Search engine for COVID-19

IN104

Jessica López Espejel

ENSTA, Institut Polytechnique Paris

Problem and approach

Due to the pandemic situation, thousands of institutions and scientists are mobilized to tackle the Coronavirus (Covid-19). In consequence, the number of medical articles has been increasing exponentially.

Image taken from internet

Problem and approach

- The number of health articles increased by 92% in 2020 (nature, 2020)

- In 2020, the number of submissions to Elsevier's journals increased by 58% between February and May compared to the same period in 2019 (Squazzoni et al., 2020)

Problem and approach

Natural Language Processing is an important field in computer science that comprises many techniques to help us process information faster. In our case study, we do it through a search engine focused on articles related to Covid-19.

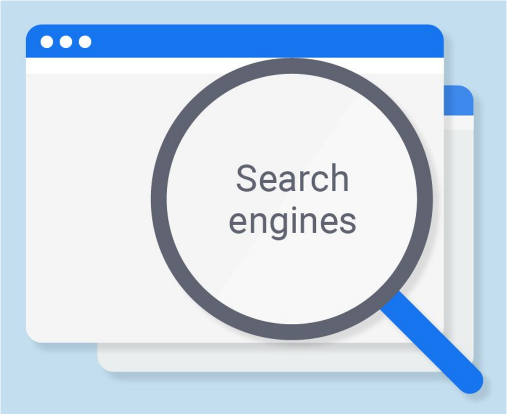


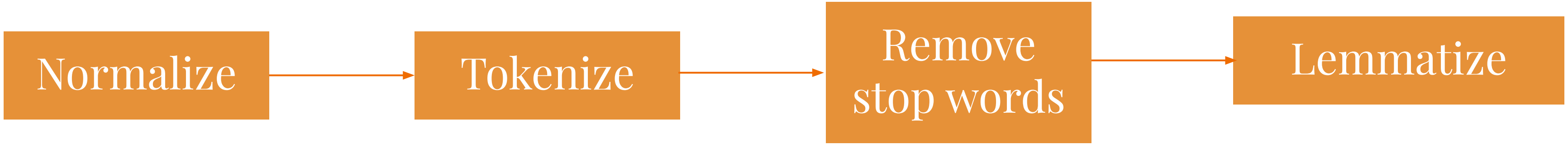
Image taken from internet

Tools

- **NLTK (Natural Language Toolkit)** is a python library focused on Natural Language Processing. We can use it to develop various tasks such as classification, tokenization, stemming, tagging, parsing, etc.

- **Whoosh** is a search engine library developed in Python.

Workflow



Data processing

- Load data
- Clean text (for example: remove the tags)
- **Normalize** (lowercase, remove punctuation, remove extra spaces)
- **Remove stop words**
- **Lemmatize**
- Remove articles that are not written in English
- Write the content and title of the articles in a different folder (details in the next slide)

Lower case

- It is the first step to pre-process the data
- We can use simply: `string.lower()`

Example:

```
>> text = “Clinical features of culture-proven Mycoplasma pneumoniae
infections at King Abdulaziz University Hospital, Jeddah, Saudi Arabia”
```

```
>> text.lower()
```

```
clinical features of culture-proven mycoplasma pneumoniae infections at king
abdulaziz university hospital, jeddah, saudi arabia
```

Punctuation

```
import string

punctuation = string.punctuation

print(punctuation, type(punctuation))
```

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

Stop words removal

- Stop words are common words that we find in the texts, such as, “a”, “the”, “is”

Sample text with stop words	Same text without stop words
facebook **and** google failed **to** remove online scam adverts after fraud victims reported **them**, according **to** consumer watchdog **which**?	facebook google failed remove online scam adverts fraud victims reported , according consumer watchdog ?
on facebook, **the** biggest reason people **did not** report **the** scam **was they** doubted anything would **be** done.	facebook , biggest reason people report scam doubted anything would done .

Stop words removal

```
from nltk.corpus import stopwords

stop_words = set(stopwords.words('english'))

print(stop_words, type(stop_words))
```

```
{"you'll", 'shan', 'did', 'weren', 'them', 'all', "hasn't", 'does', "you're", "needn't", 'wasn', 'me', 'doing', "won't", 'during', 'through', 'too', 'will',
"aren't", "don't", 'mightn', 'whom', 'into', "should've", "wasn't", 'what', 'him', 'other', "shouldn't", 'himself', 'who', 'have', 'o', "weren't", 'only', 'or',
'about', 'below', 'once', 'his', 'until', 'not', 'is', 'it', 'being', 'haven', 'ain', 'out', 'yourselves', 'because', 'yourself', 'been', "mustn't", 'if', 'by',
'had', 'won', "didn't", 'her', "mightn't", 'over', 'more', 'same', 've', 'up', 'as', 'off', 'wouldn', 'when', 'both', 'that', 'than', 'can', "couldn't", 'why',
'from', "that'll", 'itself', 'here', 'your', 'should', 'with', 'against', 'under', 'between', 'isn', 'and', 'then', 'so', 'be', 'no', 'before', 'yours',
'herself', 'where', 'hers', 'were', 'my', 'most', "isn't", "she's", 'needn', "you've", "doesn't", 'any', 'ourselves', 'mustn', 'but', 'few', 'we', 'this',
"haven't", 'themselves', 'he', 'they', 'some', 'y', 'she', 'couldn', 'there', 'each', 'at', 'now', 's', 'myself', 'those', 'shouldn', 'are', 'doesn', 'just',
"hadn't", 'own', 'such', 'theirs', 'of', 'after', 'don', 'further', 'very', "it's", 'do', "you'd", 'the', 'on', 'i', 'hasn', 'these', "wouldn't", 'aren', 'am',
'm', "shan't", 'to', 'above', 'll', 'was', 'its', 'again', 'ours', 'hadn', 'down', 'our', 'which', 'while', 't', 'didn', 'their', 'nor', 'how', 'has', 'you', 'd',
'in', 're', 'a', 'for', 'an', 'ma', 'having'}  <class 'set'>
```

Tokenization

It is the process of getting tokens from the source text.

What is a token?

A token is the most valuable piece of information

Tokenization

```
from nltk import word_tokenize, sent_tokenize
```

```
text = “on facebook, the biggest reason people did not report the scam was
they doubted anything would be done.”
```

```
lst_words = [word for word in word_tokenize(text)]
```

```
['on', 'facebook', ',', 'the', 'biggest', 'reason', 'people', 'did', 'not',
'report', 'the', 'scam', 'was', 'they', 'doubted', 'anything', 'would',
'be', 'done', '.']
```

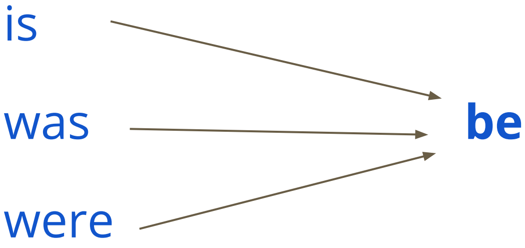
```
text = ‘ ’.join(lst_words)
```

```
'on facebook , the biggest reason people did not report the scam was they
doubted anything would be done .'
```

Lemmatization

Reduce words to normalize the text. Here, the transformation uses a dictionary to replace various words with a unique representative one. For instance, we can replace multiple verb forms by their infinitive.

Example:



is

was

were

be

Lemmatization

```
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print("methods :", lemmatizer.lemmatize("methods"))

print("worst :", lemmatizer.lemmatize("worst"))
```

Results

```
methods: method

worst: bad
```

Activity

- **utils.py**
  - Complete the method `write_file`
- **extract_data.py**
  - Complete the method `get_text`
- **preprocess_data.py**
  - Complete the methods `remove_punctuation, remove_number, remove_special_character`
- **`main.py`**
  - Call your methods as the script indicates

Installation example

```
pip install nltk

pip install whoosh

>> python

>> import nltk

>> nltk.download('wordnet')
```

Recommended bibliography

- https://www.nltk.org/
- https://www.guru99.com/tokenize-words-sentences-nltk.html