Data cleaning

```
import string
sentence='This is a  book. It is a      weg.'
sentence=sentence.split(' ')
sentence=[word.strip(string.punctuation+string.whitespace) for word in sentence]
sentence=[word for word in sentence if len(word)>1 or (word.lower()=='a' or word.lower=='i')]
print(sentence)
```

```
     ['This', 'is', 'a', 'book', 'It', 'is', 'a', 'weg']
```

More cleaning

```
import re
sentence='This is a book? It is\n\n\ndfsjfjdkjk q[123]q \a\b'
content=re.sub('\n|[[\d+\]]',' ',sentence)
print(content)
content=bytes(content,'UTF-8')
print(content)
content=content.decode('ascii','ignore')
print(content)
```

```
     This is a book? It is   dfsjfjdkjk q     q
     b'This is a book? It is   dfsjfjdkjk q     q \x07\x08'
     This is a book? It is   dfsjfjdkjk q     q
```

```
print(string.punctuation)
```

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

Extracting sentences

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
import string
def cleanSentence(sentence):
  sentence=sentence.split(' ')
  sentence=[word.strip(string.punctuation+string.whitespace) for word in sentence]
  sentence=[word for word in sentence if len(word)>1 or (word.lower()=='a' or word.lower=='i')]
  return sentence
def cleanInput(content):
  content=re.sub('\n|[[\d+\]]',' ', content)
  content=bytes(content,'UTF-8')
  content=content.decode('ascii','ignore')
  sentences=content.split('. ')
  return [cleanSentence(sentence) for sentence in sentences]
def getNgramsFromSentences(content,n):
  output=[]
  for i in range(len(content)-n+1):
    output.append(content[i:i+n])
  return output
def getNgrams(content,n):
  content=cleanInput(content)
  ngrams=[]
  for sentence in content:
    ngrams.extend(getNgramsFromSentences(sentence,n))
  return ngrams
```

Generating ngrams from the above code

```
html=urlopen('https://en.wikipedia.org/wiki/Python_(programming_language)')
bs=BeautifulSoup(html,'html.parser')
```

```
content=bs.find('div',{'id':'mw-content-text'}).get_text()
ngrams=getNgrams(content,2)
print(ngrams)
print('2-gram count is ',len(ngrams))
```

```
[['General-purpose', 'programming'], ['programming', 'language'], ['language', 'mw-parser-output'], ['mw-parser-output', 'infob
2-gram count is  9591
```

Finding the number of occurances of n-grams

```
from collections import Counter
def getNgrams(content,n):
  content=cleanInput(content)
  ngrams=Counter()
  for sentence in content:
    newNgrams=[' '.join(ngram) for ngram in getNgramsFromSentences(sentence,2)]
    ngrams.update(newNgrams)
  return ngrams
print(getNgrams(content,2))
```

```
Counter({'from the': 218, 'the original': 209, 'original on': 207, 'Archived from': 200, 'on June': 60, 'Software Foundation':
```