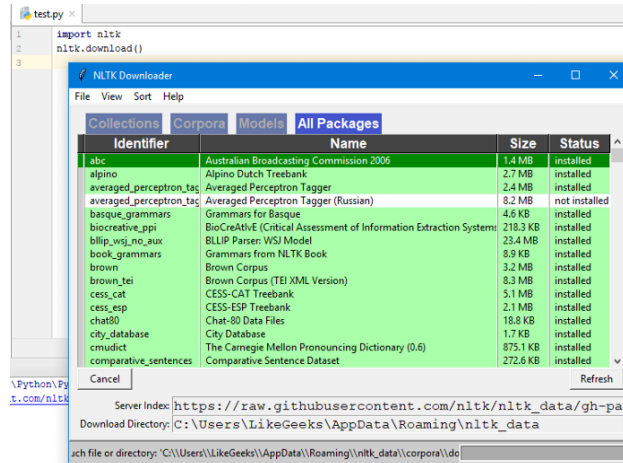
### Import nltk

If everything goes fine, that means you've successfully installed NLTK library.

Once you've installed NLTK, you should install the NLTK packages by running the following code:

```
import nltk
nltk.download()
```

This will show the NLTK downloader to choose what packages need to be installed.



You can install all packages since they have small sizes, so no problem. Now let's start the show.

## Here we will learn how to identify what the web page is about using NLTK in Python

First, we will grab a webpage and analyze the text to see what the page is about.

urllib module will help us to crawl the webpage

```
import urllib.request
response =
urllib.request.urlopen('https://en.wikipedia.org/wiki/SpaceX')
html = response.read()
print(html)
```

Top highlight

It's pretty clear from the link that page is about SpaceX now let us see whether our code is able to correctly identify the page's context.

We will use **Beautiful Soup** which is a Python library for pulling data out of HTML and XML files. We will use beautiful soup to clean our webpage text of HTML tags.

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, 'html5lib')
text = soup.get_text(strip = True)
print(text)
```

The screenshot shows a large block of text that has been cleaned of HTML tags. It starts with "SpaceX - Wikipedia" and continues with a detailed description of the company, including its founding, milestones, and current status. The text is wrapped in a monospaced font, typical of a terminal or code editor output.

You will get an output somewhat like this

Now we have clean text from the crawled web page, let's convert the text into tokens.

```
tokens = [t for t in text.split()]
print(tokens)
```

your output text is now converted into tokens

```
[ 'Spacek', '-', 'Wikipediadocument.documentElement.className', '-', 'document.documentElement.className.replace(/(\\|\\s|c
lient-nojs|\\u|\\$|),, "Bilient-jid2", ');(window.RLQ=window.RLQ||[]).push(function(){mw.config.set({"wgCanonicalNamespac
e":"","wgCanonicalSpecialPageName":false,"wgNamespacesNumber":0,"wgNamespaces":{"Spacek":"","wgArticle":"","wgRevisionId":863
93776,"wgRevisionId":86393776,"wgArticleId":832774,"wgIsArticle":true,"wgIsRedirect":false,"wgAction":"view","wgUserName":n
ull,"wgUserGroups":[""],"wgCategories":["CSI","main","BOT"],"original-url":{"status","unknown"},"CSI","errors","miss
ing","author","or","editor"},"CSI","main":{"url"},"Pages","containing","links","to","subscription-only","co
ntent"},"CSI","French-language","sources":{"fr"},"Articles","with","short","description","Use","American","English",
"from","November","2015"},"All","Wikipedia","articles","written","in","American","English","True","body","dates","fro
m","November","2015"},"Coordinates","not","on","Wikidata"},"Articles","containing","potentially","dated","statemen
t","from","April","2017"},"All","articles","containing","potentially","dated","statements"},"Articles","containing",
"potentially","dated","statements","from","March","2013"},"All","articles","with","unsourced","statements"},"Article
s","with","unsourced","statements","from","September","2018"},"Articles","containing","potentially","dated","stateme
nts","from","May","2012"},"Articles","containing","potentially","dated","statements","from","February","2018"},"Artic
les","containing","potentially","dated","statements","from","March","2017"},"Wikipedia","articles","in","need","o
f","updating","from","May","2017"},"All","Wikipedia","articles","in","need","of","updating"},"Articles","containin
g","potentially","dated","statements","from","October","2012"},"Wikipedia","articles","in","need","of","updating",
"from","January","2011"},"Official","websites","different","in","Wikidata","and","Wikipedia"},"bitter","username","is
me","as","Wikidata"},"bitter","username","different","from","Wikidata"},"Wikipedia","articles","with","R1050","iden
tifiers"},"Wikipedia","articles","with","GND"},"Identifiers"},"Wikipedia","articles","with","ISNI"},"Identifiers"},"Wikiped
```

## Count word Frequency

nlTK offers a function **FreqDist()** which will do the job for us. Also, we will remove stop words (a, at, the, for etc) from our web page as we don't need them to hamper our word frequency count. We will plot the graph for most frequently occurring words in the webpage in order to get the clear picture of the context of the web page

```
from nltk.corpus import stopwords
sr= stopwords.words('english')
clean_tokens = tokens[:]
for token in tokens:
    if token in stopwords.words('english'):
        clean_tokens.remove(token)

freq = nltk.FreqDist(clean_tokens)
for key,val in freq.items():
    print(str(key) + ':' + str(val))

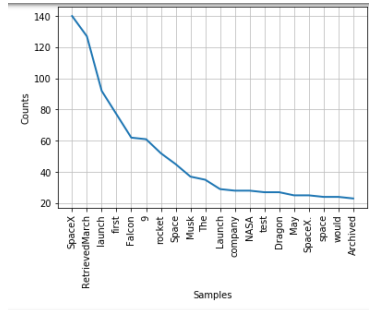
freq.plot(20, cumulative=False)
```

```

SpaceX:148
~:5
Wikipedia:document.documentElement.className:1
~:1
document.documentElement.className.replace(/(?!\\s)client-nojs(?:\\s|$)/,1
$!client-js&2":1
);(window.RLQ=window.RLQ||[]).push(function(){mw.config.set({"wgCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wg
NamespaceNumber":0,"wgPageName":"SpaceX","wgTitle":"SpaceX","wgCurRevisionId":863913776,"wgRevisionId":863913776,"wgArticleI
d":"1307794","wgArticle":"true","wgIsRedirect":false,"wgAction":"view","wgUserName":null,"wgUserGroups":[""],"wgCategories":["C
S:1
Main:1:15
BOT:1:12
original-url:1:12
status:1:2
unknown":,"CS:1:1
errors:2
missing:2
author:2

```

frequency word count output



graph of 20 most frequent words.

Great!!! the code has correctly identified that the web page speaks about **SpaceX**.

It was so simple and interesting right !!! you can similarly identify the news articles, blogs etc.

I have done my best to make the article simple and interesting for you, hope you found it useful and interesting too.

You have successfully taken your first step towards NLP, there is an ocean to explore for you...

If you liked this post give it a **Clap**, it inspires me to write and share more with you guys :)

Thank you...

### Sign up for The Daily Pick

By Towards Data Science

Hands-on real-world examples, research, tutorials, and cutting-edge techniques delivered Monday to Thursday. Make learning your daily ritual. [Take a look](#)



Get this newsletter

Create a free Medium account to get The Daily Pick in your inbox.

Machine Learning

Artificial Intelligence

NLP

Python

Data Science

#### Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. [Watch](#)

#### Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. [Explore](#)

#### Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. [Upgrade](#)

Medium

About

Help

Legal