(Exercise 1: Do you remember how to check python version andinstalled libraries? Write steps below:)

In [ ]:
```
import numpy
numpy.version.version
```

In [1]:
```
from platform import python_version
print(python_version())
```

```
3.7.4
```

In [6]:
```
pip install gensim
```

```
Requirement already satisfied: gensim in d:\users\abadi\anaconda3\lib\site-pack
ages (3.8.3)
Requirement already satisfied: scipy>=0.18.1 in d:\users\abadi\anaconda3\lib\si
te-packages (from gensim) (1.3.1)
Requirement already satisfied: Cython==0.29.14 in d:\users\abadi\anaconda3\lib
\site-packages (from gensim) (0.29.14)
Requirement already satisfied: numpy>=1.11.3 in d:\users\abadi\anaconda3\lib\si
te-packages (from gensim) (1.16.5)
Requirement already satisfied: six>=1.5.0 in d:\users\abadi\anaconda3\lib\site-
packages (from gensim) (1.12.0)
Requirement already satisfied: smart-open>=1.8.1 in d:\users\abadi\anaconda3\li
b\site-packages (from gensim) (2.1.1)
Requirement already satisfied: requests in d:\users\abadi\anaconda3\lib\site-pa
ckages (from smart-open>=1.8.1->gensim) (2.22.0)
Requirement already satisfied: boto in d:\users\abadi\anaconda3\lib\site-packag
es (from smart-open>=1.8.1->gensim) (2.49.0)
Requirement already satisfied: boto3 in d:\users\abadi\anaconda3\lib\site-packa
ges (from smart-open>=1.8.1->gensim) (1.14.60)
Requirement already satisfied: certifi>=2017.4.17 in d:\users\abadi\anaconda3\l
ib\site-packages (from requests->smart-open>=1.8.1->gensim) (2019.9.11)
Requirement already satisfied: idna<2.9,>=2.5 in d:\users\abadi\anaconda3\lib\s
ite-packages (from requests->smart-open>=1.8.1->gensim) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in d:\us
ers\abadi\anaconda3\lib\site-packages (from requests->smart-open>=1.8.1->gensi
m) (1.24.2)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in d:\users\abadi\anaconda
3\lib\site-packages (from requests->smart-open>=1.8.1->gensim) (3.0.4)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in d:\users\abadi\anacond
a3\lib\site-packages (from boto3->smart-open>=1.8.1->gensim) (0.10.0)
Requirement already satisfied: botocore<1.18.0,>=1.17.60 in d:\users\abadi\anac
onda3\lib\site-packages (from boto3->smart-open>=1.8.1->gensim) (1.17.60)
Requirement already satisfied: s3transfer<0.4.0,>=0.3.0 in d:\users\abadi\anaco
nda3\lib\site-packages (from boto3->smart-open>=1.8.1->gensim) (0.3.3)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in d:\users\abadi\an
aconda3\lib\site-packages (from botocore<1.18.0,>=1.17.60->boto3->smart-open>=
1.8.1->gensim) (2.8.0)
Requirement already satisfied: docutils<0.16,>=0.10 in d:\users\abadi\anaconda3
\lib\site-packages (from botocore<1.18.0,>=1.17.60->boto3->smart-open>=1.8.1->g
ensim) (0.15.2)
Note: you may need to restart the kernel to use updated packages.
```

In [7]:
```
import gensim as gs
print(gs.__version__)
```

```
3.8.3
```

Exercise 2: Tokenize the following sentence and write down the result you obtain!

In [27]:
```
Sentence= 'Tokenization is the process of breaking down text document apart into
print(Sentence)
```

```
Tokenization is the process of breaking down text document apart into those pie
ces
```

In [28]:
```
import gensim as gs
tokenizedWord = list(gs.utils.tokenize(Sentence))
```

In [29]:
```
tokenizedWord
```

Out[29]:
```
['Tokenization',
 'is',
 'the',
 'process',
 'of',
 'breaking',
 'down',
 'text',
 'document',
 'apart',
 'into',
 'those',
 'pieces']
```

In [30]:
```python
gs.utils.tokenize
help(gs.utils.tokenize)
```

```
Help on function tokenize in module gensim.utils:

tokenize(text, lowercase=False, deacc=False, encoding='utf8', errors='strict',
to_lower=False, lower=False)
    Iteratively yield tokens as unicode strings, optionally removing accent mar
ks and lowercasing it.

    Parameters
    ----------
    text : str or bytes
        Input string.
    deacc : bool, optional
        Remove accentuation using :func:`~gensim.utils.deaccent`?
    encoding : str, optional
        Encoding of input string, used as parameter for :func:`~gensim.utils.to
_unicode`.
    errors : str, optional
        Error handling behaviour, used as parameter for :func:`~gensim.utils.to
_unicode`.
    lowercase : bool, optional
        Lowercase the input string?
    to_lower : bool, optional
        Same as `lowercase`. Convenience alias.
    lower : bool, optional
        Same as `lowercase`. Convenience alias.

    Yields
    ------
    str
        Contiguous sequences of alphabetic characters (no digits!), using :fun
c:`~gensim.utils.simple_tokenize`

    Examples
    --------
    .. sourcecode:: pycon

        >>> from gensim.utils import tokenize
        >>> list(tokenize('Nic nemůže letět rychlostí vyšší, než 300 tisíc kilo
metrů za sekundu!', deacc=True))
        [u'Nic', u'nemuze', u'letet', u'rychlosti', u'vyssi', u'nez', u'tisic',
u'kilometru', u'za', u'sekundu']
```

In [34]:
```
import gensim
from gensim import corpora
from pprint import pprint
text = ["""In computer science, artificial intelligence (AI), sometimes
called machine intelligence, is intelligence demonstrated by machines, in
contrast to the natural intelligence displayed by humans and animals. Compute
r science defines AI research as the study of intelligent agents: any device th
at perceives its environment and takes actions that maximize its chance of succe
achieving its goals."""]
tokens = [[token for token in sentence.split()] for sentence in text]
gensim_dictionary = corpora.Dictionary()
gensim_corpus = [gensim_dictionary.doc2bow(token, allow_update=True) for token i
print(gensim_corpus)
```

```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 2), (8, 1), (9,
1), (10, 1), (11, 1), (12, 1), (13, 2), (14, 1), (15, 1), (16, 1), (17, 1), (1
8, 1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1), (25, 1), (26, 3),
(27, 1), (28, 1), (29, 1), (30, 3), (31, 1), (32, 1), (33, 1), (34, 1), (35,
2), (36, 1), (37, 1), (38, 1), (39, 1), (40, 1), (41, 1), (42, 1), (43, 1), (4
4, 1), (45, 1), (46, 1), (47, 2), (48, 1)]]
```

What do you see?

It shows the index for the word and the number of times it is repeated in the text, but without appearing what the word is, we have a key and index

In [36]:
```
word_frequencies = [[(gensim_dictionary[id], frequence) for id, frequence in cou
print(word_frequencies)
```

```
[[('(AI),', 1), ('AI', 1), ('Compute', 1), ('In', 1), ('achieving', 1), ('actio
ns', 1), ('agents:', 1), ('and', 2), ('animals.', 1), ('any', 1), ('artificia
l', 1), ('as', 1), ('at', 1), ('by', 2), ('called', 1), ('chance', 1), ('comput
er', 1), ('contrast', 1), ('defines', 1), ('demonstrated', 1), ('device', 1),
('displayed', 1), ('environment', 1), ('goals.', 1), ('humans', 1), ('in', 1),
('intelligence', 3), ('intelligence,', 1), ('intelligent', 1), ('is', 1), ('it
s', 3), ('machine', 1), ('machines,', 1), ('maximize', 1), ('natural', 1), ('o
f', 2), ('perceives', 1), ('r', 1), ('research', 1), ('science', 1), ('scienc
e,', 1), ('sometimes', 1), ('study', 1), ('successfully', 1), ('takes', 1), ('t
h', 1), ('that', 1), ('the', 2), ('to', 1)]]
```

Home exercise: Create a bag of words corpus by reading a text file.

In [51]:
```
from gensim.utils import simple_preprocess
from smart_open import smart_open
import os
tokens = [simple_preprocess(sentence, deacc=True) for sentence in open(r'D:\User
gensim_dictionary = corpora.Dictionary()
gensim_corpus = [gensim_dictionary.doc2bow(token, allow_update=True) for token i
word_frequencies = [[(gensim_dictionary[id], frequence) for id, frequence in cou

print(word_frequencies)
```

```
[[('achieving', 1), ('actions', 1), ('agents', 1), ('ai', 2), ('and', 2), ('ani
mals', 1), ('any', 1), ('artificial', 1), ('as', 1), ('by', 2), ('called', 1),
('chance', 1), ('computer', 2), ('contrast', 1), ('defines', 1), ('demonstrate
d', 1), ('device', 1), ('displayed', 1), ('environment', 1), ('goals', 1), ('hu
mans', 1), ('in', 2), ('intelligence', 4), ('intelligent', 1), ('is', 1), ('it
s', 3), ('machine', 1), ('machines', 1), ('maximize', 1), ('natural', 1), ('o
f', 2), ('perceives', 1), ('research', 1), ('science', 2), ('sometimes', 1),
('study', 1), ('successfully', 1), ('takes', 1), ('that', 2), ('the', 2), ('t
o', 1)]]
```