Example of corpus-raw text preprocessing

In [1]:
```
import nltk
#nltk.download('stopwords') # download and update package stopwords
from nltk.corpus import stopwords
from nltk.tokenize import WordPunctTokenizer
from nltk.stem.porter import PorterStemmer

from nltk.tokenize import RegexpTokenizer
```

In [2]:
```
stemmer = PorterStemmer()

raw_text = """ The next preprocessing step is breaking up the units of text into individual words or tokens.
               earlier, stopwords are very common words. Words like "we" and "are" probably do not help at
               such as sentiment analysis or text classifications. Hence, we can remove stopwords to save c
               efforts in processing large volumes of text. In our case, we used spaCy's inbuilt stopwords,
               but we should be cautious and modify the stopwords list accordingly. E.g.,
               for sentiment analysis, the word "not" is important in the meaning of a text such as "not go
               However, spaCy included "not" as a stopword.
           """
```

In [3]:
```
#lower case
text = raw_text.lower()
print("text :", text)
print("\n")
```

```
text :  the next preprocessing step is breaking up the units of text into individual words or tokens... as m
entioned
               earlier, stopwords are very common words. words like "we" and "are" probably do not help at
all in nlp tasks
               such as sentiment analysis or text classifications. hence, we can remove stopwords to save c
omputing time and
               efforts in processing large volumes of text. in our case, we used spacy's inbuilt stopwords,
               but we should be cautious and modify the stopwords list accordingly. e.g.,
               for sentiment analysis, the word "not" is important in the meaning of a text such as "not go
od".
               however, spacy included "not" as a stopword.


```

In [4]:
```
#tokenization
tokenizer = RegexpTokenizer(r'\w+') # remove punctuation
tokens = tokenizer.tokenize(text)
print("tokens :", tokens)
print("\n")
print("tokens[0] :", tokens[0])
```

```
tokens : ['the', 'next', 'preprocessing', 'step', 'is', 'breaking', 'up', 'the', 'units', 'of', 'text', 'int
o', 'individual', 'words', 'or', 'tokens', 'as', 'mentioned', 'earlier', 'stopwords', 'are', 'very', 'commo
n', 'words', 'words', 'like', 'we', 'and', 'are', 'probably', 'do', 'not', 'help', 'at', 'all', 'in', 'nlp',
'tasks', 'such', 'as', 'sentiment', 'analysis', 'or', 'text', 'classifications', 'hence', 'we', 'can', 'remo
ve', 'stopwords', 'to', 'save', 'computing', 'time', 'and', 'efforts', 'in', 'processing', 'large', 'volume
s', 'of', 'text', 'in', 'our', 'case', 'we', 'used', 'spacy', 's', 'inbuilt', 'stopwords', 'but', 'we', 'sho
uld', 'be', 'cautious', 'and', 'modify', 'the', 'stopwords', 'list', 'accordingly', 'e', 'g', 'for', 'sentim
ent', 'analysis', 'the', 'word', 'not', 'is', 'important', 'in', 'the', 'meaning', 'of', 'a', 'text', 'suc
h', 'as', 'not', 'good', 'however', 'spacy', 'included', 'not', 'as', 'a', 'stopword']


tokens[0] : the
```

In [5]:
```
stopWords = set(stopwords.words('english'))
stopWords
```

Out[5]:
```
{'a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'ain',
 'all',
 'am',
 'an',
 'and',
 'any',
 'are',
 'aren',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 'couldn',
 "couldn't",
 'd',
 'did',
 'didn',
 "didn't",
 'do',
 'does',
 'doesn',
 "doesn't",
 'doing',
 'don',
 "don't",
 'down',
 'during',
 'each',
 'few',
 'for',
 'from',
 'further',
 'had',
 'hadn',
 "hadn't",
 'has',
 'hasn',
 "hasn't",
 'have',
 'haven',
 "haven't",
 'having',
 'he',
 'her',
 'here',
 'hers',
 'herself',
 'him',
 'himself',
 'his',
 'how',
 'i',
 'if',
 'in',
 'into',
 'is',
 'isn',
 "isn't",
 'it',
 "it's",
 'its',
 'itself',
 'just',
 'll',
 'm',
 'ma',
 'me',
 'mightn',
 "mightn't",
 'more',
 'most',
 'mustn',
 "mustn't",
 'my',
 'myself',
 'needn',
 "needn't",
 'no',
 'nor',
 'not',
 'now',
 'o',
 'of',
 'off',
 'on',
 'once',
 'only',
 'or',
 'other',
 'our',
 'ours',
 'ourselves',
 'out',
 'over',
 'own',
 're',
 's',
 'same',
 'shan',
 "shan't",
 'she',
 "she's",
 'should',
 "should've",
 'shouldn',
 "shouldn't",
 'so',
 'some',
 'such',
 't',
 'than',
 'that',
 "that'll",
 'the',
 'their',
 'theirs',
 'them',
 'themselves',
 'then',
 'there',
 'these',
 'they',
 'this',
 'those',
 'through',
 'to',
 'too',
 'under',
 'until',
 'up',
 've',
 'very',
 'was',
 'wasn',
 "wasn't",
 'we',
 'were',
 'weren',
 "weren't",
 'what',
 'when',
 'where',
 'which',
 'while',
 'who',
 'whom',
 'why',
 'will',
 'with',
 'won',
 "won't",
 'wouldn',
 "wouldn't",
 'y',
 'you',
 "you'd",
 "you'll",
 "you're",
 "you've",
 'your',
 'yours',
 'yourself',
 'yourselves'}
```

In [6]:
```
# stopwords removal
tokens_clean = [t for t in tokens if t not in stopWords]
print("\n tokens_clean : ", tokens_clean)
```

```
 tokens_clean :  ['text', 'preprocessing', 'step', 'breaking', 'units', 'text', 'individual', 'words', 'toke
ns', 'mentioned', 'earlier', 'stopwords', 'common', 'words', 'words', 'like', 'probably', 'help', 'nlp', 'ta
sks', 'sentiment', 'analysis', 'text', 'classifications', 'hence', 'remove', 'stopwords', 'save', 'computin
g', 'time', 'efforts', 'processing', 'large', 'volumes', 'text', 'case', 'used', 'spacy', 'inbuilt', 'stopwo
rds', 'cautious', 'modify', 'stopwords', 'list', 'accordingly', 'e', 'g', 'sentiment', 'analysis', 'word',
'important', 'meaning', 'text', 'good', 'however', 'spacy', 'included', 'stopword']
```

In [7]:
```
#stemming
stems = [stemmer.stem(t) for t in tokens_clean]
print("\n stems : " ,stems)
```

```
 stems :  ['next', 'preprocess', 'step', 'break', 'unit', 'text', 'individu', 'word', 'token', 'mention', 'e
arlier', 'stopword', 'common', 'word', 'word', 'like', 'probabl', 'help', 'nlp', 'task', 'sentiment', 'analy
si', 'text', 'classif', 'henc', 'remov', 'stopword', 'save', 'comput', 'time', 'effort', 'process', 'larg',
'volum', 'text', 'case', 'use', 'spaci', 'inbuilt', 'stopword', 'cautiou', 'modifi', 'stopword', 'list', 'ac
cordingli', 'e', 'g', 'sentiment', 'analysi', 'word', 'import', 'mean', 'text', 'good', 'howev', 'spaci', 'i
nclud', 'stopword']
```

In [ ]:
```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js