온라인상에서 바로 데이터 수집해서 실습

https://www.forbes.com/sites/adrianbridgwater/2019/04/15/what-drove-the-ai-renaissance/#5ba50c691f25

데이터수집

```
[ ]  1 import requests
     2 from bs4 import BeautifulSoup
     3 url = 'https://www.forbes.com/sites/adrianbridgwater/2019/04/15/what-drove-the-ai-renaissance/?ss=ai-big-data#45dd5dd61f25'
     4 response = requests.get(url)
     5 soup = BeautifulSoup(response.text,'html.parser')
```

```
     1 div = soup.find('div',class_='article-body fs-article fs-responsive-text current-article')
     2 p_tag = div.find_all('p')
     3 content=''
     4 for i in p_tag:
     5     content+=i.text
     6 content
```

```
'Italian Renaissance: Vitruvian Man by Leonardo da VinciIt is the present-day darling of the tech world. The current renaissance of Artificial Intelligence (AI) with
its sister discipline Machine Learning (ML) has led every IT firm worth its salt to engineer some form of AI onto its platform, into its toolsets and throughout its s
oftware applications.IBM CEO Ginni Rometty has already proclaimed that AI will change 100 percent of jobs over the next decade.And yes, she does mean everybody\'s job
from yours to mine and onward to the role of grain farmers in Egypt, pastry chefs in Paris and dog walkers in Oregon i.e. every job. We will now be able to help direc
t all workers' actions and behavior with a new degree of intelligence that comes from predictive analytics, all stemming from the AI engines we will now increasingly
depend upon.When did it all go so right?But AI used to be a fanciful notion mostly confined science fiction, so when did it all go right?In recent years we've had som
e bi…'
```

영문토큰화

punkt tokenizer 참고 : https://www.nltk.org/_modules/nltk/tokenize/punkt.html

word_tokenize() : 마침표와 구두점(온점(.), 컴마(,), 물음표(?), 세미콜론(;), 느낌표(!) 등과 같은 기호)으로 구분하여 토큰화

```
[ ]  1 # word_tokenize
     2 import nltk
     3 # nltk punkt tokenizer download
     4 nltk.download('punkt')
     5 from nltk.tokenize import word_tokenize
     6 token1 = word_tokenize(content)
     7 print(token1)
```

```
['Italian', 'Renaissance', ':', 'Vitruvian', 'Man', 'by', 'Leonardo', 'da', 'VinciIt', 'is', 'the', 'prese                    ', 'tech', 'world', '.', 'The',
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

WordPunctTokenizer() : 알파벳이 아닌문자를 구분하여 토큰화

```
[ ]  1 # WordPunctTokenizer() : 알파벳이 아닌문자를 구분하여 토큰화
     2 import nltk
     3 from nltk.tokenize import WordPunctTokenizer
     4 token2 = WordPunctTokenizer().tokenize(content)
     5 print(token2)
```

```
['Italian', 'Renaissance', ':', 'Vitruvian', 'Man', 'by', 'Leonardo', 'da', 'VinciIt', 'is', 'the', 'present', '-', 'day', 'darling', 'of', 'the', 'tech', 'world', '.
```

TreebankWordTokenizer() : 정규표현식에 기반한 토큰화

```
[ ]  1 # TreebankWordTokenizer() : 정규표현식에 기반한 토큰화
     2 import nltk
     3 from nltk.tokenize import TreebankWordTokenizer
     4 token = TreebankWordTokenizer().tokenize(content)
     5 print(token[:20])
```

```
['Italian', 'Renaissance', ':', 'Vitruvian', 'Man', 'by', 'Leonardo', 'da', 'VinciIt', 'is', 'the', 'present-day', 'darling', 'of', 'the', 'tech', 'world.', 'The', 'cu
```

영문 품사부착

분리한 토큰마다 품사를 부착한다

https://www.nltk.org/api/nltk.tag.html

태크목록 : https://pythonprogramming.net/natural-language-toolkit-nltk-part-speech-tagging/

```
[ ]  1 from nltk import pos_tag
     2 nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
True
```

```
[ ]  1 taggedToken = pos_tag(token1)
     2 print(taggedToken[:20])
```

```
[('Italian', 'JJ'), ('Renaissance', 'NNP'), (':', ':'), ('Vitruvian', 'JJ'), ('Man', 'NN'), ('by', 'IN'), ('Leonardo', 'NNP'), ('da', 'NN'), ('VinciIt', 'NNP'), ('is',
```

영문 개체명인식

http://www.nltk.org/api/nltk.chunk.html

```
[ ]  1 # 예시 : Barack Obama likes fried chicken very much
     2 # word_tokenize() : 마침표와 구두점(온점(.), 컴마(,), 물음표(?), 세미콜론(;), 느낌표(!) 등과 같은 기호)으로 구분하여 토큰화
     3 nltk.download('words')
     4 nltk.download('maxent_ne_chunker')
```

```
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Package words is already up-to-date!
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package maxent_ne_chunker is already up-to-date!
True
```

```
[ ]  1 import nltk
     2 nltk.download('punkt')
     3 from nltk.tokenize import word_tokenize
     4 # 토큰화
     5 token1 = word_tokenize('Barack Obama likes fried chicken very much')
     6 print('token:',token1)
```

```
token: ['Barack', 'Obama', 'likes', 'fried', 'chicken', 'very', 'much']
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
[ ]  1 # pos-tag
     2 taggedToken = pos_tag(token1)
     3 print('pos-tag:',taggedToken)
```

```
pos-tag: [('Barack', 'NNP'), ('Obama', 'NNP'), ('likes', 'VBZ'), ('fried', 'VBN'), ('chicken', 'JJ'), ('very', 'RB'), ('much', 'JJ')]
```

```
[ ]  1 # chunking
     2 from nltk import ne_chunk
     3 neToken = ne_chunk(taggedToken)
     4 print(neToken)
```

```
(S
  (PERSON Barack/NNP)
  (ORGANIZATION Obama/NNP)
  likes/VBZ
  fried/VBN
  chicken/JJ
  very/RB
  much/JJ)
```