In [2]:

```
!pip install Gensim # install gnsim
```

```
Requirement already satisfied: Gensim in g:\anacondaprogram\lib\site-p
(ackages (3.8.3
Requirement already satisfied: six>=1.5.0 in g:\anacondaprogram\lib\si
(te-packages (from Gensim) (1.15.0
Requirement already satisfied: scipy>=0.18.1 in g:\anacondaprogram\lib
(\site-packages (from Gensim) (1.5.0
Requirement already satisfied: numpy>=1.11.3 in g:\anacondaprogram\lib
(\site-packages (from Gensim) (1.18.5
Requirement already satisfied: smart-open>=1.8.1 in g:\anacondaprogram
(\lib\site-packages (from Gensim) (2.1.1
Requirement already satisfied: Cython==0.29.14 in g:\anacondaprogram\l
(ib\site-packages (from Gensim) (0.29.14
Requirement already satisfied: requests in g:\anacondaprogram\lib\site
(-packages (from smart-open>=1.8.1->Gensim) (2.24.0
Requirement already satisfied: boto in g:\anacondaprogram\lib\site-pac
(kages (from smart-open>=1.8.1->Gensim) (2.49.0
Requirement already satisfied: boto3 in g:\anacondaprogram\lib\site-pa
(ckages (from smart-open>=1.8.1->Gensim) (1.15.1
 Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in g:\anacondaprogram\lib\site-packages (from requests->smart-open>=1.
(8.1->Gensim) (1.25.9
Requirement already satisfied: idna<3,>=2.5 in g:\anacondaprogram\lib
(\site-packages (from requests->smart-open>=1.8.1->Gensim) (2.10
Requirement already satisfied: certifi>=2017.4.17 in g:\anacondaprogra
m\lib\site-packages (from requests->smart-open>=1.8.1->Gensim) (2020.
(6.20
Requirement already satisfied: chardet<4,>=3.0.2 in g:\anacondaprogram
(\lib\site-packages (from requests->smart-open>=1.8.1->Gensim) (3.0.4
Requirement already satisfied: botocore<1.19.0,>=1.18.1 in g:\anaconda
program\lib\site-packages (from boto3->smart-open>=1.8.1->Gensim) (1.1
(8.1
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in g:\anacondapr
ogram\lib\site-packages (from boto3->smart-open>=1.8.1->Gensim) (0.10.
(0
Requirement already satisfied: s3transfer<0.4.0,>=0.3.0 in g:\anaconda
program\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 g:\anaco
ndaprogram\lib\site-packages (from botocore<1.19.0,>=1.18.1->boto3->sm
(art-open>=1.8.1->Gensim) (2.8.1
```

In [3]:

```
from platform import python_version
print(python_version()) #python version lab1 ex 1
```

```
3.8.3
```

In [4]:

```
import gensim as gs
print(gs.__version__)
```

```
3.8.3
```

In [6]:

```
Sentence= "Tokenization is the process of breaking down text document apart into th
```

In [7]:

```
import gensim as gs
tokenizedWord= list(gs.utils.tokenize(Sentence)) # ex 2
```

In [8]:

```
tokenizedWord
```

Out[8]:

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

In [9]:

```
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 a
ccent marks 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:`~gensi
m.utils.to_unicode`.
errors : str, optional
Error handling behaviour, used as parameter for :func:`~gensi
m.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!), us
ing :func:`~gensim.utils.simple_tokenize`

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

from gensim.utils import tokenize >>>
list(tokenize('Nic nemůže letět rychlostí vyšší, než 300 t >>>
((isíc kilometrů 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 [10]:

```
Sentence= "In computer science, artificial intelligence (AI), sometimes called mach
```

In [15]:

```
import gensim
from gensim import corpora
from pprint import pprint
```

In [16]:

```
y device that perceives its environment and takes actions that maximize its chance o
```

In [17]:

```
tokens = [[token for token in sentence.split()] for sentence
in text]
```

In [18]:

```
gensim_dictionary = corpora.Dictionary()
```

In [19]:

```
gensim_corpus = [gensim_dictionary.doc2bow(token,
allow_update=True) for token in tokens]
print(gensim_corpus)
```

```
 ,8) ,(2 ,7) ,(1 ,6) ,(1 ,5) ,(1 ,4) ,(1 ,3) ,(1 ,2) ,(1 ,1) ,(1 ,0)]]
 ,16) ,(1 ,15) ,(1 ,14) ,(1 ,13) ,(2 ,12) ,(1 ,11) ,(1 ,10) ,(1 ,9) ,(1
2) ,(1 ,23) ,(1 ,22) ,(1 ,21) ,(1 ,20) ,(1 ,19) ,(1 ,18) ,(1 ,17) ,(1
 ,(1 ,31) ,(1 ,30) ,(3 ,29) ,(1 ,28) ,(1 ,27) ,(1 ,26) ,(3 ,25) ,(1 ,4
 ,39) ,(1 ,38) ,(1 ,37) ,(1 ,36) ,(1 ,35) ,(2 ,34) ,(1 ,33) ,(1 ,32)
[[(1 ,45) ,(2 ,44) ,(2 ,43) ,(1 ,42) ,(1 ,41) ,(1 ,40) ,(1
```

In [20]:

```
print(gensim_dictionary)
```

```
Dictionary(46 unique tokens: ['(AI),', 'AI', 'Computer', 'In', 'achiev
(...['ing
```

In [23]:

```
word_frequencies = [[(gensim_dictionary[id], frequence) for id, frequence in couple
for couple in gensim_corpus]
print(word_frequencies)
```

```
 AI),', 1), ('AI', 1), ('Computer', 1), ('In', 1), ('achieving',)')]]
1), ('actions', 1), ('agents:', 1), ('and', 2), ('animals.', 1), ('an
y', 1), ('artificial', 1), ('as', 1), ('by', 2), ('called', 1), ('chan
ce', 1), ('computer', 1), ('contrast', 1), ('defines', 1), ('demonstra
ted', 1), ('device', 1), ('displayed', 1), ('environment', 1), ('goal
s.', 1), ('humans', 1), ('in', 1), ('intelligence', 3), ('intelligenc
 e,', 1), ('intelligent', 1), ('is', 1), ('its', 3), ('machine', 1),
('machines,', 1), ('maximize', 1), ('natural', 1), ('of', 2), ('percei
ves', 1), ('research', 1), ('science', 1), ('science,', 1), ('sometime
 s', 1), ('study', 1), ('successfully', 1), ('takes', 1), ('that', 2),
[[(('the', 2), ('to', 1
```

In [31]:

```
from gensim.utils import simple_preprocess
from smart_open import smart_open
import osfaisal.txt
```

In [43]:

```
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'faisal.txt
gensim_dictionary = corpora.Dictionary()
gensim_corpus = [gensim_dictionary.doc2bow(token, allow_update=True) for token in t
word_frequencies = [[(gensim_dictionary[id], frequence) for id, frequence in couple
for couple in gensim_corpus]
print(word_frequencies)
```

```
ai', 1), ('artificial', 1), ('computer', 1), ('in', 1), ('intell')]]
igence', 1), ('science', 1), ('sometimes', 1)], [('in', 1), ('intell
 igence', 2), ('by', 1), ('called', 1), ('demonstrated', 1), ('is',
 1), ('machine', 1), ('machines', 1)], [('intelligence', 1), ('by',
1), ('and', 1), ('animals', 1), ('compute', 1), ('contrast', 1), ('d
 isplayed', 1), ('humans', 1), ('natural', 1), ('the', 1), ('to',
 1)], [('ai', 1), ('science', 1), ('the', 1), ('agents', 1), ('any',
 1), ('as', 1), ('defines', 1), ('device', 1), ('intelligent', 1),
 ('of', 1), ('research', 1), ('study', 1), ('th', 1)], [('and', 1),
('of', 1), ('actions', 1), ('at', 1), ('chance', 1), ('environment',
 1), ('its', 2), ('maximize', 1), ('perceives', 1), ('successfully',
1), ('takes', 1), ('that', 1)], [('its', 1), ('achieving', 1), ('goa
[[(ls', 1
```

In [ ]:

```

```