

```
In [1]: import os  
import nltk  
#nltk.download()
```

```
In [2]: AI = '''Artificial intelligence (AI) is the intelligence of machines or software, as c
```

```
In [3]: from nltk.tokenize import word_tokenize
```

```
In [4]: AI_tokens = word_tokenize(AI)  
AI_tokens
```

```
Out[4]: ['Artificial',
'intelligence',
'(',
'AI',
')',
'is',
'the',
'intelligence',
'of',
'machines',
'on',
'software',
',',
'as',
'opposed',
'to',
'the',
'intelligence',
'of',
'humans',
'on',
'animals',
'.',
'It',
'is',
'also',
'the',
'field',
'of',
'study',
'in',
'computer',
'science',
'that',
'develops',
'and',
'studies',
'intelligent',
'machines',
'.',
'..',
'',
'AI',
'''',
'',
'may',
'also',
'refer',
'to',
'the',
'machines',
'themselvess',
'.']
```

```
In [5]: len(AI_tokens)
```

```
Out[5]: 51
```

```
In [6]: AI
```

```
Out[6]: 'Artificial intelligence (AI) is the intelligence of machines or software, as opposed to the intelligence of humans or animals. It is also the field of study in computer science that develops and studies intelligent machines. "AI" may also refer to the machines themselves.'
```

```
In [7]: from nltk.tokenize import sent_tokenize #GIVES SENTENCES
```

```
In [8]: AI_sent = sent_tokenize(AI)  
AI_sent
```

```
Out[8]: ['Artificial intelligence (AI) is the intelligence of machines or software, as opposed to the intelligence of humans or animals.',  
 'It is also the field of study in computer science that develops and studies intelligent machines.',  
 '"AI" may also refer to the machines themselves.']}
```

```
In [9]: #from nltk.tokenize import punct_tokenize
```

```
In [ ]:
```

```
In [ ]:
```

```
In [10]: AI = '''Artificial intelligence (AI) is the intelligence of machines or software, as opposed to the intelligence of humans or animals. It is also the field of study in computer science that develops and studies intelligent machines. "AI" may also refer to the machines themselves. '''
```

```
In [11]: from nltk.tokenize import blankline_tokenize #GIVES YOU HOW MANY PARAGRAPHS  
AI_blank = blankline_tokenize(AI)  
AI_blank
```

```
Out[11]: ['Artificial intelligence (AI) is the intelligence of machines or software, as opposed to the intelligence of humans or animals. It is also the field of study in computer science that develops and studies intelligent machines. "AI" may also refer to the machines themselves. ']
```

```
In [12]: len(AI_blank)
```

```
Out[12]: 1
```

```
In [13]: from nltk.tokenize import WhitespaceTokenizer  
WhitespaceTokenizer().tokenize(AI)
```

```
Out[13]: ['Artificial',
'intelligence',
'(AI)',
'is',
'the',
'intelligence',
'of',
'machines',
'on',
'software,' ,
'as',
'opposed',
'to',
'the',
'intelligence',
'of',
'humans',
'on',
'animals.',
'It',
'is',
'also',
'the',
'field',
'of',
'study',
'in',
'computer',
'science',
'that',
'develops',
'and',
'studies',
'intelligent',
'machines.',
'"AI"',
'may',
'also',
'refer',
'to',
'the',
'machines',
'themselves.]
```

```
In [14]: AI_tokens
```

```
Out[14]: ['Artificial',
'intelligence',
'(',
'AI',
')',
'is',
'the',
'intelligence',
'of',
'machines',
'on',
'software',
',',
'as',
'opposed',
'to',
'the',
'intelligence',
'of',
'humans',
'on',
'animals',
'.',
'It',
'is',
'also',
'the',
'field',
'of',
'study',
'in',
'computer',
'science',
'that',
'develops',
'and',
'studies',
'intelligent',
'machines',
'.',
'..',
'',
'AI',
'''',
'may',
'also',
'refer',
'to',
'the',
'machines',
'themselvess',
'.']
```

```
In [15]: #WE WILL USE UNI GRAM, BI GRAM, N GRAM USING NLTK
from nltk.util import bigrams, trigrams, ngrams
```

```
In [16]: string = "the best and most beautiful thing in the word cannot be seen or even touched"
quotes_tokens = word_tokenize(string)
quotes_tokens
```

```
Out[16]: ['the',
  'best',
  'and',
  'most',
  'beautiful',
  'thing',
  'in',
  'the',
  'word',
  'can',
  'not',
  'be',
  'seen',
  'on',
  'even',
  'touched',
  ',',
  'they',
  'must',
  'be',
  'felt',
  'with',
  'heart',
  '..']
```

```
In [17]: string
```

```
Out[17]: 'the best and most beautiful thing in the word cannot be seen or even touched, they must be felt with heart.'
```

```
In [18]: quotes_tokens
```

```
Out[18]: ['the',
  'best',
  'and',
  'most',
  'beautiful',
  'thing',
  'in',
  'the',
  'word',
  'can',
  'not',
  'be',
  'seen',
  'on',
  'even',
  'touched',
  ',',
  'they',
  'must',
  'be',
  'felt',
  'with',
  'heart',
  '..']
```

```
In [19]: len(quotes_tokens)
```

```
Out[19]: 24
```

```
In [20]: quotes_bigrams = list(nltk.bigrams(quotes_tokens))
quotes_bigrams
```

```
Out[20]: [('the', 'best'),
('best', 'and'),
('and', 'most'),
('most', 'beautiful'),
('beautiful', 'thing'),
('thing', 'in'),
('in', 'the'),
('the', 'word'),
('word', 'can'),
('can', 'not'),
('not', 'be'),
('be', 'seen'),
('seen', 'or'),
('or', 'even'),
('even', 'touched'),
('touched', ','),
(',', 'they'),
('they', 'must'),
('must', 'be'),
('be', 'felt'),
('felt', 'with'),
('with', 'heart'),
('heart', '.')]
```

```
In [21]: quotes_tokens
```

```
Out[21]: ['the',
'best',
'and',
'most',
'beautiful',
'thing',
'in',
'the',
'word',
'can',
'not',
'be',
'seen',
'on',
'even',
'touched',
',',
'they',
'must',
'be',
'felt',
'with',
'heart',
'.']
```

```
In [22]: quotes_trigrams = list(nltk.trigrams(quotes_tokens))
quotes_trigrams
```

```
Out[22]: [('the', 'best', 'and'),
('best', 'and', 'most'),
('and', 'most', 'beautiful'),
('most', 'beautiful', 'thing'),
('beautiful', 'thing', 'in'),
('thing', 'in', 'the'),
('in', 'the', 'word'),
('the', 'word', 'can'),
('word', 'can', 'not'),
('can', 'not', 'be'),
('not', 'be', 'seen'),
('be', 'seen', 'or'),
('seen', 'or', 'even'),
('or', 'even', 'touched'),
('even', 'touched', ','),
('touched', ',', 'they'),
(',', 'they', 'must'),
('they', 'must', 'be'),
('must', 'be', 'felt'),
('be', 'felt', 'with'),
('felt', 'with', 'heart'),
('with', 'heart', '.')]
```

```
In [23]: quotes_ngrams = list(nltk.ngrams(quotes_tokens))
quotes_ngrams
```

#TypeError: ngrams() missing 1 required positional argument: 'n'

TypeError

Traceback (most recent call last)

Cell In[23], line 1

```
----> 1 quotes_ngrams = list(nltk.ngrams(quotes_tokens))
      2 quotes_ngrams
```

TypeError: ngrams() missing 1 required positional argument: 'n'

```
In [24]: quotes_ngrams = list(nltk.ngrams(quotes_tokens,6))
quotes_ngrams
```

```
Out[24]: [('the', 'best', 'and', 'most', 'beautiful', 'thing'),
('best', 'and', 'most', 'beautiful', 'thing', 'in'),
('and', 'most', 'beautiful', 'thing', 'in', 'the'),
('most', 'beautiful', 'thing', 'in', 'the', 'word'),
('beautiful', 'thing', 'in', 'the', 'word', 'can'),
('thing', 'in', 'the', 'word', 'can', 'not'),
('in', 'the', 'word', 'can', 'not', 'be'),
('the', 'word', 'can', 'not', 'be', 'seen'),
('word', 'can', 'not', 'be', 'seen', 'or'),
('can', 'not', 'be', 'seen', 'or', 'even'),
('not', 'be', 'seen', 'or', 'even', 'touched'),
('be', 'seen', 'or', 'even', 'touched', ','),
('seen', 'or', 'even', 'touched', ',', 'they'),
('or', 'even', 'touched', ',', 'they', 'must'),
('even', 'touched', ',', 'they', 'must', 'be'),
('touched', ',', 'they', 'must', 'be', 'felt'),
(',', 'they', 'must', 'be', 'felt', 'with'),
('they', 'must', 'be', 'felt', 'with', 'heart'),
('must', 'be', 'felt', 'with', 'heart', '.')]
```

```
In [25]: quotes_ngrams = list(nltk.ngrams(quotes_tokens,8))
quotes_ngrams
```

```
Out[25]: [('the', 'best', 'and', 'most', 'beautiful', 'thing', 'in', 'the'),
('best', 'and', 'most', 'beautiful', 'thing', 'in', 'the', 'word'),
('and', 'most', 'beautiful', 'thing', 'in', 'the', 'word', 'can'),
('most', 'beautiful', 'thing', 'in', 'the', 'word', 'can', 'not'),
('beautiful', 'thing', 'in', 'the', 'word', 'can', 'not', 'be'),
('thing', 'in', 'the', 'word', 'can', 'not', 'be', 'seen'),
('in', 'the', 'word', 'can', 'not', 'be', 'seen', 'or'),
('the', 'word', 'can', 'not', 'be', 'seen', 'or', 'even'),
('word', 'can', 'not', 'be', 'seen', 'or', 'even', 'touched'),
('can', 'not', 'be', 'seen', 'or', 'even', 'touched', ','),
('not', 'be', 'seen', 'or', 'even', 'touched', ',', 'they'),
('be', 'seen', 'or', 'even', 'touched', ',', 'they', 'must'),
('seen', 'or', 'even', 'touched', ',', 'they', 'must', 'be'),
('or', 'even', 'touched', ',', 'they', 'must', 'be', 'felt'),
('even', 'touched', ',', 'they', 'must', 'be', 'felt', 'with'),
('touched', ',', 'they', 'must', 'be', 'felt', 'with', 'heart'),
(',', 'they', 'must', 'be', 'felt', 'with', 'heart', '.')]
```

```
In [26]: quotes_ngrams = list(nltk.ngrams(quotes_tokens,24))
quotes_ngrams
```

```
Out[26]: [('the',
'best',
'and',
'most',
'beautiful',
'thing',
'in',
'the',
'word',
'can',
'not',
'be',
'seen',
'or',
'even',
'touched',
',',
',',
'they',
'must',
'be',
'felt',
'with',
'heart',
'.')]
```

```
In [27]: quotes_ngrams = list(nltk.ngrams(quotes_tokens,25))
quotes_ngrams
```

```
Out[27]: []
```

```
In [28]: from nltk.stem import PorterStemmer #give root word
pst = PorterStemmer()
```

```
In [29]: pst.stem('affection')
```

```
Out[29]: 'affect'
```

```
In [30]: pst.stem('playing')
```

```
Out[30]: 'play'
```

```
In [31]: words_to_stem = ['give', 'giving', 'given', 'gave']
for words in words_to_stem:
    print(words+ ':' + pst.stem(words))
```

```
give:give
giving:give
given:given
gave:gave
```

```
In [32]: words_to_stem = ['give', 'giving', 'given', 'gaven', 'thinking', 'loving', 'maximum']
for words in words_to_stem:
    print(words+ ':' + pst.stem(words))
```

```
give:give
giving:give
given:given
gaven:gaven
thinking:think
loving:love
maximum:maximum
```

```
In [33]: from nltk.stem import LancasterStemmer #GIVES CORE ROOT WORD
lst = LancasterStemmer()
for words in words_to_stem:
    print(words+ ':' + lst.stem(words))
```

```
give:giv
giving:giv
given:giv
gaven:gav
thinking:think
loving:lov
maximum:maxim
```

```
In [34]: from nltk.stem import SnowballStemmer
sbst = SnowballStemmer('english')
for words in words_to_stem:
    print(words+ ':' + sbst.stem(words))
```

```
give:give
giving:give
given:given
gaven:gaven
thinking:think
loving:love
maximum:maximum
```

```
In [35]: from nltk.stem import wordnet
from nltk.stem import WordNetLemmatizer #Proper word
word_lem = WordNetLemmatizer()
```

```
In [36]: words_to_stem
```

```
Out[36]: ['give', 'giving', 'given', 'gaven', 'thinking', 'loving', 'maximum']
```

```
In [37]: for words in words_to_stem:  
    print(words+ ':' +word_lemmatize(words))
```

```
give:give  
giving:giving  
given:given  
gaven:gaven  
thinking:thinking  
loving:loving  
maximum:maximum
```

```
In [38]: from nltk.corpus import stopwords #commonly used words
```

```
In [39]: stopwords.words('english')
```

```
out[39]: ['i',
'me',
'my',
'myself',
'we',
'our',
'ours',
'ourselves',
'you',
"you're",
"you've",
"you'll",
"you'd",
'your',
'yours',
'yourself',
'yourselves',
/he',
'him',
'his',
'himself',
'she',
"she's",
'her',
'hers',
'herself',
'it',
"it's",
'its',
'itself',
'they',
'them',
'their',
'theirs',
'themselves',
'what',
'which',
'who',
'whom',
'this',
'that',
"that'll",
'these',
'those',
'am',
'is',
'are',
'was',
'were',
'be',
'been',
'being',
'have',
'has',
'had',
'having',
'do',
'does',
'did',
'doing',
```

'a',
'an',
'the',
'and',
'but',
'if',
'or',
'because',
'as',
'until',
'while',
'of',
'at',
'by',
'for',
'with',
'about',
'against',
'between',
'into',
'through',
'during',
'before',
'after',
'above',
'below',
'to',
'from',
'up',
'down',
'in',
'out',
'on',
'off',
'over',
'under',
'again',
'further',
'then',
'once',
'here',
'there',
'when',
'where',
'why',
'how',
'all',
'any',
'both',
'each',
'few',
'more',
'most',
'other',
'some',
'such',
'no',
'nor',
'not',
'only',

```
'own',
'same',
'so',
'than',
'too',
'very',
's',
't',
'can',
'will',
'just',
'don',
"don't",
'should',
"should've",
'now',
'd',
'll',
'm',
'o',
're',
've',
'y',
'ain',
'aren',
"aren't",
'couldn',
"couldn't",
'didn',
"didn't",
'doesn',
"doesn't",
'hadn',
"hadn't",
'hasn',
"hasn't",
'haven',
"haven't",
'isn',
"isn't",
'ma',
'mightn',
"mightn't",
'mustn',
"mustn't",
'needn',
"needn't",
'shan',
"shan't",
'shouldn',
"shoudln't",
'wasn',
"wasn't",
'weren',
"weren't",
>won',
>won't",
>wouldn',
>wouldn't']
```

```
In [40]: len(stopwords.words('english'))
```

```
Out[40]: 179
```

```
In [41]: stopwords.words('french')
```

```
Out[41]: ['au',
 'aux',
 'avec',
 'ce',
 'ces',
 'dans',
 'de',
 'des',
 'du',
 'elle',
 'en',
 'et',
 'eux',
 'il',
 'ils',
 'je',
 'la',
 'le',
 'les',
 'leur',
 'lui',
 'ma',
 'mais',
 'me',
 'même',
 'mes',
 'moi',
 'mon',
 'ne',
 'nos',
 'notre',
 'nous',
 'on',
 'ou',
 'par',
 'pas',
 'pour',
 'qu',
 'que',
 'qui',
 'sa',
 'se',
 'ses',
 'son',
 'sur',
 'ta',
 'te',
 'tes',
 'toi',
 'ton',
 'tu',
 'un',
 'une',
 'vos',
 'votre',
 'vous',
 'c',
 'd',
 'j',
 'l',
```

'à',
'm',
'n',
's',
't',
'y',
'été',
'étée',
'étées',
'étés',
'étant',
'étante',
'étants',
'étantes',
'suis',
'es',
'est',
'sommes',
'êtes',
'sont',
'serai',
'seras',
'sera',
'serons',
'serez',
'seront',
'serais',
'serait',
'serions',
'seriez',
'seraient',
'étais',
'était',
'étions',
'étiez',
'étaient',
'fus',
'fut',
'fûmes',
'fûtes',
'furent',
'sois',
'soit',
'soyons',
'soyez',
'soient',
'fusse',
'fusses',
'fût',
'fussions',
'fussiez',
'fussent',
'ayant',
'ayante',
'ayantes',
'ayants',
'eu',
'eue',
'eues',
'eus',

```
'ai',
'as',
'avons',
'avez',
'ont',
'aurai',
'auras',
'aura',
'aurons',
'aurez',
'auront',
'aurais',
'aurait',
'aurions',
'auriez',
'auraient',
'avais',
'avait',
'avions',
'aviez',
'avaient',
'eut',
'eûmes',
'eûtes',
'eurent',
'aie',
'aies',
'ait',
'ayons',
'ayez',
'aient',
'eusse',
'eusses',
'eût',
'eussions',
'eussiez',
'eussent']
```

```
In [42]: len(stopwords.words('french'))
```

```
Out[42]: 157
```

```
In [43]: stopwords.words('german')
```

```
out[43]: ['aber',
 'alle',
 'allem',
 'allen',
 'aller',
 'alles',
 'als',
 'also',
 'am',
 'an',
 'ander',
 'andere',
 'anderem',
 'anderen',
 'anderer',
 'anderes',
 'anderm',
 'andern',
 'anderr',
 'anders',
 'auch',
 'auf',
 'aus',
 'bei',
 'bin',
 'bis',
 'bist',
 'da',
 'damit',
 'dann',
 'der',
 'den',
 'des',
 'dem',
 'die',
 'das',
 'dass',
 'daß',
 'derselbe',
 'derselben',
 'denselben',
 'desselben',
 'demselben',
 'dieselbe',
 'dieselben',
 'dasselbe',
 'dazu',
 'dein',
 'deine',
 'deinem',
 'deinen',
 'deiner',
 'deines',
 'denn',
 'derer',
 'dessen',
 'dich',
 'dir',
 'du',
 'dies',
```

'diese',
'diesem',
'diesen',
'dieser',
'dieses',
'doch',
'dort',
'durch',
'ein',
'eine',
'einem',
'einen',
'einer',
'eines',
'einig',
'einige',
'einigem',
'einigen',
'einiger',
'einiges',
'einmal',
'er',
'ihn',
'ihm',
'es',
'etwas',
'euer',
'eure',
'eurem',
'euren',
'eurer',
'eures',
'für',
'gegen',
'gewesen',
'hab',
'habe',
'haben',
'hat',
'hatte',
'hatten',
'hier',
'hin',
'hinter',
'ich',
'mich',
'mir',
'ihr',
'ihre',
'ihrem',
'ihren',
'ihrer',
'ihres',
'euch',
'im',
'in',
'indem',
'ins',
'ist',
'jede',

'jedem',
'jeden',
'jeder',
'jedes',
'jene',
'jenem',
'jenen',
'jener',
'jenes',
'jetzt',
'kann',
'kein',
'keine',
'keinem',
'keinen',
'keiner',
'keines',
'können',
'könnte',
'machen',
'man',
'manche',
'manchem',
'manchen',
'mancher',
'manches',
'mein',
'meine',
'meinem',
'meinen',
'meiner',
'meines',
'mit',
'muss',
'musste',
'nach',
'nicht',
'nichts',
'noch',
'nun',
'nur',
'ob',
'oder',
'ohne',
'sehr',
'sein',
'seine',
'seinem',
'seinen',
'seiner',
'seines',
'selbst',
'sich',
'sie',
'ihnen',
'sind',
'so',
'solche',
'solchem',
'solchen',

```
'solcher',
'solches',
'soll',
'sollte',
'sondern',
'sonst',
'über',
'um',
'und',
'uns',
'unsere',
'unserem',
'unseren',
'unser',
'unseres',
'unter',
'veiel',
'vem',
'ven',
'ver',
'vehrend',
'ver',
'veren',
'verst',
'ves',
'veg',
'veil',
'veiter',
'velche',
'velchem',
'velchen',
'velcher',
'velches',
'venn',
'verde',
'verden',
've',
'veder',
'vell',
'ver',
'verd',
'veirst',
've',
'vellen',
'vellte',
'verde',
'verden',
've',
'vem',
'ven',
'ver',
'ver',
'vear',
'veischen']
```

In [44]: `len(stopwords.words('german'))`

Out[44]: 232

In [45]: `stopwords.words('chinese')`

```
Out[45]: ['一',  
          '一下',  
          '一些',  
          '一切',  
          '一则',  
          '一天',  
          '一定',  
          '一方面',  
          '一旦',  
          '一时',  
          '一来',  
          '一样',  
          '一次',  
          '一片',  
          '一直',  
          '一致',  
          '一般',  
          '一起',  
          '一边',  
          '一面',  
          '万一',  
          '上下',  
          '上升',  
          '上去',  
          '上来',  
          '上述',  
          '上面',  
          '下列',  
          '下去',  
          '下来',  
          '下面',  
          '不一',  
          '不久',  
          '不仅',  
          '不会',  
          '不但',  
          '不光',  
          '不单',  
          '不变',  
          '不只',  
          '不可',  
          '不同',  
          '不够',  
          '不如',  
          '不得',  
          '不怕',  
          '不惟',  
          '不成',  
          '不拘',  
          '不敢',  
          '不断',  
          '不是',  
          '不比',  
          '不然',  
          '不特',  
          '不独',  
          '不管',  
          '不能',  
          '不要',  
          '不论',
```

'不足',
'不过',
'不问',
'与',
'与其',
'与否',
'与此同时',
'专门',
'且',
'两者',
'严格',
'严重',
'个',
'个人',
'个别',
'中小',
'中间',
'丰富',
'临',
'为',
'为主',
'为了',
'为什么',
'为什麼',
'为何',
'为着',
'主张',
'主要',
'举行',
'乃',
'乃至',
'么',
'之',
'之一',
'之前',
'之后',
'之後',
'之所以',
'之类',
'鸟乎',
'乎',
'乘',
'也',
'也好',
'也是',
'也罢',
'了',
'了解',
'争取',
'于',
'于是',
'于是乎',
'云云',
'互相',
'产生',
'人们',
'人家',
'什么',
'什么样',
'什麼',

'今后',
'今天',
'今年',
'今後',
'仍然',
'从',
'从事',
'从而',
'他',
'他人',
'他们',
'他的',
'代替',
'以',
'以上',
'以下',
'以为',
'以便',
'以免',
'以前',
'以及',
'以后',
'以外',
'以後',
'以来',
'以至',
'以至于',
'以致',
'们',
'任',
'任何',
'任凭',
'任务',
'企图',
'伟大',
'似乎',
'似的',
'但',
'但是',
'何',
'何况',
'何处',
'何时',
'作为',
'你',
'你们',
'你的',
'使得',
'使用',
'例如',
'依',
'依照',
'依靠',
'促进',
'保持',
'俺',
'俺们',
'倘',
'倘使',
'倘或',

'当然',
'倘若',
'假使',
'假如',
'假若',
'做到',
'像',
'允许',
'充分',
'先后',
'先後',
'先生',
'全部',
'全面',
'兮',
'共同',
'关于',
'其',
'其一',
'其中',
'其二',
'其他',
'其余',
'其它',
'其实',
'其次',
'具体',
'具体地说',
'具体说来',
'具有',
'再者',
'再说',
'冒',
'冲',
'决定',
'况且',
'准备',
'几',
'几乎',
'几时',
'凭',
'凭借',
'出去',
'出来',
'出现',
'分别',
'则',
'别',
'别的',
'别说',
'到',
'前后',
'前者',
'前进',
'前面',
'加之',
'加以',
'加入',
'加强',
'十分',

'即',
'即令',
'即使',
'即便',
'即或',
'即若',
'却不',
'原来',
'又',
'及',
'及其',
'及时',
'及至',
'双方',
'反之',
'反应',
'反映',
'反过来',
'反过来说',
'取得',
'受到',
'变成',
'另',
'另一方面',
'另外',
'只是',
'只有',
'只要',
'只限',
'叫',
'叫做',
'召开',
'叮咚',
'可',
'可以',
'可是',
'可能',
'可见',
'各',
'各个',
'各人',
'各位',
'各地',
'各种',
'各级',
'各自',
'合理',
'同',
'同一',
'同时',
'同样',
'后来',
'后面',
'向',
'向着',
'吓',
'吗',
'否则',
'吧',
'吧哒',

'哎',
'呀',
'呃',
'呕',
'呗',
'呜',
'呜呼',
'呢',
'周围',
'呵',
'呸',
'呼哧',
'咋',
'和',
'咚',
'咦',
'咱',
'咱们',
'咳',
'哇',
'哈',
'哈哈',
'哉',
'哎',
'哎呀',
'哎哟',
'哗',
'哟',
'哦',
'哩',
'哪',
'哪个',
'哪些',
'哪儿',
'哪天',
'哪年',
'哪怕',
'哪样',
'哪边',
'哪里',
'哼',
'哼唷',
'唉',
'啊',
'啐',
'啥',
'啦',
'啪达',
'喂',
'喏',
'喔唷',
'嗡嗡',
'嗬',
'嗯',
'嗳',
'嘎',
'嘎登',
'嘘',
'嘛',
'嘻',

'嘿',
'因',
'因为',
'因此',
'因而',
'固然',
'在',
'在下',
'地',
'坚决',
'坚持',
'基本',
'处理',
'复杂',
'多',
'多少',
'多数',
'多次',
'大力',
'大多数',
'大大',
'大家',
'大批',
'大约',
'大量',
'失去',
'她',
'她们',
'她的',
'好的',
'好象',
'如',
'如上所述',
'如下',
'如何',
'如其',
'如果',
'如此',
'如若',
'存在',
'宁',
'宁可',
'宁愿',
'宁肯',
'它',
'它们',
'它们的',
'它的',
'安全',
'完全',
'完成',
'实现',
'实际',
'宣布',
'容易',
'密切',
'对',
'对于',
'对应',
'将',

'少数',
'尔后',
'尚且',
'尤其',
'就',
'就是',
'就是说',
'尽',
'尽管',
'属于',
'岂但',
'左右',
'巨大',
'巩固',
'已',
'已经',
'帮助',
'常常',
'并',
'并不',
'并不是',
'并且',
'并没有',
'广大',
'广泛',
'应当',
'应用',
'应该',
'开外',
'开始',
'开展',
'引起',
'强烈',
'强调',
'归',
'当',
'当前',
'当时',
'当然',
'当着',
'形成',
'彻底',
'彼',
'彼此',
'往',
'往往',
'待',
'後來',
'後面',
'得',
'得出',
'得到',
'心里',
'必然',
'必要',
'必须',
'怎',
'怎么',
'怎么办',
'怎么样',

'怎样',
'怎麽',
'总之',
'总是',
'总的来看',
'总的来说',
'总的说来',
'总结',
'总而言之',
'恰恰相反',
'您',
'意思',
'愿意',
'慢说',
'成为',
'我',
'我们',
'我的',
'或',
'或是',
'或者',
'战斗',
'所',
'所以',
'所有',
'所谓',
'打',
'扩大',
'把',
'抑或',
'拿',
'按',
'按照',
'换句话说',
'换言之',
'据',
'掌握',
'接着',
'接著',
'故',
'故此',
'整个',
'方便',
'方面',
'旁人',
'无宁',
'无法',
'无论',
'既',
'既是',
'既然',
'时候',
'明显',
'明确',
'是',
'是否',
'是的',
'显然',
'显著',
'普通',

'普遍',
'更加',
'曾经',
'替',
'最后',
'最大',
'最好',
'最後',
'最近',
'最高',
'有',
'有些',
'有关',
'有利',
'有力',
'有所',
'有效',
'有时',
'有点',
'有的',
'有着',
'有著',
'望',
'朝',
'朝着',
'本',
'本着',
'来',
'来着',
'极了',
'构成',
'果然',
'果真',
'某',
'某个',
'某些',
'根据',
'根本',
'欢迎',
'正在',
'正如',
'正常',
'此',
'此外',
'此时',
'此间',
'毋宁',
'每',
'每个',
'每天',
'每年',
'每当',
'比',
'比如',
'比方',
'比较',
'毫不',
'没有',
'沿',
'沿着',

'注意',
'深入',
'清楚',
'满足',
'漫说',
'焉',
'然则',
'然后',
'然後',
'然而',
'照',
'照着',
'特别是',
'特殊',
'特点',
'现代',
'现在',
'甚么',
'甚而',
'甚至',
'用',
'由',
'由于',
'由此可见',
'的',
'的话',
'目前',
'直到',
'直接',
'相似',
'相信',
'相反',
'相同',
'相对',
'相对而言',
'相应',
'相当',
'相等',
'省得',
'看出',
'看到',
'看来',
'看看',
'看见',
'真是',
'真正',
'着',
'着呢',
'矣',
'知道',
'确定',
'离',
'积极',
'移动',
'突出',
'突然',
'立即',
'第',
'等',
'等等',

'管',
'紧接着',
'纵',
'纵令',
'纵使',
'纵然',
'练习',
'组成',
'经',
'经常',
'经过',
'结合',
'结果',
'给',
'绝对',
'继续',
'继而',
'维持',
'综上所述',
'罢了',
'考虑',
'者',
'而',
'而且',
'而况',
'而外',
'而已',
'而是',
'而言',
'联系',
'能',
'能否',
'能够',
'腾',
'自',
'自个儿',
'自从',
'自各儿',
'自家',
'自己',
'自身',
'至',
'至于',
'良好',
'若',
'若是',
'若非',
'范围',
'莫若',
'获得',
'虽',
'虽则',
'虽然',
'虽说',
'行为',
'行动',
'表明',
'表示',
'被',
'要',

'要不',
'要不是',
'要不然',
'要么',
'要是',
'要求',
'规定',
'觉得',
'认为',
'认真',
'认识',
'让',
'许多',
'论',
'设使',
'设若',
'该',
'说明',
'诸位',
'谁',
'谁知',
'赶',
'起',
'起来',
'起见',
'趁',
'趁着',
'越是',
'跟',
'转动',
'转变',
'转贴',
'较',
'较之',
'边',
'达到',
'迅速',
'过',
'过去',
'过来',
'运用',
'还是',
'还有',
'这',
'这个',
'这么',
'这么些',
'这样',
'这么点儿',
'这些',
'这会儿',
'这儿',
'这就是说',
'这时',
'这样',
'这点',
'这种',
'这边',
'这里',
'这麼',

'进入',
'进步',
'进而',
'进行',
'连',
'连同',
'适应',
'适当',
'适用',
'逐步',
'逐渐',
'通常',
'通过',
'造成',
'遇到',
'遭到',
'避免',
'那',
'那个',
'那么',
'那么些',
'那样样',
'那些',
'那时',
'那样',
'那边',
'那里',
'那麽',
'部分',
'鄙人',
'采取',
'里面',
'重大',
'重新',
'重要',
'鉴于',
'问题',
'防止',
'阿',
'附近',
'限制',
'除',
'除了',
'除此之外',
'除非',
'随',
'随着',
'随著',
'集中',
'需要',
'非但',
'非常',
'非徒',
'靠',
'顺',
'顺着',
'首先',

'高兴',
'是不是']

In [46]: `len(stopwords.words('chinese'))`

Out[46]: 841

In [48]: `stopwords.words('hindi') #research phase`

In []: `stopwords.words('telugu')`

In []: `import nltk
from nltk.corpus import stopwords`

In []: