

```
In [1]:  
import spacy  
nlp = spacy.load("en_core_web_sm")  
doc = nlp("data science and ai has greate career ahead")
```

```
In [2]: doc
```

```
Out[2]: data science and ai has greate career ahead
```

```
In [4]:  
for token in doc:  
    print(token.text)
```

```
data  
science  
and  
ai  
has  
greate  
career  
ahead
```

```
In [5]: doc
```

```
Out[5]: data science and ai has greate career ahead
```

```
In [6]:  
for token in doc:  
    print(token.text)
```

```
data  
science  
and  
ai  
has  
greate  
career  
ahead
```

```
In [8]: import spacy
```

```
nlp = spacy.load("en_core_web_sm")  
doc = nlp("Apple is looking at buying UK startup for 1$ billon")  
  
for token in doc:  
    print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,  
          token.shape_, token.is_alpha, token.is_stop)
```

```
Apple Apple PROPN NNP 429 Xxxxx True False
is be AUX VBZ 405 xx True True
looking look VERB VBG 8206900633647566924 xxxx True False
at at ADP IN 443 xx True True
buying buy VERB VBG 438 xxxx True False
UK UK PROPN NNP 7037928807040764755 XX True False
startup startup NOUN NN 416 xxxx True False
for for ADP IN 443 xxx True True
1 1 NUM CD 12837356684637874264 d False False
$ $ SYM $ 12837356684637874264 $ False False
billon billion NOUN NN 439 xxxx True False
```

In [9]:

```
for token in doc:
    print(token.pos_)
```

```
PROPN
AUX
VERB
ADP
VERB
PROPN
NOUN
ADP
NUM
SYM
NOUN
```

In [10]:

```
for token in doc:
    print(token.text,token.pos_)
```

```
Apple PROPN
is AUX
looking VERB
at ADP
buying VERB
UK PROPN
startup NOUN
for ADP
1 NUM
$ SYM
billon NOUN
```

In [11]:

```
for token in doc:
    print(token.text,token.pos_,token.lemma_)
```

```
Apple PROPN Apple
is AUX be
looking VERB look
at ADP at
buying VERB buy
UK PROPN UK
startup NOUN startup
for ADP for
1 NUM 1
$ SYM $
billon NOUN billon
```

In [12]:

```
text = """There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs. An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary. Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured """
```

In [13]:

```
text
```

Out[13]: 'There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.\nAn example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.\nImage collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured '

In [15]:

```
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
```

In [16]:

```
stopwords = list(STOP_WORDS)
stopwords
```

```
Out[16]: ['ourselves',
'someone',
'own',
'once',
'please',
'out',
'do',
'nine',
'noone',
"'m",
'can',
'two',
'already',
'sometimes',
'to',
'side',
'back',
'put',
'so',
'one',
'therefore',
'afterwards',
'while',
'though',
'quite',
'he',
'former',
'seem',
'behind',
'yourself',
'over',
'him',
'hereupon',
're',
'whereas',
'your',
'moreover',
'thus',
'in',
'full',
'this',
'seeming',
'bottom',
'whose',
'his',
'of',
'four',
'least',
'thereby',
'i',
'and',
'there',
"'ve",
'`ve',
'therein',
```

```
're',
'enough',
'very',
'latterly',
'had',
'below',
'myself',
'being',
'whether',
'by',
'then',
've',
'hundred',
'how',
"'re",
'hereby',
'them',
'these',
'beside',
'an',
'itself',
'what',
'wherever',
'nor',
'anything',
'is',
'amongst',
'anyway',
'thereafter',
'each',
'really',
'am',
'whatever',
'twelve',
'veia',
'get',
'done',
'rather',
'doing',
'without',
'none',
'together',
'except',
'or',
'since',
'may',
'move',
'a',
'empty',
'last',
'with',
'alone',
'whence',
'on',
'go',
```

```
'call',
'nothing',
'regarding',
'now',
'also',
'fifty',
'‘d',
'hereafter',
'three',
'off',
'ever',
'much',
'thence',
'anyhow',
'becoming',
'often',
'until',
'herself',
'others',
'could',
'nevertheless',
'either',
'made',
'you',
'has',
'keep',
'during',
'seemed',
'must',
'among',
'wherein',
'under',
'towards',
'top',
'through',
'otherwise',
'first',
'here',
'due',
'meanwhile',
'after',
'only',
'used',
'mostly',
'part',
'we',
'n’t',
'my',
'seems',
'between',
'somehow',
'next',
'‘ll',
'himself',
'whereby',
```

```
'about',
'themselves',
'might',
'all',
'"m',
'six',
'namely',
'five',
'ours',
'see',
'yet',
'beforehand',
'us',
'did',
'some',
'other',
'thru',
'further',
'beyond',
'does',
'sometime',
'their',
'whereupon',
'same',
'but',
'when',
'several',
'whoever',
'she',
'nobody',
'toward',
'show',
'any',
'were',
'name',
'anywhere',
'was',
'than',
'give',
'against',
'still',
'onto',
'across',
'at',
'formerly',
'however',
'if',
'neither',
'serious',
'"d",
'those',
'as',
'throughout',
'hers',
"n't",
```

```
'they',
'along',
'up',
'else',
'well',
'every',
'third',
'cannot',
'take',
"'s",
'twenty',
'why',
'who',
'became',
'would',
'becomes',
'less',
'perhaps',
'thereupon',
'have',
'most',
'such',
'mine',
'"m',
'both',
'indeed',
'within',
'nowhere',
'another',
'that',
'hence',
'"ll',
'around',
'will',
'everything',
'many',
'because',
'whole',
'fifteen',
'too',
'be',
'been',
'for',
'not',
'amount',
'ca',
'almost',
'everywhere',
'its',
'make',
'front',
'sixty',
'her',
'whom',
'again',
```

```
's',
'which',
'say',
'besides',
'latter',
'whither',
'something',
'just',
'never',
'are',
'ten',
'should',
'd',
'more',
'before',
'eleven',
'no',
'verious',
'anyone',
'our',
'n't',
's',
'become',
'per',
'unless',
'down',
're',
'using',
'eight',
'the',
'"ll',
'forty',
'always',
'yourselves',
'herein',
'somewhere',
'above',
'into',
'whenever',
'me',
'everyone',
'although',
'yours',
'upon',
'it',
'few',
'where',
'elsewhere',
'whereafter',
'even',
'from']
```

In [17]: `len(stopwords)`

Out[17]: 326

In [18]: `nlp = spacy.load('en_core_web_sm')`

In [19]: `text`

Out[19]: 'There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.\nAn example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.\nImage collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured '

In [20]: `doc = nlp(text)`

In [21]: `doc`

Out[21]: There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.

An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured

In [22]:

```
tokens = [token.text for token in doc]
print(tokens)
```

```
['There', 'are', 'broadly', 'two', 'types', 'of', 'extractive', 'summarization', 'tasks', 'depending', 'on', 'what', 'the', 'summarization', 'program', 'focuses', 'on', '.', 'The', 'first', 'is', 'generic', 'summarization', ',', 'which', 'focuses', 'on', 'obtaining', 'a', 'generic', 'summary', 'or', 'abstract', 'of', 'the', 'collection', '(', 'whether', 'documents', ',', 'or', 'sets', 'of', 'images', ',', 'or', 'videos', ',', 'news', 'stories', 'etc', '.', ')', '.', 'The', 'second', 'is', 'query', 'relevant', 'summarization', ',', 'sometimes', 'called', 'query', '--', 'based', 'summarization', ',', 'which', 'summarizes', 'objects', 'specific', 'to', 'a', 'query', '.', 'Summarization', 'systems', 'are', 'able', 'to', 'create', 'both', 'query', 'relevant', 'text', 'summaries', 'and', 'generic', 'machine', '--', 'generated', 'summaries', 'depending', 'on', 'what', 'the', 'user', 'needs', '.', '\n', 'An', 'example', 'of', 'a', 'summarization', 'problem', 'is', 'document', 'summarization', ',', 'which', 'attempts', 'to', 'automatically', 'produce', 'an', 'abstract', 'from', 'a', 'given', 'document', '.', 'Sometimes', 'one', 'might', 'be', 'interested', 'in', 'generating', 'a', 'summary', 'from', 'a', 'single', 'source', 'document', ',', 'while', 'others', 'can', 'use', 'multiple', 'source', 'documents', '(', 'for', 'example', ',', 'a', 'cluster', 'of', 'articles', 'on', 'the', 'same', 'topic', ')', '.', 'This', 'problem', 'is', 'called', 'multi', '--', 'document', 'summarization', '.', 'A', 'related', 'application', 'is', 'summarizing', 'news', 'articles', '.', 'Imagine', 'a', 'system', ',', 'which', 'automatically', 'pulls', 'together', 'news', 'articles', 'on', 'a', 'given', 'topic', '(', 'from', 'the', 'web', ')', ',', 'and', 'concisely', 'represents', 'the', 'latest', 'news', 'as', 'a', 'summary', '.', '\n', 'Image', 'collection', 'summarization', 'is', 'another', 'application', 'example', 'of', 'automatic', 'summarization', '.', 'It', 'consists', 'in', 'selecting', 'a', 'representative', 'set', 'of', 'images', 'from', 'a', 'larger', 'set', 'of', 'images.[4', ']', 'A', 'summary', 'in', 'this', 'context', 'is', 'useful', 'to', 'show', 'the', 'most', 'representative', 'images', 'of', 'results', 'in', 'an', 'image', 'collection', 'exploration', 'system', '.', 'Video', 'summarization', 'is', 'a', 'related', 'domain', ',', 'where', 'the', 'system', 'automatically', 'creates', 'a', 'trailer', 'of', 'a', 'long', 'video', '.', 'This', 'also', 'has', 'applications', 'in', 'consumer', 'or', 'personal', 'videos', ',', 'where', 'one', 'might', 'want', 'to', 'skip', 'the', 'boring', 'or', 'repetitive', 'actions', '.', 'Similarly', ',', 'in', 'surveillance', 'videos', ',', 'one', 'would', 'want', 'to', 'extract', 'important', 'and', 'suspicious', 'activity', ',', 'while', 'ignoring', 'all', 'the', 'boring', 'and', 'redundant', 'frames', 'captured']
```

In [23]:

tokens

```
Out[23]: ['There',
  'are',
  'broadly',
  'two',
  'types',
  'of',
  'extractive',
  'summarization',
  'tasks',
  'depending',
  'on',
  'what',
  'the',
  'summarization',
  'program',
  'focuses',
  'on',
  '.',
  'The',
  'first',
  'is',
  'generic',
  'summarization',
  ',',
  'which',
  'focuses',
  'on',
  'obtaining',
  'a',
  'generic',
  'summary',
  'or',
  'abstract',
  'of',
  'the',
  'collection',
  '(',
  'whether',
  'documents',
  ',',
  'or',
  'sets',
  'of',
  'images',
  ',',
  'or',
  'videos',
  ',',
  'news',
  'stories',
  'etc',
  '.',
  ')',
  '.',
  'The',
```

```
'second',
'is',
'query',
'relevant',
'summarization',
',',
'sometimes',
'called',
'query',
'-',
'based',
'summarization',
',',
'which',
'summarizes',
'objects',
'specific',
'to',
'a',
'query',
'.',
'Summarization',
'systems',
'are',
'able',
'to',
'create',
'both',
'query',
'relevant',
'text',
'summaries',
'and',
'generic',
'machine',
'-',
'generated',
'summaries',
'depending',
'on',
'what',
'the',
'user',
'needs',
'.',
'\n',
'An',
'example',
'of',
'a',
'summarization',
'problem',
'is',
'document',
'summarization',
```

',
'which',
'attempts',
'to',
'automatically',
'produce',
'an',
'abstract',
'from',
'a',
'given',
'document',
'.',
'Sometimes',
'one',
'might',
'be',
'interested',
'in',
'generating',
'a',
'summary',
'from',
'a',
'single',
'source',
'document',
'',
'',
'while',
'others',
'can',
'use',
'multiple',
'source',
'documents',
'(',
'for',
'example',
',',
'a',
'cluster',
'of',
'articles',
'on',
'the',
'same',
'topic',
)',
'.',
'This',
'problem',
'is',
'called',
'multi',
'-',

```
'document',
'summarization',
'.',
'A',
'related',
'application',
'is',
'summarizing',
'news',
'articles',
'.',
'Imagine',
'a',
'system',
',',
'which',
'automatically',
'pulls',
'together',
'news',
'articles',
'on',
'a',
'given',
'topic',
'(',
'from',
'the',
'web',
')',
',',
'and',
'concisely',
'represents',
'the',
'latest',
'news',
'as',
'a',
'summary',
'.',
'\n',
'Image',
'collection',
'summarization',
'is',
'another',
'application',
'example',
'of',
'automatic',
'summarization',
'.',
'It',
'consists',
```

```
'in',
'selecting',
'a',
'representative',
'set',
'of',
'images',
'from',
'a',
'larger',
'set',
'of',
'images.[4',
']',
'A',
'summary',
'in',
'this',
'context',
'is',
'useful',
'to',
'show',
'the',
'most',
'representative',
'images',
'of',
'results',
'in',
'an',
'image',
'collection',
'exploration',
'system',
'.',
'Video',
'summarization',
'is',
'a',
'related',
'domain',
',',
'where',
'the',
'system',
'automaticaly',
'creates',
'a',
'trailer',
'of',
'a',
'long',
'video',
'.',
```

```
'This',
'also',
'has',
'applications',
'in',
'consumer',
'or',
'personal',
'videos',
',',
'where',
'one',
'might',
'want',
'to',
'skip',
'the',
'boring',
'or',
'repetitive',
'actions',
'.',
'.',
'Similarly',
',',
'in',
'surveillance',
'videos',
',',
'one',
'would',
'want',
'to',
'extract',
'important',
'and',
'suspicious',
'activity',
',',
'while',
'ignoring',
'all',
'the',
'boring',
'and',
'redundant',
'frames',
'captured']
```

In [24]: `len(tokens)`

Out[24]: 322

In [25]: `punctuation`

```
Out[25]: '!"#$%&\'()*)+, -./:; <=>?@[\\\]^_`{|}~'
```

```
In [40]: doc
```

Out[40]: There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.

An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured

```
In [43]: word_frequencies = {}
for word in doc:
    if word.text.lower() not in stopwords:
        if word.text.lower() not in punctuation:
            if word.text not in word_frequencies.keys():
                word_frequencies[word.text] = 1
            else:
                word_frequencies[word.text] +=1
```

```
In [44]: word_frequencies
```

```
Out[44]: {'broadly': 1,
 'types': 1,
 'extractive': 1,
 'summarization': 11,
 'tasks': 1,
 'depending': 2,
 'program': 1,
 'focuses': 2,
 'generic': 3,
 'obtaining': 1,
 'summary': 4,
 'abstract': 2,
 'collection': 3,
 'documents': 2,
 'sets': 1,
 'images': 3,
 'videos': 3,
 'news': 4,
 'stories': 1,
 'etc': 1,
 'second': 1,
 'query': 4,
 'relevant': 2,
 'called': 2,
 'based': 1,
 'summarizes': 1,
 'objects': 1,
 'specific': 1,
 'Summarization': 1,
 'systems': 1,
 'able': 1,
 'create': 1,
 'text': 1,
 'summaries': 2,
 'machine': 1,
 'generated': 1,
 'user': 1,
 'needs': 1,
 '\n': 2,
 'example': 3,
 'problem': 2,
 'document': 4,
 'attempts': 1,
 'automatically': 3,
 'produce': 1,
 'given': 2,
 'interested': 1,
 'generating': 1,
 'single': 1,
 'source': 2,
 'use': 1,
 'multiple': 1,
 'cluster': 1,
 'articles': 3,
 'topic': 2,
```

```
'multi': 1,  
'related': 2,  
'application': 2,  
'summarizing': 1,  
'Imagine': 1,  
'system': 3,  
'pulls': 1,  
'web': 1,  
'concisely': 1,  
'represents': 1,  
'latest': 1,  
'Image': 1,  
'automatic': 1,  
'consists': 1,  
'selecting': 1,  
'representative': 2,  
'set': 2,  
'larger': 1,  
'images.[4': 1,  
'context': 1,  
'useful': 1,  
'results': 1,  
'image': 1,  
'exploration': 1,  
'Video': 1,  
'domain': 1,  
'creates': 1,  
'trailer': 1,  
'long': 1,  
'video': 1,  
'applications': 1,  
'consumer': 1,  
'personal': 1,  
'want': 2,  
'skip': 1,  
'boring': 2,  
'repetitive': 1,  
'actions': 1,  
'Similarly': 1,  
'surveillance': 1,  
'extract': 1,  
'important': 1,  
'suspicious': 1,  
'activity': 1,  
'ignoring': 1,  
'redundant': 1,  
'frames': 1,  
'captured': 1}
```

In [46]: `len(word_frequencies)`

Out[46]: 103

In [47]: `word_frequencies`

```
Out[47]: {'broadly': 1,
 'types': 1,
 'extractive': 1,
 'summarization': 11,
 'tasks': 1,
 'depending': 2,
 'program': 1,
 'focuses': 2,
 'generic': 3,
 'obtaining': 1,
 'summary': 4,
 'abstract': 2,
 'collection': 3,
 'documents': 2,
 'sets': 1,
 'images': 3,
 'videos': 3,
 'news': 4,
 'stories': 1,
 'etc': 1,
 'second': 1,
 'query': 4,
 'relevant': 2,
 'called': 2,
 'based': 1,
 'summarizes': 1,
 'objects': 1,
 'specific': 1,
 'Summarization': 1,
 'systems': 1,
 'able': 1,
 'create': 1,
 'text': 1,
 'summaries': 2,
 'machine': 1,
 'generated': 1,
 'user': 1,
 'needs': 1,
 '\n': 2,
 'example': 3,
 'problem': 2,
 'document': 4,
 'attempts': 1,
 'automatically': 3,
 'produce': 1,
 'given': 2,
 'interested': 1,
 'generating': 1,
 'single': 1,
 'source': 2,
 'use': 1,
 'multiple': 1,
 'cluster': 1,
 'articles': 3,
 'topic': 2,
```

```
'multi': 1,  
'related': 2,  
'application': 2,  
'summarizing': 1,  
'Imagine': 1,  
'system': 3,  
'pulls': 1,  
'web': 1,  
'concisely': 1,  
'represents': 1,  
'latest': 1,  
'Image': 1,  
'automatic': 1,  
'consists': 1,  
'selecting': 1,  
'representative': 2,  
'set': 2,  
'larger': 1,  
'images.[4': 1,  
'context': 1,  
'useful': 1,  
'results': 1,  
'image': 1,  
'exploration': 1,  
'Video': 1,  
'domain': 1,  
'creates': 1,  
'trailer': 1,  
'long': 1,  
'video': 1,  
'applications': 1,  
'consumer': 1,  
'personal': 1,  
'want': 2,  
'skip': 1,  
'boring': 2,  
'repetitive': 1,  
'actions': 1,  
'Similarly': 1,  
'surveillance': 1,  
'extract': 1,  
'important': 1,  
'suspicious': 1,  
'activity': 1,  
'ignoring': 1,  
'redundant': 1,  
'frames': 1,  
'captured': 1}
```

```
In [48]:  
max_frequency = max(word_frequencies.values())  
max_frequency
```

```
Out[48]: 11
```

```
In [50]: for word in word_frequencies.keys():
    word_frequencies[word] = word_frequencies[word]/max_frequency
```



```
In [51]: word_frequencies
```

```
Out[51]: {'broadly': 0.008264462809917356,
'types': 0.008264462809917356,
'extractive': 0.008264462809917356,
'summarization': 0.09090909090909091,
'tasks': 0.008264462809917356,
'depending': 0.01652892561983471,
'program': 0.008264462809917356,
'focuses': 0.01652892561983471,
'generic': 0.024793388429752063,
'obtaining': 0.008264462809917356,
'summary': 0.03305785123966942,
'abstract': 0.01652892561983471,
'collection': 0.024793388429752063,
'documents': 0.01652892561983471,
'sets': 0.008264462809917356,
'images': 0.024793388429752063,
'videos': 0.024793388429752063,
'news': 0.03305785123966942,
'stories': 0.008264462809917356,
'etc': 0.008264462809917356,
'second': 0.008264462809917356,
'query': 0.03305785123966942,
'relevant': 0.01652892561983471,
'called': 0.01652892561983471,
'based': 0.008264462809917356,
'summarizes': 0.008264462809917356,
'objects': 0.008264462809917356,
'specific': 0.008264462809917356,
'Summarization': 0.008264462809917356,
'systems': 0.008264462809917356,
'able': 0.008264462809917356,
'create': 0.008264462809917356,
'text': 0.008264462809917356,
'summaries': 0.01652892561983471,
'machine': 0.008264462809917356,
'generated': 0.008264462809917356,
'user': 0.008264462809917356,
'needs': 0.008264462809917356,
'\n': 0.01652892561983471,
'example': 0.024793388429752063,
'problem': 0.01652892561983471,
'document': 0.03305785123966942,
'attempts': 0.008264462809917356,
'automaticaly': 0.024793388429752063,
'produce': 0.008264462809917356,
'given': 0.01652892561983471,
'interested': 0.008264462809917356,
'generating': 0.008264462809917356,
'single': 0.008264462809917356,
'source': 0.01652892561983471,
'use': 0.008264462809917356,
'multiple': 0.008264462809917356,
'cluster': 0.008264462809917356,
'articles': 0.024793388429752063,
'topic': 0.01652892561983471,
```

```
'multi': 0.008264462809917356,
'related': 0.01652892561983471,
'application': 0.01652892561983471,
'summarizing': 0.008264462809917356,
'Imagine': 0.008264462809917356,
'system': 0.024793388429752063,
'pulls': 0.008264462809917356,
'web': 0.008264462809917356,
'concisely': 0.008264462809917356,
'represents': 0.008264462809917356,
'latest': 0.008264462809917356,
'Image': 0.008264462809917356,
'automatic': 0.008264462809917356,
'consists': 0.008264462809917356,
'selecting': 0.008264462809917356,
'representative': 0.01652892561983471,
'set': 0.01652892561983471,
'larger': 0.008264462809917356,
'images.[4': 0.008264462809917356,
'context': 0.008264462809917356,
'useful': 0.008264462809917356,
'results': 0.008264462809917356,
'image': 0.008264462809917356,
'exploration': 0.008264462809917356,
'Video': 0.008264462809917356,
'domain': 0.008264462809917356,
'creates': 0.008264462809917356,
'trailer': 0.008264462809917356,
'long': 0.008264462809917356,
'video': 0.008264462809917356,
'applications': 0.008264462809917356,
'consumer': 0.008264462809917356,
'personal': 0.008264462809917356,
'want': 0.01652892561983471,
'skip': 0.008264462809917356,
'boring': 0.01652892561983471,
'repetitive': 0.008264462809917356,
'actions': 0.008264462809917356,
'Similarly': 0.008264462809917356,
'surveilliance': 0.008264462809917356,
'extract': 0.008264462809917356,
'important': 0.008264462809917356,
'suspicious': 0.008264462809917356,
'activity': 0.008264462809917356,
'ignoring': 0.008264462809917356,
'redundant': 0.008264462809917356,
'frames': 0.008264462809917356,
'captured': 0.008264462809917356}
```

In [52]: `sentence_tokens = [sent for sent in doc.sents]`

In [53]: `sentence_tokens`

Out[53]: [There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on.,
The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.),,
The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query.,
Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.,
An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document.,
Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic).,
This problem is called multi-document summarization.,
A related application is summarizing news articles.,
Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.,
Image collection summarization is another application example of automatic summarization.,
It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system.,
Video summarization is a related domain, where the system automatically creates a trailer of a long video.,
This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions.,
Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured]

In [54]: `len(sentence_tokens)`

Out[54]: 14

In [55]: `sentence_tokens`

Out[55]: [There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on.,
The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.),,
The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query.,
Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.,
An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document.,
Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic).,
This problem is called multi-document summarization.,
A related application is summarizing news articles.,
Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.,
Image collection summarization is another application example of automatic summarization.,
It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system.,
Video summarization is a related domain, where the system automatically creates a trailer of a long video.,
This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions.,
Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured]

In [58]:

```
sentence_score = {}

for sent in sentence_tokens:
    for word in sent:
        if word.text.lower() in word_frequencies.keys():
            if sent not in sentence_score.keys():
                sentence_score[sent] =
word_frequencies[word.text.lower()]
            else:
                sentence_score[sent] +=
word_frequencies[word.text.lower()]
```

In [59]:

```
sentence_score
```

```
Out[59]: {There are broadly two types of extractive summarization tasks depending on what  
the summarization program focuses on.: 0.25619834710743805,  
The first is generic summarization, which focuses on obtaining a generic summar  
y or abstract of the collection (whether documents, or sets of images, or videos  
, news stories etc.): 0.36363636363636354,  
The second is query relevant summarization, sometimes called query-based summar  
ization, which summarizes objects specific to a query.: 0.3553719008264462,  
Summarization systems are able to create both query relevant text summaries and  
generic machine-generated summaries depending on what the user needs.: 0.2975206  
6115702475,  
An example of a summarization problem is document summarization, which attempts  
to automatically produce an abstract from a given document.: 0.3636363636363636  
5,  
Sometimes one might be interested in generating a summary from a single source  
document, while others can use multiple source documents (for example, a cluster  
of articles on the same topic).: 0.23140495867768596,  
This problem is called multi-document summarization.: 0.1652892561983471,  
A related application is summarizing news articles.: 0.09917355371900827,  
Imagine a system, which automatically pulls together news articles on a given t  
opic (from the web), and concisely represents the latest news as a summary.: 0.2  
644628099173554,  
Image collection summarization is another application example of automatic summ  
arization.: 0.2644628099173554,  
It consists in selecting a representative set of images from a larger set of im  
ages.[4] A summary in this context is useful to show the most representative ima  
ges of results in an image collection exploration system.: 0.2727272727272727,  
Video summarization is a related domain, where the system automatically creates  
a trailer of a long video.: 0.2066115702479339,  
This also has applications in consumer or personal videos, where one might want  
to skip the boring or repetitive actions.: 0.10743801652892562,  
Similarly, in surveillance videos, one would want to extract important and susp  
icious activity, while ignoring all the boring and redundant frames captured:  
0.1322314049586777}
```

```
In [60]: from heapq import nlargest
```

```
In [61]: select_length = int(len(sentence_tokens)*0.4)
```

```
In [62]: select_length
```

```
Out[62]: 5
```

```
In [65]: summary = nlargest(select_length,sentence_score,key =  
sentence_score.get)
```

```
In [66]: summary
```

Out[66]: [An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document.,
The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos , news stories etc.),,
The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query.,
Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.,
It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system.]

In [67]: sentence_score

Out[67]: {There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on.: 0.25619834710743805,
The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos , news stories etc.): 0.36363636363636354,
The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query.: 0.3553719008264462,
Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.: 0.2975206615702475,
An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document.: 0.36363636363636365,
Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic).: 0.23140495867768596,
This problem is called multi-document summarization.: 0.1652892561983471,
A related application is summarizing news articles.: 0.09917355371900827,
Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.: 0.2644628099173554,
Image collection summarization is another application example of automatic summarization.: 0.2644628099173554,
It consists in selecting a representative set of images from a larger set of images.[4] A summary in this context is useful to show the most representative images of results in an image collection exploration system.: 0.2727272727272727,
Video summarization is a related domain, where the system automatically creates a trailer of a long video.: 0.2066115702479339,
This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions.: 0.10743801652892562,
Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured: 0.1322314049586777}

In []: final_summary = [word.text for word in words]