



Get unlimited access

Open in app



Published in Towards Data Science

This is your **last** free member-only story this month. [Upgrade for unlimited access.](#)

Frank Andrade [Follow](#)Mar 13, 2021 · 7 min read ★ · [Listen](#)

Save



5 Simple Ways to Tokenize Text in Python

Tokenizing text, a large corpus and sentences of different language.



Photo by [Laurentiu lordache](#) on [Unsplash](#)

Tokenization is a common task a data scientist comes across when working with text





Get unlimited access

Open in app

it's the foundation for developing good models and helps better understand the text we have.

Although tokenization in Python could be as simple as writing `.split()`, that method might not be the most efficient in some projects. That's why, in this article, I'll show 5 ways that will help you tokenize small texts, a large corpus or even text written in a language other than English.

Table of Contents

1. [Simple tokenization with `.split`](#)
2. [Tokenization with NLTK](#)
3. [Convert a corpus to a vector of token counts with Count Vectorizer \(sklearn\)](#)
4. [Tokenize text in different languages with spaCy](#)
5. [Tokenization with Gensim](#)

Note: Tokenization is one of the many tasks a data scientist do when cleaning and preparing data. In the article below, I wrote a guide to help you with these tedious tasks. The code of both articles is available on my [Github](#).

A Straightforward Guide to Cleaning and Preparing Data in Python

How to Identify and deal with dirty data.

towardsdatascience.com

1. Simple tokenization with `.split`

As we mentioned before, this is the simplest method to perform tokenization in Python. If you type `.split()`, the text will be separated at each blank space.

For this and the following examples, we'll be using a text narrated by Steve Jobs in the "Think Different" Apple commercial.





Get unlimited access

Open in app

```
human race forward, and while some may see them as the crazy ones,  
we see genius, because the ones who are crazy enough to think  
that they can change the world, are the ones who do."""
```

```
text.split()
```

If we write the code above, we'll obtain the following output.

```
['Here's', 'to', 'the', 'crazy', 'ones,', 'the', 'misfits,', 'the',  
'rebels,', 'the', 'troublemakers,', 'the', 'round', 'pegs', 'in',  
'the', 'square', 'holes.', 'The', 'ones', 'who', 'see', 'things',  
'differently', '-', 'they're', 'not', 'fond', 'of', 'rules.', 'You',  
'can', 'quote', 'them,', 'disagree', 'with', 'them,', 'glorify',  
'or', 'vilify', 'them,', 'but', 'the', 'only', 'thing', 'you',  
'can't', 'do', 'is', 'ignore', 'them', 'because', 'they', 'change',  
'things.', 'They', 'push', 'the', 'human', 'race', 'forward,',  
'and', 'while', 'some', 'may', 'see', 'them', 'as', 'the', 'crazy',  
'ones,', 'we', 'see', 'genius,', 'because', 'the', 'ones', 'who',  
'are', 'crazy', 'enough', 'to', 'think', 'that', 'they', 'can',  
'change', 'the', 'world,', 'are', 'the', 'ones', 'who', 'do.']
```

As you can see above, the `split()` method doesn't consider punctuation symbols as a separate token. This might change your project results.

4 Free and Paid Web Scraping Courses Every Data Scientist Should Take

Acquire this must-have skill every data scientist should have.

medium.com

2. Tokenization with NLTK

NLTK stands for Natural Language Toolkit. This is a suite of libraries and programs for statistical natural language processing for English written in Python.

NLTK contains a module called `tokenize` with a `word_tokenize()` method that will help us split a text into tokens. Once you installed NLTK, write the following code to





Get unlimited access

Open in app

```
from nltk.tokenize import word_tokenize
word_tokenize(text)
```

In this case, the default output is slightly different from the `.split` method showed above.

```
['Here', "'", 's', 'to', 'the', 'crazy', 'ones', ',', 'the',
'misfits', ',', 'the', 'rebels', ',', 'the', 'troublemakers', ',',
...]
```

In this case, the apostrophe (') in “*here’s*” and the comma (,) in “*ones,*” were considered as tokens.

3. Convert a corpus to a vector of token counts with Count Vectorizer (sklearn)

The previous methods become less useful when dealing with a large corpus because you’ll need to represent the tokens differently. Count Vectorizer will help us convert a collection of text documents to a vector of token counts. In the end, we’ll get a vector representation of the text data.

For this example, I’ll add a quote from Bill Gates to the previous text to build a dataframe that will be an example of a corpus.

```
import pandas as pd
texts = [
    """Here's to the crazy ones, the misfits, the rebels, the
troublemakers, the round pegs in the square holes. The ones who see
things differently – they're not fond of rules. You can quote them,
disagree with them, glorify or vilify them, but the only thing you
can't do is ignore them because they change things. They push the
human race forward, and while some may see them as the crazy ones,
we see genius, because the ones who are crazy enough to think that
they can change the world, are the ones who do.""" ,
    'I choose a lazy person to do a hard job. Because a lazy person will
find an easy way to do it.'
```





Get unlimited access

Open in app

Now we'll use `Count Vectorizer` to transform these texts within the `df` dataframe in a vector of token counts.

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 # initialize
3 cv = CountVectorizer(stop_words='english')
4 cv_matrix = cv.fit_transform(df['text'])
5 # create document term matrix
6 df_dtm = pd.DataFrame(cv_matrix.toarray(), index=df['author'].values, columns=cv.get_feature_names())
```

tokenization.py hosted with ❤ by GitHub

view raw

If you run that code, you'll get a frame that counts the number of times a word was mention in both texts.

	change	choose	crazy	differently	disagree	easy	fond	forward	genius	glorify	...	round	rules	square	thing
jobs	2	0	3	1	1	0	1	1	1	1	...	1	1	1	1
gates	0	1	0	0	0	1	0	0	0	0	...	0	0	0	0

2 rows x 34 columns

Image by Author

This becomes extremely useful when the dataframe contains a large corpus because it provides a matrix with words encoded as integers values, which are used as inputs in machine learning algorithms.

Count Vectorizer can have different parameters like `stop_words` that we defined above. However, keep in mind that the default regexp used by `Count Vectorizer` selects **tokens of 2 or more alphanumeric characters** (punctuation is completely ignored and always treated as a token separator)

4. Tokenize text in different languages with spaCy

When you need to tokenize text written in a language other than English, you can use spaCy. This is a library for advanced natural language processing, written in Python and Cython, that supports tokenization for more than 65 languages.

Let's tokenize the same Steve Jobs text but now translated in Spanish





Get unlimited access

Open in app

```
4 text_spanish = """Por los locos. Los marginados. Los rebeldes. Los problematicos.
5 Los inadaptados. Los que ven las cosas de una manera distinta. A los que no les gustan
6 las reglas. Y a los que no respetan el "status quo". Puedes citarlos, discrepar de ellos,
7 ensalzarlos o vilipendiarlos. Pero lo que no puedes hacer es ignorarlos... Porque ellos
8 cambian las cosas, empujan hacia adelante la raza humana y, aunque algunos puedan
9 considerarlos locos, nosotros vemos en ellos a genios. Porque las personas que están
10 lo bastante locas como para creer que pueden cambiar el mundo, son las que lo logran."""
11
12 doc = nlp(text_spanish)
13
14 tokens = [token.text for token in doc]
15 print(tokens)
```

tokenization.py hosted with ❤ by GitHub

[view raw](#)

In this case, we imported `Spanish` from `spacy.lang.es` but if you're working with text in English, just import `English` from `spacy.lang.en` Check the list of languages available [here](#).

If you run this code, you'll get the following output.

```
['Por', 'los', 'locos', '.', 'Los', 'marginados', '.', 'Los',
'rebeldes', '.', 'Los', 'problematicos', '.', '\n', 'Los',
'inadaptados', '.', 'Los', 'que', 'ven', 'las', 'cosas', 'de',
'una', 'manera', 'distinta', '.', 'A', 'los', 'que', 'no', 'les',
'gustan', '\n', 'las', 'reglas', '.', 'Y', 'a', 'los', 'que', 'no',
'respetan', 'el', '"', 'status', 'quo', '"', '.', 'Puedes',
'citarlos', ',', 'discrepar', 'de', 'ellos', ',', '\n',
'ensalzarlos', 'o', 'vilipendiarlos', '.', 'Pero', 'lo', 'que',
'no', 'puedes', 'hacer', 'es', 'ignorarlos', '...', 'Porque', 'ellos',
'\n', 'cambian', 'las', 'cosas', ',', 'empujan', 'hacia',
'adelante', 'la', 'raza', 'humana', 'y', ',', 'aunque', 'algunos',
'puedan', '\n', 'considerarlos', 'locos', ',', 'nosotros', 'vemos',
'en', 'ellos', 'a', 'genios', '.', 'Porque', 'las', 'personas',
'que', 'están', '\n', 'lo', 'bastante', 'locas', 'como', 'para',
'creer', 'que', 'pueden', 'cambiar', 'el', 'mundo', ',', 'son',
'las', 'que', 'lo', 'logran', '.']
```

As you can see spaCy, considers punctuation symbols as a separate token (even the new lines `\n` were included).





Get unlimited access

Open in app

Although for languages like Spanish and English, tokenization will be as simple as separating by whitespace, for non-romance languages such as Chinese and Japanese, the orthography might have no spaces to delimit “words” or “tokens.” In such cases, a library like spaCy will come in handy. [Here](#) you check more about the importance of tokenization in different languages.

5. Tokenization with Gensim

Gensim is a library for unsupervised topic modeling and natural language processing and also contains a tokenizer. Once you install Gensim, tokenizing text will be as simple as writing the following code.

```
from gensim.utils import tokenize
list(tokenize(text))
```

The output to this code is this.

```
['Here', 's', 'to', 'the', 'crazy', 'ones', 'the', 'misfits', 'the',
'rebels', 'the', 'troublemakers', 'the', 'round', 'pegs', 'in',
'the', 'square', 'holes', 'The', 'ones', 'who', 'see', 'things',
'differently', 'they', 're', 'not', 'fond', 'of', 'rules', 'You',
'can', 'quote', 'them', 'disagree', 'with', 'them', 'glorify', 'or',
'vilify', 'them', 'but', 'the', 'only', 'thing', 'you', 'can', 't',
'do', 'is', 'ignore', 'them', 'because', 'they', 'change', 'things',
'They', 'push', 'the', 'human', 'race', 'forward', 'and', 'while',
'some', 'may', 'see', 'them', 'as', 'the', 'crazy', 'ones', 'we',
'see', 'genius', 'because', 'the', 'ones', 'who', 'are', 'crazy',
'enough', 'to', 'think', 'that', 'they', 'can', 'change', 'the',
'world', 'are', 'the', 'ones', 'who', 'do']
```

As you can see, Gensim splits every time it encounters a punctuation symbol e.g. `Here` ,
`s` , `can`, `t`

Summary

Tokenization presents different challenges, but now you know 5 different ways to deal with them. The `split` method is a simple tokenizer that separates text by white



[Get unlimited access](#)[Open in app](#)

[Join my email list with 3k+ people to get my Python for Data Science Cheat Sheet I use in all my tutorials \(Free PDF\).](#)

If you enjoy reading stories like these and want to support me as a writer, consider signing up to become a Medium member. It's \$5 a month, giving you unlimited access to stories on Medium. If you sign up using [my link](#), I'll earn a small commission with no extra cost to you.

Read every story from Frank Andrade (and thousands of other writers on Medium)

As a Medium member, a portion of your membership fee goes to writers you read, and you get full access to every story...

frank-andrade.medium.com

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

[Get this newsletter](#)

Emails will be sent to rootoor225@gmail.com.
[Not you?](#)

