

```
# Aufbau einer Sentimentanalyse
# Import der Bibliotheken
```

```
!pip install matplotlib nltk pandas textblob
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/p
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (3.7)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.3.
Requirement already satisfied: textblob in /usr/local/lib/python3.7/dist-packages (0.
Requirement already satisfied: cyclers>=0.10 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from n
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages
```

## ▼ Using Vader for sentiment analysis

NLTK nutzt zur Analyse das Tool VADER

```
import nltk
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
nltk.download('movie_reviews')
nltk.download('punkt')
```

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

Der Aufruf Polarity gibt uns eine Analyse des jeweiligen Textkorpus an.

- neg: negative sentiments im Textkorpus
- neu: neutrale sentiments im Textkorpus

✓ 0 s Abgeschlossen um 10:50



- compound: aggregiertes sentiment

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer as SIA

sia = SIA() # Sentiment Intensity Analyzer
sia.polarity_scores("Donald Trump's hatred of looking foolish and Democrats' conviction th

{'compound': -0.2263, 'neg': 0.223, 'neu': 0.604, 'pos': 0.173}

text = "Donald Trump's hatred of looking foolish and Democrats' conviction that they have a
sia.polarity_scores(text)

{'compound': -0.2263, 'neg': 0.223, 'neu': 0.604, 'pos': 0.173}
```

Zum Bearbeiten doppelklicken (oder Eingabe)

```
#from nltk.sentiment.vader import SentimentIntensityAnalyzer as SIA

#sia = SIA() # Sentiment Intensity Analyzer
#sia.polarity_scores("you have nice eyes.")

#text = "I just got a call from my boss - does he realise it's Saturday?"
#sia.polarity_scores(text)
```

Using textblob for sentiment analysis

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
from textblob import Blobber
from textblob.sentiments import NaiveBayesAnalyzer

blob = TextBlob("Donald Trump's hatred of looking foolish and Democrats' conviction that th
blob.sentiment

Sentiment(polarity=0.5, subjectivity=0.75)

blob = TextBlob("Donald Trump's hatred of looking foolish and Democrats' conviction that th
blob.sentiment

Sentiment(polarity=0.5, subjectivity=0.75)

blobber = Blobber(analyzer=NaiveBayesAnalyzer())
```

```
blob = blobber("Donald Trump's hatred of looking foolish and Democrats' conviction that the
blob.sentiment
```

```
Sentiment(classification='pos', p_pos=0.9834582376441825, p_neg=0.01654176235582116)
```

Nachfolgende Sätze stammen aus dem Datensatz:

- They talked about Romney's trouble connecting Perry's trouble speaking Paul's radicalism Michele Bachmann's inexperience and Rick Santorum's fervor.

Nachfolgende Sätze sind frei erfunden:

- Trump is better than Biden

## Comparison chart

```
import pandas as pd
pd.set_option("display.max_colwidth", 200)

df = pd.DataFrame({'content': [
    "Donald Trump's hatred of looking foolish and Democrats' conviction that they have a wi
    "Trump berated May for Britain not doing enough, in his assessment, to contain Iran. He
    "Prime Minister May has endured Trump's churlish temper before, but still her aides wer
    "Her appearance on the friendly show may be another sign that she intends on staying re
    "There are more than a hundred Republican-held congressional districts across the count
    "There are more than a hundred Republican-held congressional districts across the count
    "If seats that look like this one in Pennsylvania are toss-ups in November, it's going
]])
df
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

	content
0	Donald Trump's hatred of looking foolish and Democrats' conviction that they have a winning hand is leaving the President with no way out the stalemate over his border wall.
1	Trump berated May for Britain not doing enough, in his assessment, to contain Iran. He questioned her over Brexit and complained about the trade deals he sees as unfair with European countries.
2	Prime Minister May has endured Trump's churlish temper before, but still her aides were shaken by his especially foul mood, according to U.S. and European officials briefed on the conversation.

this especially four mood, according to U.S. and European officials briefed on the conversation.

3 After all, if America's household wealth were distributed evenly across the population, then every family of four would have a net worth of 1.2 million dollars.

4 Hillary Clinton was asked, if she'd make a prediction on the 2020 election and said she joked, saying she'd save the insight for her upcoming book. Her appearance on the friendly show may be anoth...

5 Hillary Clinton was asked, if she'd make a prediction on the 2020 election and said she joked, saying she'd save the insight for her upcoming book.

6 Her appearance on the friendly show may be another sign that she intends on staying relevant prior to 2020.

Sätze auf deutsch:

- [0] Donald Trumps Hass darauf, dumm dazustehen, und die Überzeugung der Demokraten, dass sie die Oberhand haben, lassen dem Präsidenten keinen Ausweg aus der Pattsituation um seine Grenzmauer. [<https://edition.cnn.com/2019/01/10/politics/donald-trump-shutdown/index.html>].

- [1] Trump beschimpfte May, weil Großbritannien seiner Meinung nach nicht genug tut, um den Iran einzudämmen. Er stellte sie wegen des Brexit in Frage und beschwerte sich über die seiner Meinung nach unfairen Handelsabkommen mit europäischen Ländern. [[https://www.washingtonpost.com/politics/five-days-of-fury-inside-trumps-paris-temper-election-woes-and-staff-upheaval/2018/11/13/e90b7cba-e69e-11e8-a939-9469f1166f9d\\_story.html](https://www.washingtonpost.com/politics/five-days-of-fury-inside-trumps-paris-temper-election-woes-and-staff-upheaval/2018/11/13/e90b7cba-e69e-11e8-a939-9469f1166f9d_story.html)].

- [2] Premierministerin May hat Trumps rüpelhaftes Temperament schon früher ertragen,

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

[[https://www.washingtonpost.com/politics/five-days-of-fury-inside-trumps-paris-temper-election-woes-and-staff-upheaval/2018/11/13/e90b7cba-e69e-11e8-a939-9469f1166f9d\\_story.html](https://www.washingtonpost.com/politics/five-days-of-fury-inside-trumps-paris-temper-election-woes-and-staff-upheaval/2018/11/13/e90b7cba-e69e-11e8-a939-9469f1166f9d_story.html)].

- [3] Wenn das Vermögen der amerikanischen Haushalte gleichmäßig auf die Bevölkerung verteilt wäre, hätte jede vierköpfige Familie ein Nettovermögen von 1,2 Millionen Dollar. [[https://www.washingtonpost.com/opinions/2019/04/10/if-youre-bothered-by-bernies-millionaire-status-vote-him/?noredirect=on&utm\\_term=.cf6afaf10188](https://www.washingtonpost.com/opinions/2019/04/10/if-youre-bothered-by-bernies-millionaire-status-vote-him/?noredirect=on&utm_term=.cf6afaf10188)].

- [4] Hillary Clinton wurde gefragt, ob sie eine Vorhersage für die Wahl 2020 machen würde, und sagte scherzhaft, dass sie sich die Erkenntnisse für ihr kommendes Buch aufheben würde. Ihr Auftritt in der "Friendly Show" könnte ein weiteres Zeichen dafür sein, dass sie auch vor 2020 noch relevant bleiben will. [<https://www.foxnews.com/entertainment/hillary-clinton-gets-asked-painfully-scripted-questions-on-colbert-report>].

- [5] und [6] ist gesplittet. [<https://www.foxnews.com/entertainment/hillary-clinton-gets-asked-painfully-scripted-questions-on-colbert-report>].
- [7] Es gibt landesweit mehr als hundert von den Republikanern gehaltene Kongressbezirke, in denen der Vorsprung geringer ist als in 17. Wenn Sitze, die so aussehen wie dieser in Pennsylvania, im November in die engere Wahl kommen, wird es ein Blutbad geben. [<https://www.bbc.com/news/world-us-canada-43390652>].
- [8] und [9] sind gesplittet.

Erklärungen zu den jeweiligen Bias in den Textkorpora:

- Spin [0;2]
- Unsubstantiated Claims [3]
- Opinion Statemets presented as facts [4;6]
- Sensationalism/Emotionalism [7;9]

## Spin:

Spin ist eine Form der Medienbeeinflussung, die eine vage, dramatische oder sensationelle Sprache bedeutet. Wenn Journalisten eine Geschichte "drehen", weichen sie von objektiven, messbaren Fakten ab. Spin ist eine Form der Medienbeeinflussung, die den Blick des Lesers trübt und ihn daran hindert, sich ein genaues Bild von den Ereignissen zu machen.

Beispiele für Spin-Wörter und -Sätze:

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

Serious

Refuse

Crucial

High-stakes

Tirade

Landmark

Latest in a string of...

Major

Turn up the heat

Critical

Decrying

Offend

Stern talks

Offensive

Facing calls to...

Meaningful

Even though

Monumental

Significant

Manchmal verwenden die Medien "spin words" und Phrasen, um schlechtes Verhalten zu unterstellen. Diese Worte werden oft verwendet, ohne harte Fakten, direkte Zitate oder bezugtes Verhalten zu liefern:

Finally

Surfaced

Acknowledged

Emerged

Refusing to say

Conceded

Dodged

Admission

Came to light

Admit to

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)  
anmutende Wörter als Ersatz für das Wort "said". Zum Beispiel:

Mocked

Raged

Bragged

Fumed

Lashed out

Incensed

Scoffed

Frustration

Erupted

Rant

Roared

boasted  
Gloated

---

Beispiele (Textauszug [0;2]) für die Voreingenommenheit der Medien:

Die Washington Post verwendet eine Vielzahl dramatischer, sensationslüsterner Worte, um die Geschichte so zu beschreiben, dass Trump emotional und verstört erscheint. Sie verweisen auch auf die "Eitelkeit" des Präsidenten, ohne dafür Beweise zu liefern.

## Unbewiesene Behauptungen

Journalisten stellen in ihrer Berichterstattung manchmal Behauptungen auf, ohne sie mit Beweisen zu untermauern.

Behauptungen, die den Anschein erwecken, Tatsachen zu sein, aber keine spezifischen Beweise enthalten, sind ein wichtiges Anzeichen für diese Art der Medienbeeinflussung. Dies wird oft als eine Art von Fake News bezeichnet.

Beispiel Textauszug Dieser Kolumnist der Washington Post stellt eine Behauptung über die Vermögensverteilung auf, ohne anzugeben, woher sie stammt. Wer diese Zahlen ermittelt hat und wie geht aus dem Text nicht hervor?

Meinungsäußerungen, die als Fakten dargestellt werden

Manchmal verwenden Journalisten subjektive Formulierungen oder Aussagen unter dem Vorwand, objektiv zu berichten. Selbst wenn ein Medienunternehmen einen Artikel als sachliche

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

Eine subjektive Aussage ist eine Aussage, die auf persönlichen Meinungen, Annahmen, Überzeugungen, Vorlieben oder Interpretationen beruht. Sie spiegeln wider, wie der Autor die Realität sieht, was er für die Wahrheit hält. Es handelt sich um eine Aussage, die durch die spezifische Perspektive oder Linse des Verfassers gefärbt ist und nicht anhand konkreter Fakten und Zahlen im Artikel überprüft werden kann.

Es gibt objektive Modifikatoren - "blue" "old" "single-handedly" "statistically" "domestic", deren Bedeutung überprüft werden kann. Auf der anderen Seite gibt es subjektive Modifikatoren - "suspicious," "dangerous," "extreme," "dismissively," "apparently", die eine Frage der Interpretation sind.

Die Interpretation kann dazu führen, dass ein und dieselben Ereignisse als zwei sehr unterschiedliche Vorfälle dargestellt werden. So kann beispielsweise ein politischer Protest, bei dem sich Menschen mitten auf die Straße setzen und den Verkehr blockieren, um auf ihr

Anliegen aufmerksam zu machen, als "peaceful" und "productive" beschrieben werden, während andere ihn als "aggressive" und "disruptive" bezeichnen.

Zu den Wörtern, die subjektive Aussagen signalisieren, gehören:

Good/Better/Best  
Is considered to be  
Seemingly  
Extreme  
May mean that  
Could  
Apparently

Bad/Worse/Worst  
It's likely that  
Dangerous  
Suggests  
Would seem  
Decrying  
Possibly

Quelle: [<http://www.butte.edu/departments/cas/tipsheets/thinking/claims.html>]

Eine objektive Aussage hingegen ist eine Beobachtung von beobachtbaren Fakten. Sie beruht nicht auf Emotionen oder persönlichen Meinungen, sondern auf empirischen Beweisen - also auf etwas, das quantifizierbar und messbar ist.

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

werden:

Taipei 101 ist das höchste Gebäude der Welt.

Fünf plus vier ist gleich zehn.

Es gibt neun Planeten in unserem Sonnensystem.

Die erste Aussage ist (zum Zeitpunkt der Erstellung dieses Artikels) wahr, die beiden anderen sind

Quelle: [<http://www.butte.edu/departments/cas/tipsheets/thinking/claims.html>].

Die redaktionellen Überprüfungen von AllSides haben ergeben, dass einige Medien die Grenze zwischen subjektiven und objektiven Aussagen verwischen, was bei den Lesern zu Verwirrung



führen kann, und zwar auf zwei Arten, die unter diese Art von Medienverzerrung fallen:

Sie nehmen subjektive Aussagen in ihren Text auf und weisen sie nicht einer Quelle zu. (siehe Untere Platzierung von Meinungsäußerungen oder redaktionellen Inhalten auf der Homepage neben harten Nach

Quelle: [<https://www.allsides.com/media-bias/how-to-spot-types-of-media-bias#MindReading>]

## Sensationslust/Emotionalität

Sensationslust ist eine Art von Medienverzerrung, bei der Informationen so präsentiert werden, dass sie einen Schock auslösen oder einen tiefen Eindruck hinterlassen. Oft wird den Lesern ein falscher Eindruck vermittelt, dass alle bisherigen Berichte zu dieser ultimativen Geschichte geführt haben.

Sensationslustige Sprache ist oft dramatisch, aber vage. Oft wird übertrieben - auf Kosten der Genauigkeit - oder die Realität verzerrt, um den Leser in die Irre zu führen oder eine starke Reaktion hervorzurufen.

In Anbetracht dieser Art von Medienvorurteilen erhöhen Reporter oft die Lesbarkeit ihrer Artikel durch die Verwendung von anschaulichen Verben. Es gibt jedoch viele Verben, die schwerwiegende Implikationen haben, die nicht objektiv bestätigt werden können: "blast" "slam" "bury" "abuse" "destroy" "worry.". Zu den von den Medien verwendeten Wörtern und Phrasen, die auf Sensationslust/Emotionalität hindeuten, gehören:

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

Rips

Chaotic

Lashed out

Onslaught

Scathing

Showdown

Explosive

Slams

Forcing

Warning

Embroidered in...

Torrent of tweets

Desperate

Die BBC bedient sich der Sensationslust in Form von Übertreibungen, da es bei der Wahl wohl kaum zu einem Blutvergießen im wörtlichen Sinne kommen wird.

Quelle: <http://www.butte.edu/departments/cas/tipsheets/thinking/claims.html>.

```
def get_scores(content):
    blob = TextBlob(content)
    nb_blob = blobber(content)
    sia_scores = sia.polarity_scores(content)

    return pd.Series({
        'content': content,
        'textblob': blob.sentiment.polarity,
        'textblob_bayes': nb_blob.sentiment.p_pos - nb_blob.sentiment.p_neg,
        'nltk': sia_scores['compound'],
    })

scores = df.content.apply(get_scores)
scores.style.background_gradient(cmap='RdYlGn', axis=None, low=0.4, high=0.4)
```

	content	textblob	textblob_bayes	nltk
0	Donald Trump's hatred of looking foolish and Democrats' conviction that they have a winning hand is leaving the President with no way out the stalemate over his border wall.	0.500000	0.966916	-0.226300
1	Trump berated May for Britain not doing enough, in his assessment, to contain Iran. He questioned her over Brexit and complained about the trade deals he sees	-0.200000	0.978257	-0.735100
2	temper before, but still her aides were shaken by his especially foul mood, according to U.S. an European officials briefed on the conversation.	0.000000	0.999513	-0.329100
3	After all, if America's household wealth were distributed evenly across the population, then every family of four would have a net worth of 1.2 million dollars.	0.150000	0.977590	0.624900
4	Hillary Clinton was asked, if she'd make a prediction on the 2020 election and said she joked, saying she'd save the insight for her upcoming book. Her appearance on the friendly show may be another sign	0.187500	-0.804485	0.827100

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

Erläuterung der chart:

- Hier wird der compound angezeigt. Die aggregierte Form.

Quellen:

- <https://www.allsides.com>
- <http://www.butte.edu/departments/cas/tipsheets/thinking/claims.html>

## Überlegungen und Todos:

- Bei der Sentiment-Analyse wird "rechnerisch" ermittelt, ob ein Text positiv, negativ oder neutral ist. Sie ist auch als Meinungsanalyse bekannt, bei der die Meinung oder Einstellung eines Sprechers abgeleitet wird. Warum Stimmungsanalyse?
  - politischen Bereich wird es verwendet, um die politischen Ansichten zu verfolgen, um Konsistenz und Inkonsistenz zwischen Aussagen und Handlungen auf Regierungsebene zu erkennen. Es kann auch zur Vorhersage von Wahlergebnissen verwendet werden!
- Öffentliche Handlungen: Die Sentiment-Analyse wird auch verwendet, um soziale Phänomene zu überwachen und zu analysieren, um potenziell gefährliche Situationen zu erkennen und die allgemeine Stimmung in der Blogosphäre zu bestimmen.

## Was ist VADER?

- VADER (Valence Aware Dictionary and sEntiment Reasoner) ist ein lexikon- und regelbasiertes Stimmungsanalyse-Tool, das speziell auf die in sozialen Medien geäußerten Stimmungen abgestimmt ist. VADER verwendet eine Kombination aus einem

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

Entsprechend ihrer semantischen Ausrichtung als positiv oder negativ gekennzeichnet sind. VADER gibt nicht nur Auskunft über den Positivitäts- und Negativitätsscore, sondern auch darüber, wie positiv oder negativ ein Sentiment ist.

## Textanalyse

Methodiken:

- Counting words, big amount of documents
- Word counts with TF\_IDF
- Multi-word phrases and n-grams
- Creating own sentiment analyzer
- Bert, Neuronal Net

- WordNEt
- Word2Vec

## Word counting

```
from collections import Counter
```

```
Counter([1, 6, 7, 2, 7, 3, 1, 1, 3, 6, 1, 1])
```

```
Counter({1: 5, 2: 1, 3: 2, 6: 2, 7: 2})
```

```
# List of words
```

```
Counter(['Biden', 'rich', 'rich', 'hello', 'hello', 'Biden'])
```

```
Counter({'Biden': 2, 'hello': 2, 'rich': 2})
```

```
import re
```

```
text = """I went renting a boat, there I wanted to make a poker tournament. I wanted to play  
so I didn't catch any money. I was told I should enjoy myself,  
but I lost my whole money."""
```

```
# Force to all be lowercase because POKER and poker and Poker are the same  
text = text.lower()
```

```
# Remove anything that isn't a word character or a space
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotезugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
print("Cleaned sentence is:", text)
```

```
words = text.split(" ")  
Counter(words)
```

```
Cleaned sentence is: i went renting a boat there i wanted to make a poker tournament  
Counter({'a': 3,  
        'any': 1,  
        'because': 1,  
        'boat': 1,  
        'but': 1,  
        'catch': 1,  
        'didn't': 1,  
        'dont': 1,  
        'enjoy': 1,  
        'have': 1,  
        'holdem': 1,
```

```
'i': 9,  
'lost': 1,  
'lot': 1,  
'make': 1,  
'money': 2,  
'my': 1,  
'myself': 1,  
'of': 1,  
'often': 1,  
'play': 1,  
'poker': 3,  
'renting': 1,  
'seen': 1,  
'should': 1,  
'so': 1,  
'texas': 1,  
'that': 1,  
'there': 1,  
'to': 2,  
'told': 1,  
'tournament': 1,  
'wanted': 2,  
'was': 1,  
'went': 1,  
'whole': 1})
```

```
# Counting most common words  
Counter(words).most_common(5)
```

```
[('i', 9), ('a', 3), ('poker', 3), ('wanted', 2), ('to', 2)]
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
import requests
```

```
response = requests.get('https://www.gutenberg.org/cache/epub/43743/pg43743.txt.utf8')  
text = response.text
```

```
print(text[3100:4100])
```

ticular attention to their education. Though a man of the world, he was at the utmost pains in selecting those of distinguished worth, to whom only he committed the care of his children. Lord Finlay had promising parts; but force of mind makes a man capable of great vices or great virtues, but determines him to neither.

Education, discipline, and accidents of life, constitute him either a profound philosopher, or a great knave. The probity and disinterestedness of Mr Burt's principles recommended him to Lord

```
-----
Munster, for a tutor to his son.--He had been brought up to the
ministry, with an inclination to it, and entered into it with a fervent
desire of being as useful as he could. His education being all his
fortune, he subscribed, and took every step the church required, before
he was sufficiently acquainted with the doctrines subscribed to;--their
foundation in scripture, and the controversies which he afterwards found
had been raised, and carried on about them in the
```

```
# Counting words in textcorpora
```

```
text = text.lower()
```

```
text = re.sub("[^\w ]", "", text)
```

```
words = text.split(" ")
```

```
Counter(words).most_common(25)
```

```
[(',', 4768),
 ('the', 3080),
 ('of', 2368),
 ('to', 1867),
 ('and', 1399),
 ('a', 1105),
 ('in', 1044),
 ('that', 677),
 ('i', 667),
 ('her', 658),
 ('his', 580),
 ('is', 538),
 ('he', 504),
 ('with', 500),
 ('was', 481),
 ('it', 460),
 ('my', 458),
 ('she', 440),
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
('be', 399),
 ('but', 397),
 ('not', 388),
 ('which', 361),
 ('had', 352)]
```

```
# Extracting words with Regex
```

```
# Catch every word after 'she'
```

```
she_words = re.findall(r"\b[Sc]he (\w+)", text)
```

```
she_words[:25]
```

```
['died',
 'was',
 'was',
 'expresses',
 'saw',
```

```
'my',  
'stopping',  
'should',  
'therefore',  
'resided',  
'accompanied',  
'had',  
'went',  
'walkedforwards',  
'saw',  
'found',  
'seen',  
'wasgoing',  
'could',  
'again',  
'might',  
'told',  
'would',  
'had',  
'was']
```

```
# Extracting words with regex
```

```
# Catch every word after 'he'
```

```
he_words = re.findall(r"\b[Hh]e (\w+)", text)
```

```
he_words[:25]
```

```
['retired',  
'should',  
'was',  
'continually',  
'possessed',  
'was',
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
'subscribed',  
'afterwards',  
'dependedfor',  
'hadespoused',  
'preach',  
'disapproves',  
'preach',  
'must',  
'should',  
'then',  
'_is',  
'must',  
'thinks',  
'proved',  
'possessed',  
'had']
```

```
# Most common words after 'she'
Counter(she_words).most_common(25)
```

```
[('had', 63),
 ('was', 51),
 ('has', 15),
 ('could', 13),
 ('would', 11),
 ('is', 7),
 ('should', 5),
 ('might', 5),
 ('thought', 5),
 ('gave', 5),
 ('did', 5),
 ('told', 4),
 ('entertained', 4),
 ('the', 4),
 ('took', 4),
 ('only', 4),
 ('can', 4),
 ('saw', 3),
 ('found', 3),
 ('answered', 3),
 ('must', 3),
 ('you', 3),
 ('i', 3),
 ('will', 3),
 ('said', 3)]
```

```
# Most common words after 'he'
Counter(he_words).most_common(25)
```

```
[('had', 53),
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
('could', 14),
 ('has', 10),
 ('should', 9),
 ('must', 9),
 ('said', 9),
 ('will', 7),
 ('made', 6),
 ('then', 5),
 ('at', 5),
 ('saw', 5),
 ('told', 5),
 ('thought', 4),
 ('i', 4),
 ('never', 4),
 ('not', 4),
 ('who', 4),
 ('resided', 3),
 ('became', 3),
 ... ..]
```



```
( 'did', 3),
( 'found', 3),
( 'can', 3)]
```

```
# Comparing the words, which are top listed
```

```
import pandas as pd
```

```
df = pd.DataFrame({
    'he': Counter(he_words),
    'she': Counter(she_words)
}).fillna(0)
```

```
df['total'] = df.he + df.she
df['pct_he'] = df.he / df.total * 100
df.head()
```

```
# now we get a comparison chart
```

	he	she	total	pct_he
<b>retired</b>	1.0	1.0	2.0	50.000000
<b>should</b>	9.0	5.0	14.0	64.285714
<b>was</b>	45.0	51.0	96.0	46.875000
<b>continually</b>	1.0	0.0	1.0	100.000000
<b>possessed</b>	2.0	2.0	4.0	50.000000



Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
# Import libraries
```

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import re
from nltk.stem.porter import PorterStemmer
```

```
pd.options.display.max_columns = 30
%matplotlib inline
```

```
texts = [
    "Penny bought bright blue fishes.",
    "Penny bought bright blue and orange fish.",
    "The cat ate a fish at the store."]
```

```

    ..... ,
    "Penny went to the store. Penny ate a bug. Penny saw a fish.",
    "It meowed once at the bug, it is still meowing at the bug and the fish",
    "The cat is at the fish store. The cat is orange. The cat is meowing at the fish.",
    "Penny is a fish"
]

```

```
"Penny bought bright blue fishes".split()
```

```
['Penny', 'bought', 'bright', 'blue', 'fishes']
```

```

from sklearn.feature_extraction.text import CountVectorizer
count_vectorizer = CountVectorizer()

```

```

# .fit_transfer TOKENIZES and COUNTS
X = count_vectorizer.fit_transform(texts)

```

```
X.toarray()
```

```

array([[0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
        0],
       [1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
        0],
       [0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0,
        0],
       [0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 1, 0, 1, 1, 1,
        1],
       [1, 2, 0, 0, 0, 0, 2, 0, 1, 0, 1, 2, 1, 1, 1, 0, 0, 0, 1, 0, 3, 0,
        0],
       [0, 2, 0, 0, 0, 0, 0, 3, 2, 0, 3, 0, 0, 1, 0, 1, 0, 0, 0, 1, 5, 0,
        0],
       [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
        0],

```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
pd.DataFrame(X.toarray())
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
2	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0
3	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	3	1	0	1	1	1	1
4	1	2	0	0	0	0	2	0	1	0	1	2	1	1	1	0	0	0	1	0	3	0	0
5	0	2	0	0	0	0	0	3	2	0	3	0	0	1	0	1	0	0	0	1	5	0	0
6	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0

```
# Penny is a fish
# A fish is Penny
count_vectorizer.get_feature_names()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
  warnings.warn(msg, category=FutureWarning)
```

```
['and',
 'at',
 'ate',
 'blue',
 'bought',
 'bright',
 'bug',
 'cat',
 'fish',
 'fishes',
 'is',
 'it',
 'meowed',
 'meowing',
 'once',
 'orange',
 'penny',
 'saw',
 'still',
 'store',
 'the',
 'to',
 'went']
```

```
pd.DataFrame(X.toarray(), columns=count_vectorizer.get_feature_names())
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

0	0	0	0	1	1	1	0	0	0	1	0	0	0	0
1	1	0	0	1	1	1	0	0	1	0	0	0	0	0
2	0	1	1	0	0	0	0	1	1	0	0	0	0	0
3	0	0	1	0	0	0	1	0	1	0	0	0	0	0
4	1	2	0	0	0	0	2	0	1	0	1	2	1	1
5	0	2	0	0	0	0	0	3	2	0	3	0	0	1
6	0	0	0	0	0	0	0	0	1	0	1	0	0	0



```
# We'll make a new vectorizer
count_vectorizer = CountVectorizer(stop_words='english')
```

```
#count_vectorizer = CountVectorizer(stop_words=['the', 'and'])
# .fit_transfer TOKENIZES and COUNTS
X = count_vectorizer.fit_transform(texts)
print(count_vectorizer.get_feature_names())

['ate', 'blue', 'bought', 'bright', 'bug', 'cat', 'fish', 'fishes', 'meowed', 'meowin
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
warnings.warn(msg, category=FutureWarning)
```

```
# https://tartarus.org/martin/PorterStemmer/index-old.html
```

```
from nltk.stem.porter import PorterStemmer
porter_stemmer = PorterStemmer()
```

```
print(porter_stemmer.stem('fishes'))
print(porter_stemmer.stem('meowed'))
print(porter_stemmer.stem('oranges'))
print(porter_stemmer.stem('meowing'))
print(porter_stemmer.stem('orange'))
print(porter_stemmer.stem('go'))
print(porter_stemmer.stem('went'))
```

```
fish
meow
orang
meow
orang
go
went
```

```
porter_stemmer = PorterStemmer()
```

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

```
words = re.sub(r"[^A-Za-z0-9\-\_]", "", str_input).lower().split()
words = [porter_stemmer.stem(word) for word in words]
return words
```

```
count_vectorizer = CountVectorizer(stop_words='english', tokenizer=stemming_tokenizer)
X = count_vectorizer.fit_transform(texts)
print(count_vectorizer.get_feature_names())
```

```
['ate', 'blue', 'bought', 'bright', 'bug', 'cat', 'fish', 'meow', 'onc', 'orang', 'pe
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWa
% sorted(inconsistent)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
warnings.warn(msg, category=FutureWarning)
```

```
pd.DataFrame(X.toarray(), columns=count_vectorizer.get_feature_names())
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
warnings.warn(msg, category=FutureWarning)
```

	ate	blue	bought	bright	bug	cat	fish	meow	onc	orang	penni	saw	store	went
0	0	1	1	1	0	0	1	0	0	0	1	0	0	(
1	0	1	1	1	0	0	1	0	0	1	1	0	0	(
2	1	0	0	0	0	1	1	0	0	0	0	0	1	(
3	1	0	0	0	1	0	1	0	0	0	3	1	1	.
4	0	0	0	0	2	0	1	2	1	0	0	0	0	(
5	0	0	0	0	0	3	2	1	0	1	0	0	1	(
6	0	0	0	0	0	0	1	0	0	0	1	0	0	(



```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tfidf_vectorizer = TfidfVectorizer(stop_words='english', tokenizer=stemming_tokenizer, use_
X = tfidf_vectorizer.fit_transform(texts)
pd.DataFrame(X.toarray(), columns=tfidf_vectorizer.get_feature_names())
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWa
% sorted(inconsistent)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
warnings.warn(msg, category=FutureWarning)
```

	ate	blue	bought	bright	bug	cat	fish	meow	onc
0	0.000000	0.200000	0.200000	0.200000	0.000000	0.000	0.200000	0.000000	0.000000

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)

2	0.250000	0.000000	0.000000	0.000000	0.000000	0.250	0.250000	0.000000	0.000000
3	0.111111	0.000000	0.000000	0.000000	0.111111	0.000	0.111111	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.333333	0.000	0.166667	0.333333	0.166667
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.375	0.250000	0.125000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000	0.500000	0.000000	0.000000



Jetzt haben sich unsere Zahlen ein wenig verschoben. Es handelt sich nicht mehr nur um eine Zählung, sondern um den Prozentsatz der Wörter.

Wert = (Anzahl der Vorkommen des Wortes im Satz) / (Anzahl der Wörter im Satz)

Nachdem wir die Stoppwörter entfernt haben, ist der Begriff Fisch 50 % der Wörter in Penny ist

ein Fisch im Vergleich zu 37,5 % in Es miaute einmal den Fisch an, es miaut immer noch den Fisch an. Es miaute den Käfer und den Fisch an.

Hinweis: Wir haben den Prozentsatz der Wörter ermittelt, indem wir norm="l1" angegeben haben - sta

Jetzt erhalten wir bei der Suche relevantere Ergebnisse, da berücksichtigt wird, ob die Hälfte unserer Wörter Fische sind oder ob 1 % von Millionen von Wörtern Fische sind. Aber wir sind noch nicht fertig!

Quelle: <https://investigate.ai>

## Inverse matrix

```
tfidf_vectorizer = TfidfVectorizer(stop_words='english', tokenizer=stemming_tokenizer, use_
X = tfidf_vectorizer.fit_transform(texts)
df = pd.DataFrame(X.toarray(), columns=tfidf_vectorizer.get_feature_names())
df
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWa
% sorted(inconsistent)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning
warnings.warn(msg, category=FutureWarning)
```

	ate	blue	bought	bright	bug	cat	fish	meow	onc
0	0.000000	0.200000	0.200000	0.200000	0.000000	0.000	0.200000	0.000000	0.000000
1	0.000000	0.166667	0.166667	0.166667	0.000000	0.000	0.166667	0.000000	0.000000
3	0.111111	0.000000	0.000000	0.000000	0.111111	0.000	0.111111	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.333333	0.000	0.166667	0.333333	0.166667
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.375	0.250000	0.125000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000	0.500000	0.000000	0.000000

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)



```
# Searching the combine of "fish" and "cat"
```

```
# Just add the columns together
```

```
pd.DataFrame([df['fish'], df['cat'], df['fish'] + df['cat']], index=["fish", "cat", "fish -
```

fish cat fish + cat



	<b>Fish</b>	<b>Cat</b>	<b>Fish + Cat</b>
<b>0</b>	0.200000	0.000	0.200000
<b>1</b>	0.166667	0.000	0.166667
<b>2</b>	0.250000	0.250	0.500000
<b>3</b>	0.111111	0.000	0.111111
<b>4</b>	0.166667	0.000	0.166667
<b>5</b>	0.250000	0.375	0.625000
<b>6</b>	0.500000	0.000	0.500000

' "Penny bought bright blue fishes.", "Penny bought bright blue and orange fish.", "The cat ate a fish at the store.", "Penny went to the store. Penny ate a bug. Penny saw a fish.", "It meowed once at the bug, it is still meowing at the bug and the fish", "The cat is at the fish store. The cat is orange. The cat is meowing at the fish.", "Penny is a fish" '

Fehler beim automatischen Speichern Diese Datei wurde im Remotezugriff oder in einem anderen Tab aktualisiert. [Unterschied anzeigen](#)