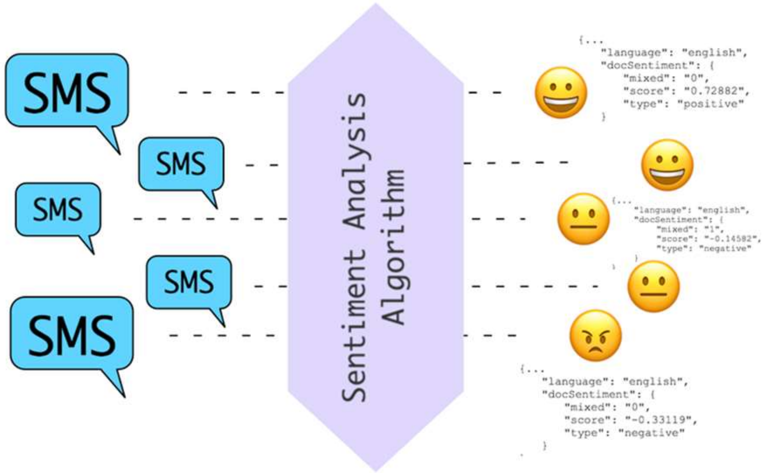
Text Data

ECE30007 Intro to AI Project

한동대학교
HANDONG GLOBAL UNIVERSITY

What is NLP?

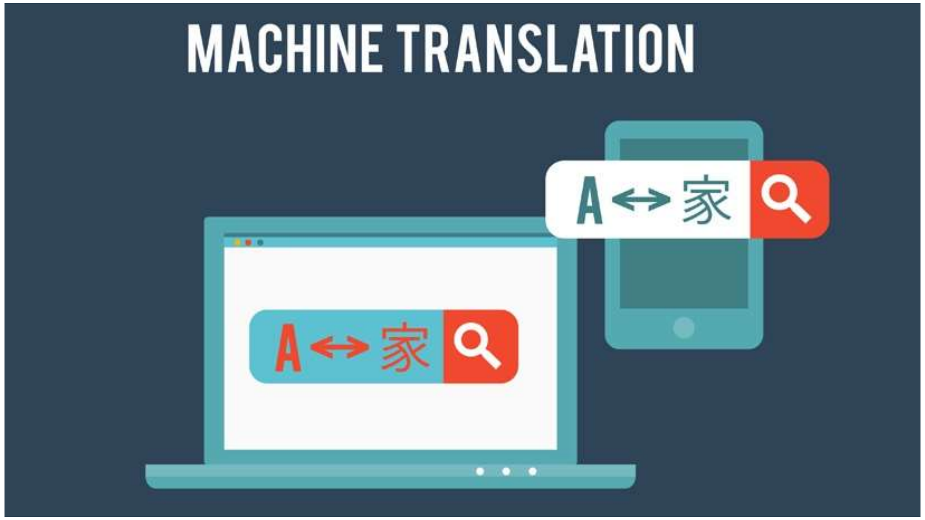


```
{...
    "language": "english",
    "docSentiment": {
        "mixed": "0",
        "score": "0.72882",
        "type": "positive"
    }
```

```
{...
    "language": "english",
    "docSentiment": {
        "mixed": "1",
        "score": "-0.14582",
        "type": "negative"
    }
}
```

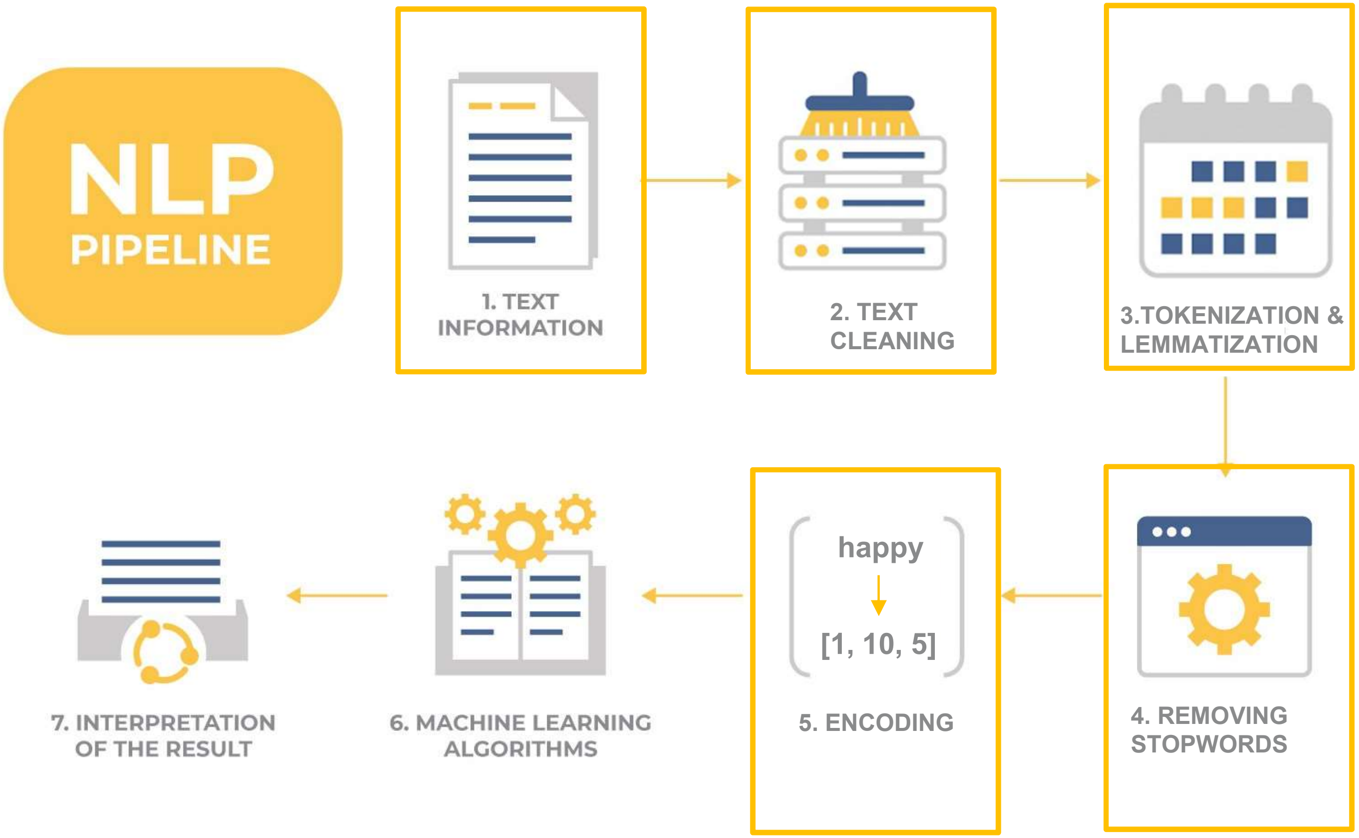
```
{...
    "language": "english",
    "docSentiment": {
        "mixed": "0",
        "score": "-0.33119",
        "type": "negative"
    }
```

https://www.kaggle.com/aadilsrivastava01/a-beginners-guide-to-sentiment-analysis



https://elearningindustry.com/4-machine-translation-tools-incorporating-machine-translation

NLP pipeline



I'm studying COMPUTER SCINECE...!!

I'm studying computer science

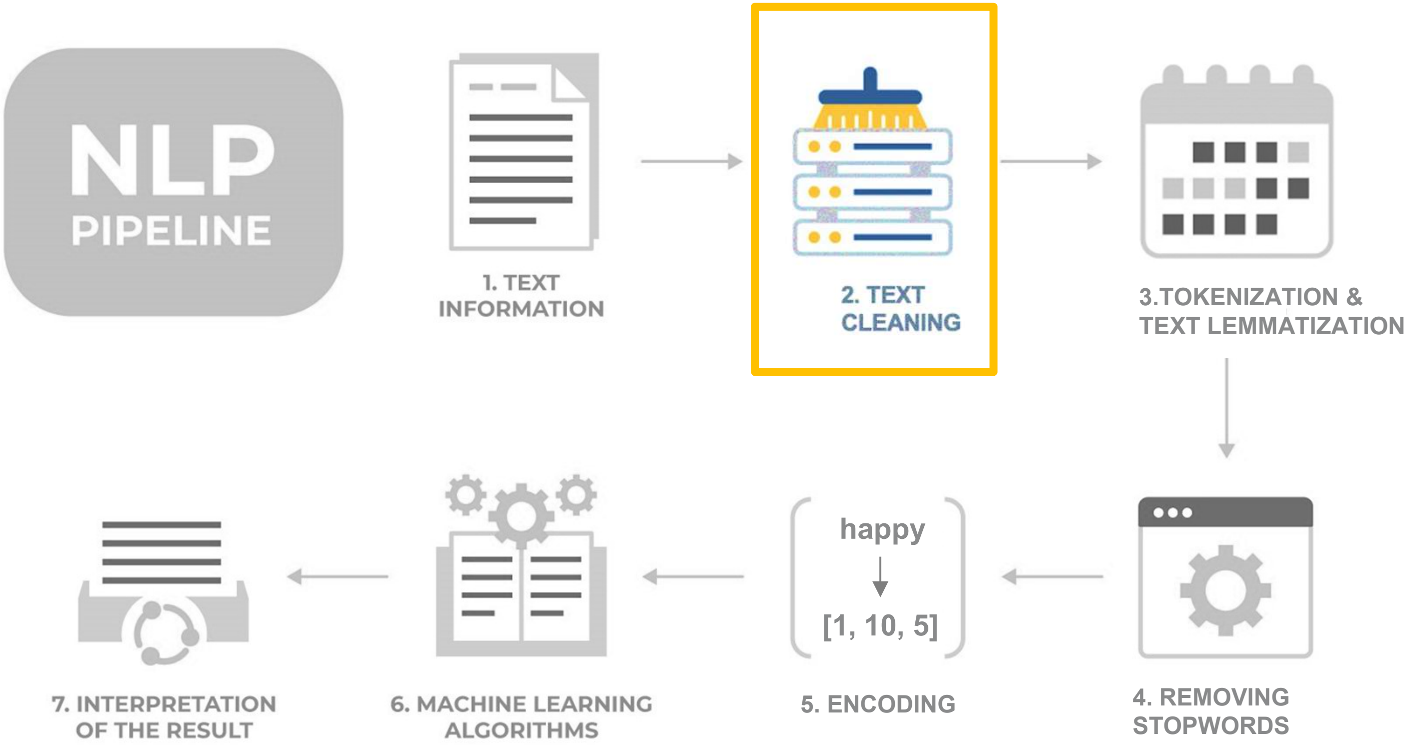
I | ‘m | studying | computer | science

I | be | study | computer | science

study | computer | science

study	10	1.2, 0.1
computer	20	0.3, 2.1
science	30	-0.2, 1.2

NLP preprocessing

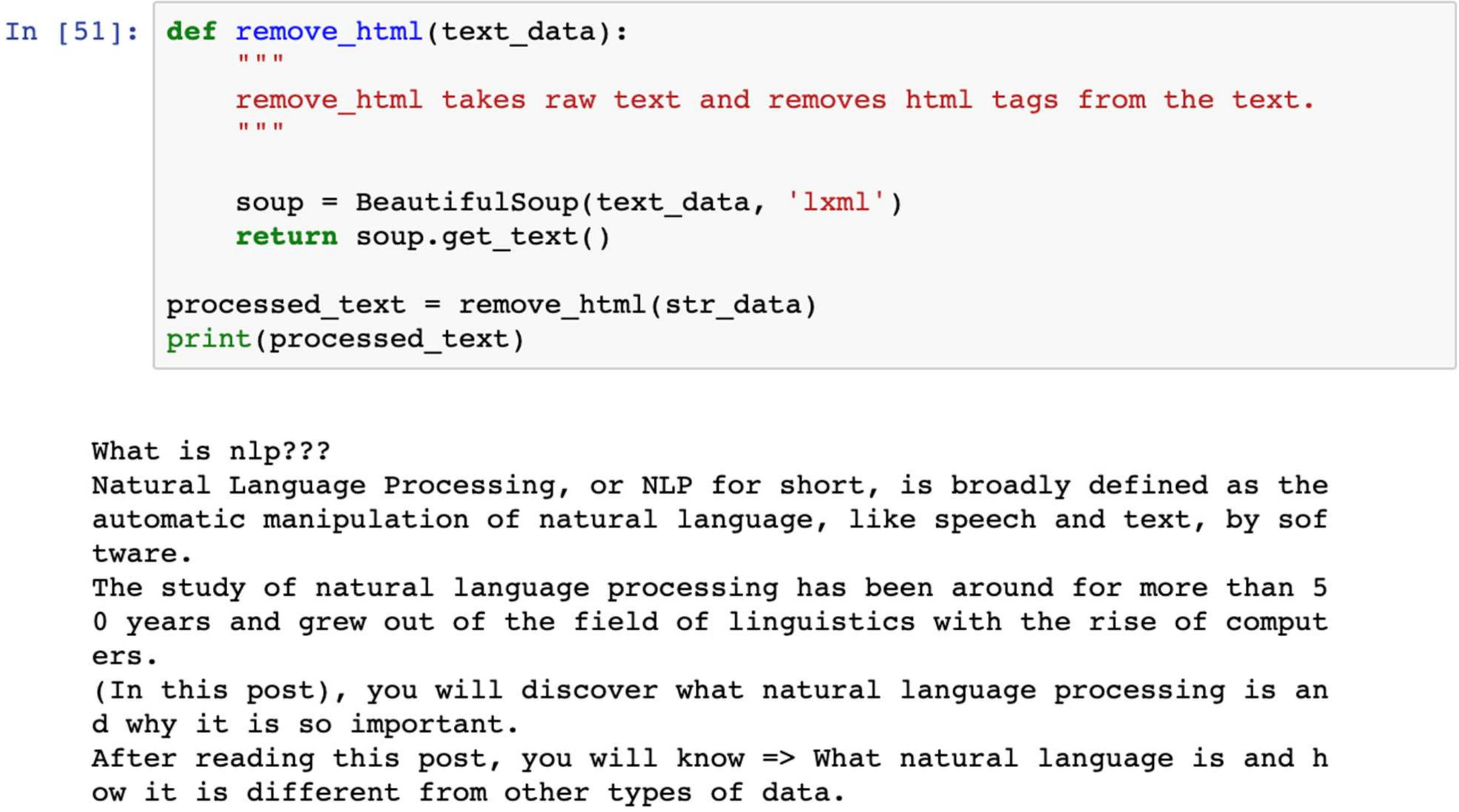

NLP
PIPELINE
1. TEXT INFORMATION
2. TEXT CLEANING
3.TOKENIZATION & TEXT LEMMATIZATION
4. REMOVING STOPWORDS
happy
[1, 10, 5]
5. ENCODING
6. MACHINE LEARNING ALGORITHMS
7. INTERPRETATION OF THE RESULT

1. Text data

'<html><h2>What is nlp??? </h2></html> \nNatural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.\nThe study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers.\n(In this post), you will discover what natural language processing is and why it is so important.\nAfter reading this post, you will know => What natural language is and how it is different from other types of data.'

2. Cleaning

- Removing HTML Tag

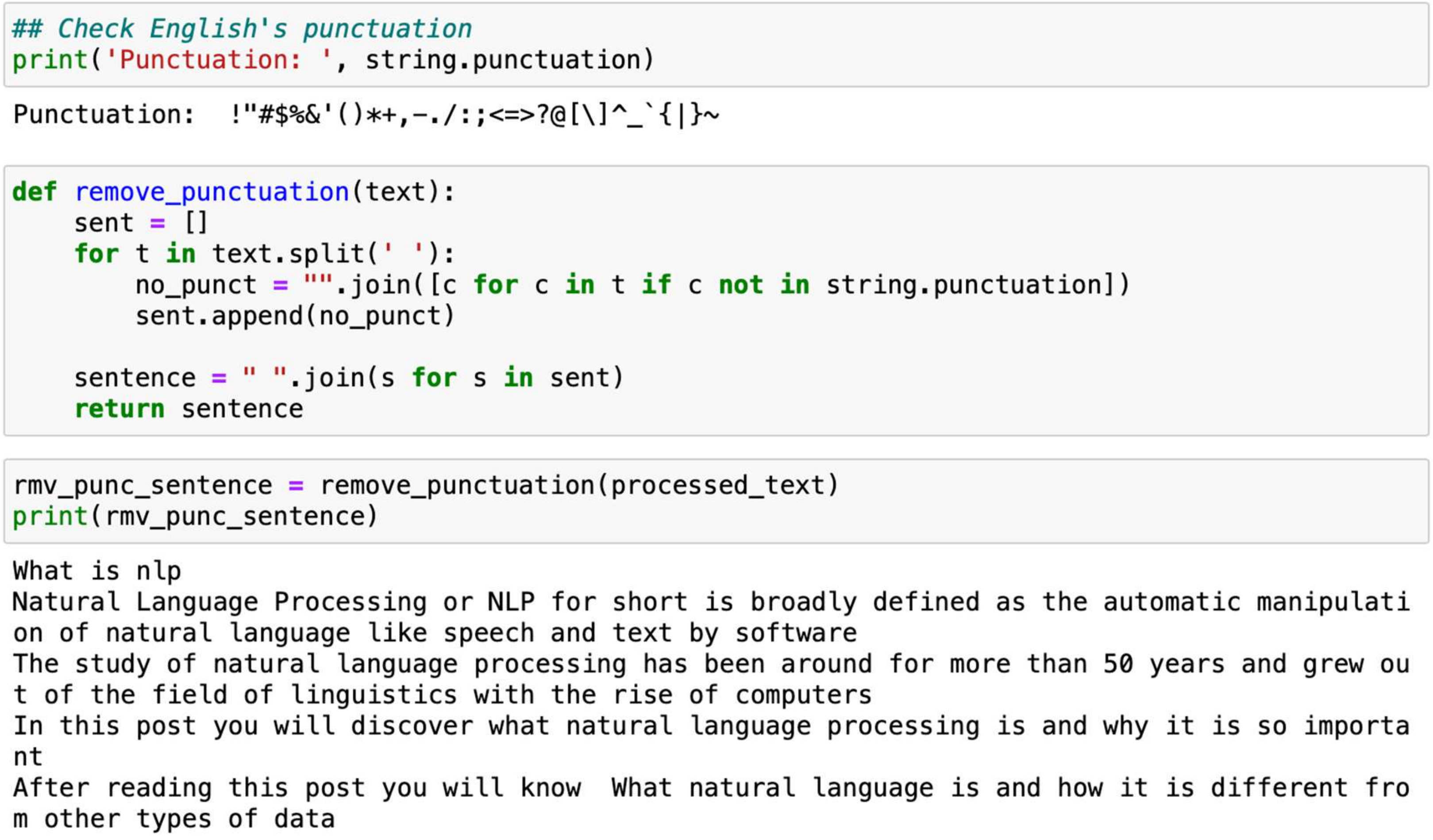


```
In [51]: def remove_html(text_data):
             """
             remove_html takes raw text and removes html tags from the text.
             """

             soup = BeautifulSoup(text_data, 'lxml')
             return soup.get_text()

         processed_text = remove_html(str_data)
         print(processed_text)
```

```
What is nlp???
Natural Language Processing, or NLP for short, is broadly defined as the
automatic manipulation of natural language, like speech and text, by sof
tware.
The study of natural language processing has been around for more than 5
0 years and grew out of the field of linguistics with the rise of comput
ers.
(In this post), you will discover what natural language processing is an
d why it is so important.
After reading this post, you will know => What natural language is and h
ow it is different from other types of data.
```

2. Cleaning

- Removing punctuation



```
## Check English's punctuation
print('Punctuation: ', string.punctuation)
```

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

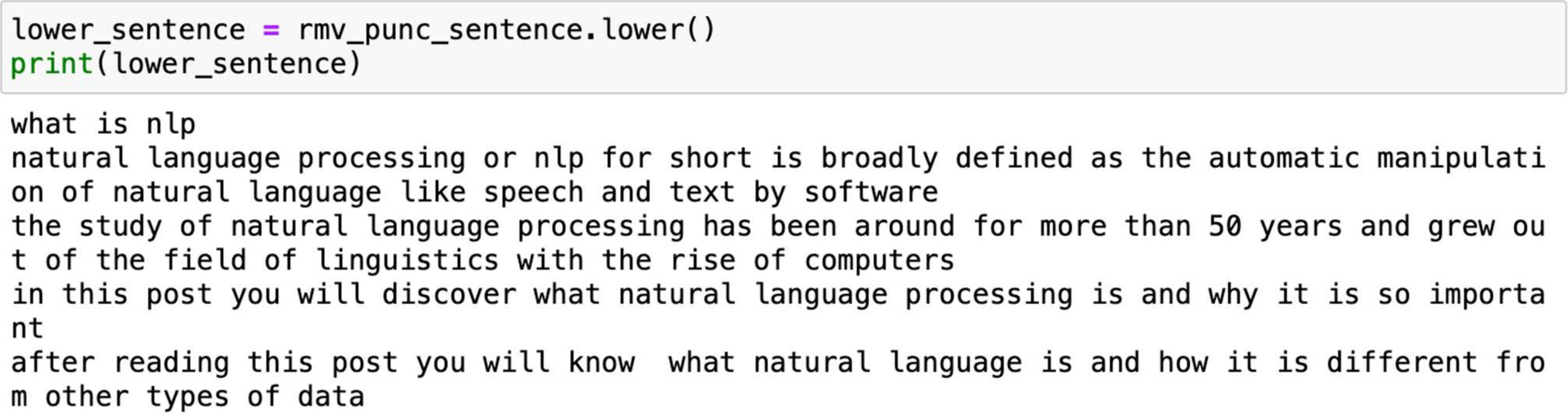
```
def remove_punctuation(text):
    sent = []
    for t in text.split(' '):
        no_punct = "".join([c for c in t if c not in string.punctuation])
        sent.append(no_punct)

    sentence = " ".join(s for s in sent)
    return sentence
```

```
rmv_punc_sentence = remove_punctuation(processed_text)
print(rmv_punc_sentence)
```

```
What is nlp
Natural Language Processing or NLP for short is broadly defined as the automatic manipulati
on of natural language like speech and text by software
The study of natural language processing has been around for more than 50 years and grew ou
t of the field of linguistics with the rise of computers
In this post you will discover what natural language processing is and why it is so importa
nt
After reading this post you will know  What natural language is and how it is different fro
m other types of data
```

2. Cleaning

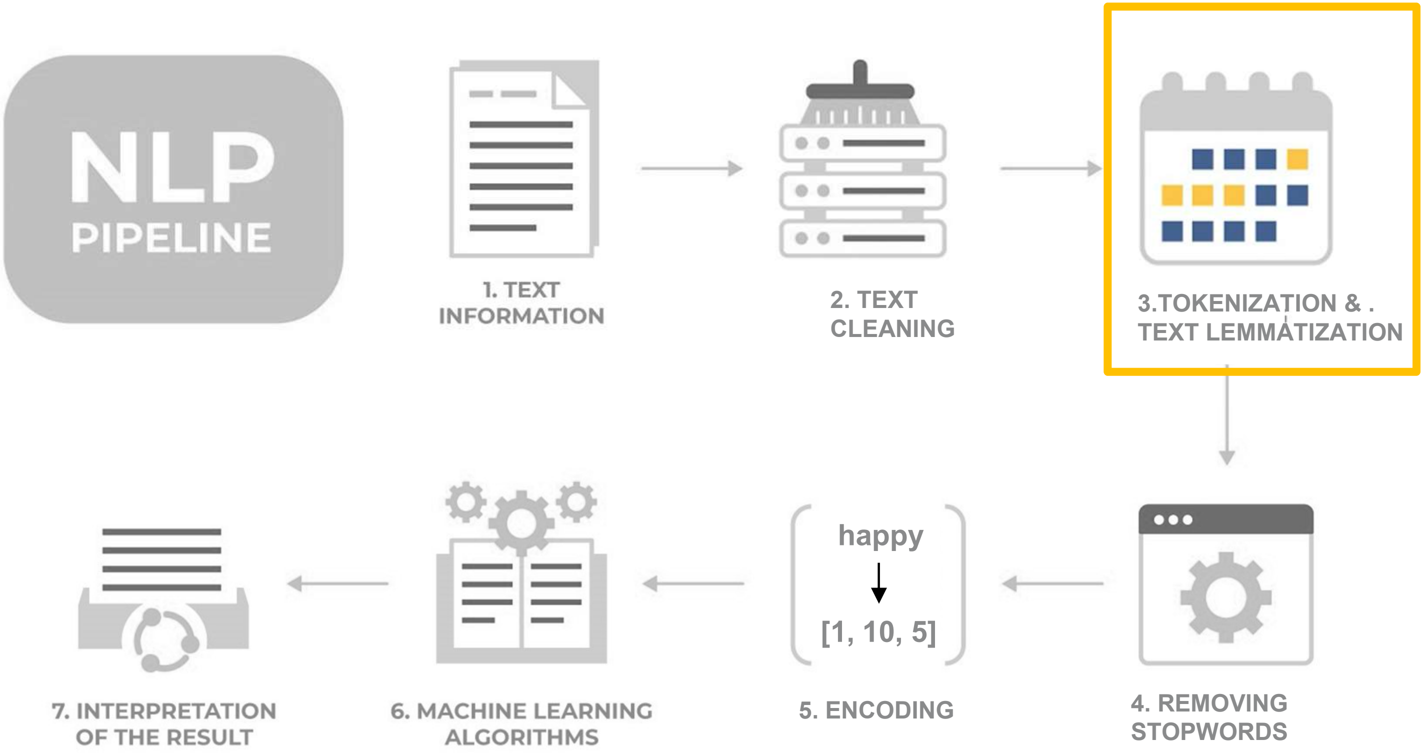
- Integration of upper and lower case letter



```
lower_sentence = rmv_punc_sentence.lower()
print(lower_sentence)
```

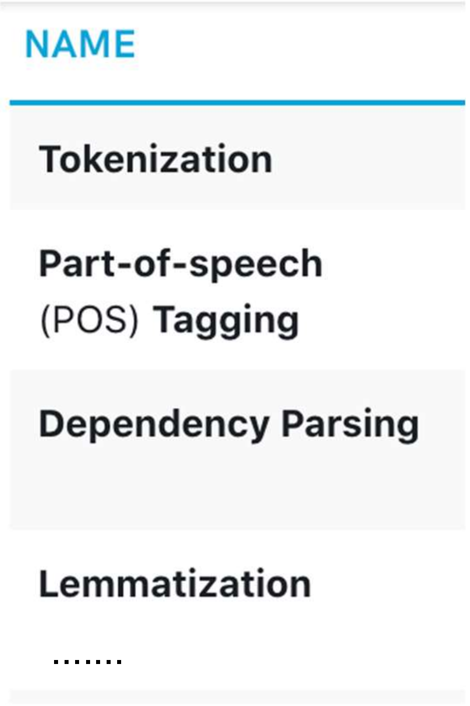
```
what is nlp
natural language processing or nlp for short is broadly defined as the automatic manipulati
on of natural language like speech and text by software
the study of natural language processing has been around for more than 50 years and grew ou
t of the field of linguistics with the rise of computers
in this post you will discover what natural language processing is and why it is so importa
nt
after reading this post you will know  what natural language is and how it is different fro
m other types of data
```

NLP preprocessing



NLP preprocessing

- What is spacy
  - a **free, open-source library** for advanced **Natural Language Processing** (NLP) in Python.

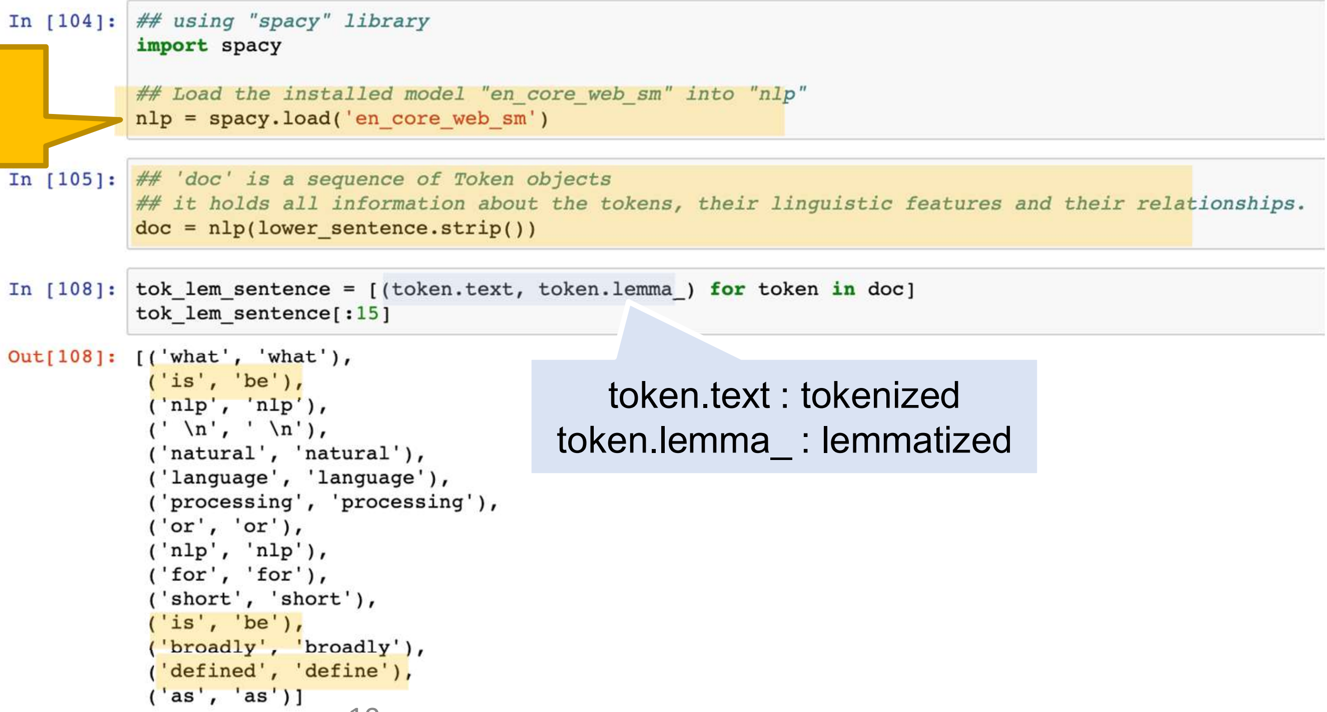


NAME
Tokenization
Part-of-speech (POS) **Tagging**
Dependency Parsing
Lemmatization
…….

SPACY

3. Tokenization & Lemmatization

- **Tokenization(Sentence & Word)**
  - The process of tokenizing or splitting a string, text into a list of tokens
- **Lemmatization**
  - Transforming the word into a proper root form.
  - Considering the context and converting the word to its meaningful base form

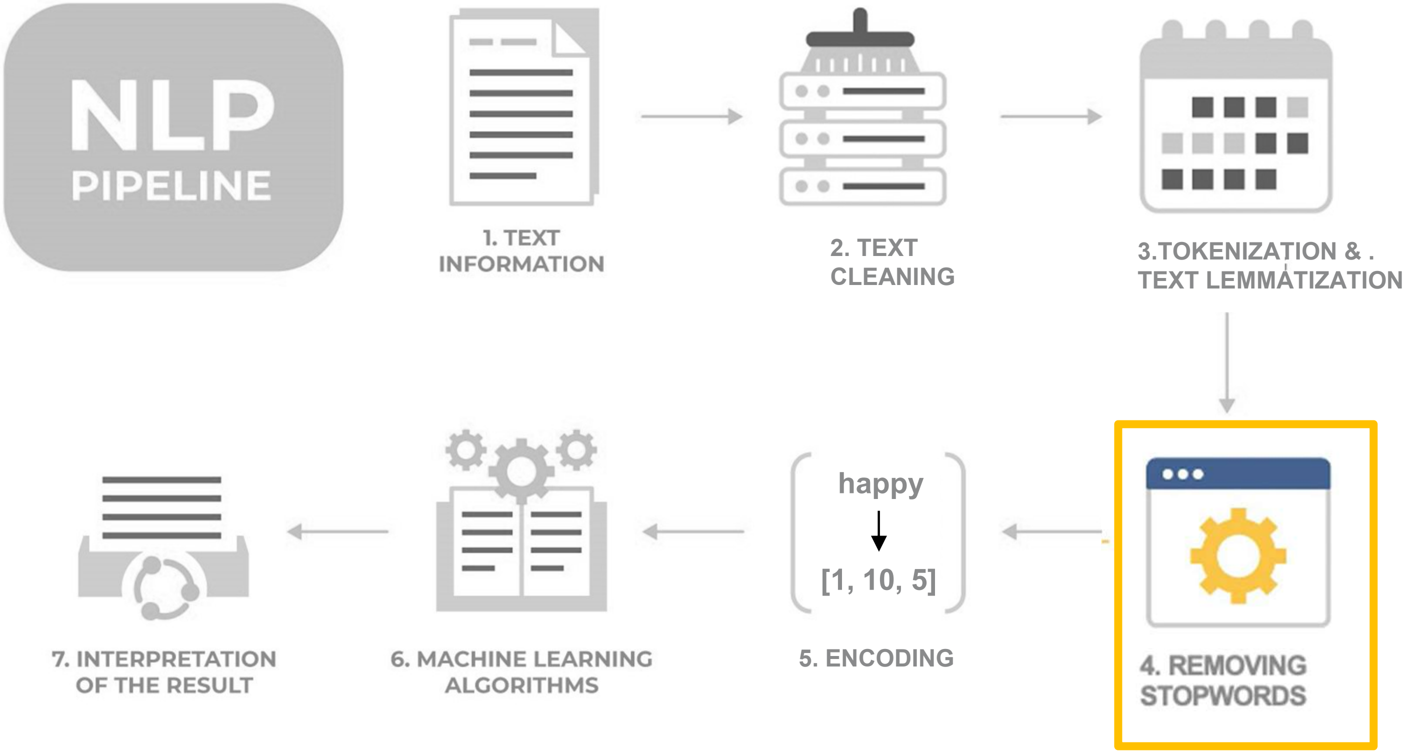


```
In [104]: ## using "spacy" library
          import spacy

          ## Load the installed model "en_core_web_sm" into "nlp"
          nlp = spacy.load('en_core_web_sm')

In [105]: ## 'doc' is a sequence of Token objects
          ## it holds all information about the tokens, their linguistic features and their relationships.
          doc = nlp(lower_sentence.strip())

In [108]: tok_lem_sentence = [(token.text, token.lemma_) for token in doc]
          tok_lem_sentence[:15]

Out[108]: [('what', 'what'),
           ('is', 'be'),
           ('nlp', 'nlp'),
           (' \n', ' \n'),
           ('natural', 'natural'),
           ('language', 'language'),
           ('processing', 'processing'),
           ('or', 'or'),
           ('nlp', 'nlp'),
           ('for', 'for'),
           ('short', 'short'),
           ('is', 'be'),
           ('broadly', 'broadly'),
           ('defined', 'define'),
           ('as', 'as')]
```

'nlp' is installed model

token.text : tokenized
token.lemma_ : lemmatized

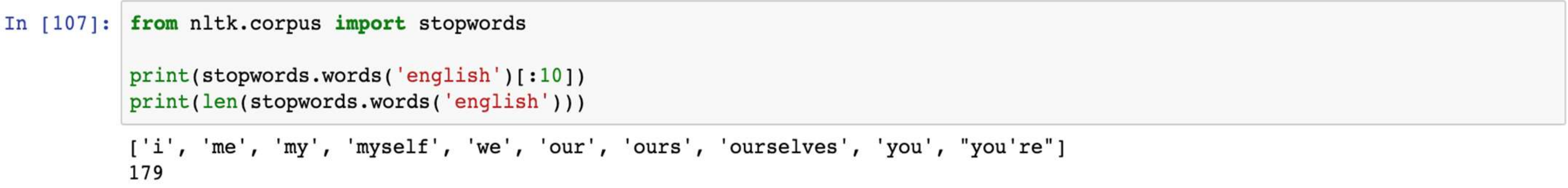
NLP preprocessing



4. Removing Stopwords

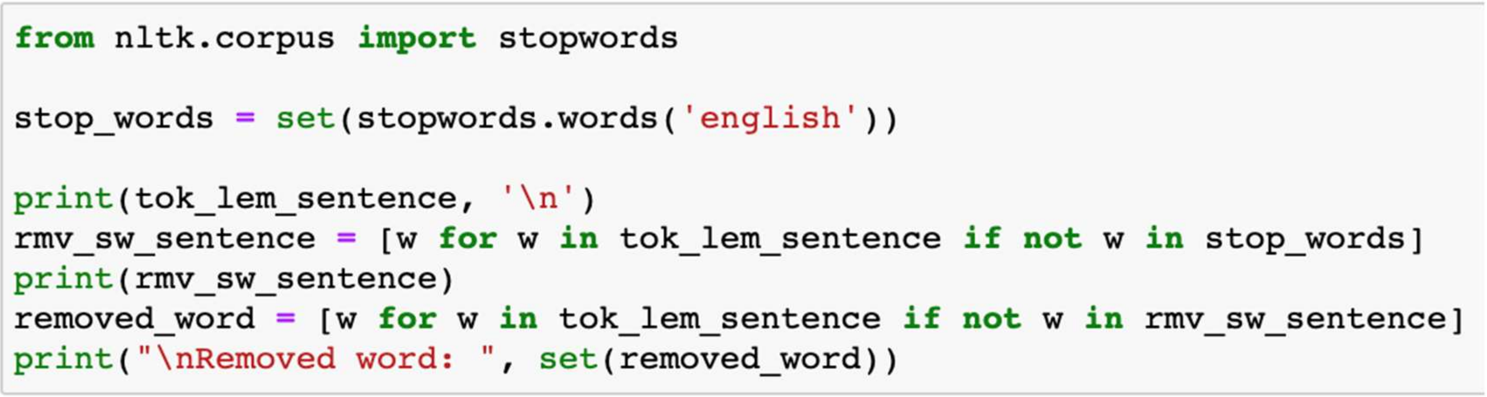
# If No 'stopwords' exists
then run nltk.download('stopwords') first
to download them.

- Stopwords
  - a commonly used word such as ‘the’, ‘a’, ‘an’, ‘in’.



```
In [107]: from nltk.corpus import stopwords

          print(stopwords.words('english')[:10])
          print(len(stopwords.words('english')))

          ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"]
          179
```

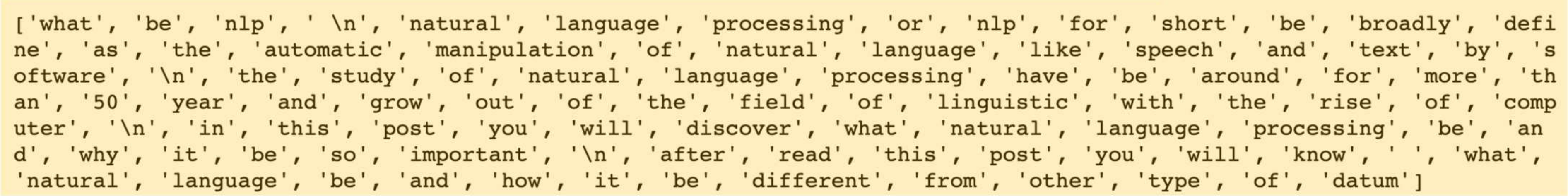
  - Removing stop words with NLTK



```
from nltk.corpus import stopwords

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

print(tok_lem_sentence, '\n')
rmv_sw_sentence = [w for w in tok_lem_sentence if not w in stop_words]
print(rmv_sw_sentence)
removed_word = [w for w in tok_lem_sentence if not w in rmv_sw_sentence]
print("\nRemoved word: ", set(removed_word))
```

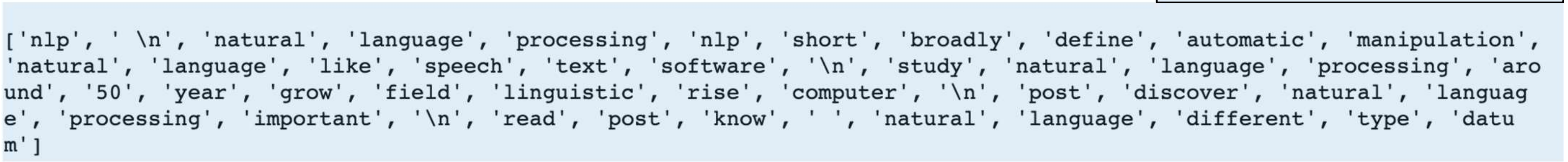
4. Removing Stopwords

Before removing stopwords



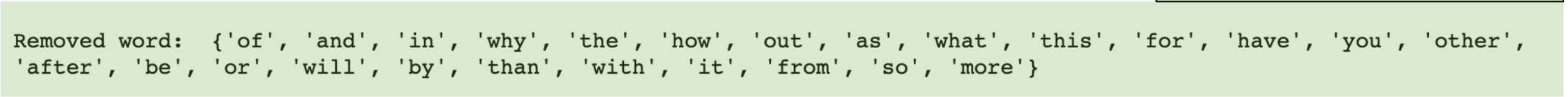
```
['what', 'be', 'nlp', ' \n', 'natural', 'language', 'processing', 'or', 'nlp', 'for', 'short', 'be', 'broadly', 'defi
ne', 'as', 'the', 'automatic', 'manipulation', 'of', 'natural', 'language', 'like', 'speech', 'and', 'text', 'by', 's
oftware', '\n', 'the', 'study', 'of', 'natural', 'language', 'processing', 'have', 'be', 'around', 'for', 'more', 'th
an', '50', 'year', 'and', 'grow', 'out', 'of', 'the', 'field', 'of', 'linguistic', 'with', 'the', 'rise', 'of', 'comp
uter', '\n', 'in', 'this', 'post', 'you', 'will', 'discover', 'what', 'natural', 'language', 'processing', 'be', 'an
d', 'why', 'it', 'be', 'so', 'important', '\n', 'after', 'read', 'this', 'post', 'you', 'will', 'know', ' ', 'what',
'natural', 'language', 'be', 'and', 'how', 'it', 'be', 'different', 'from', 'other', 'type', 'of', 'datum']
```

After removing stopwords



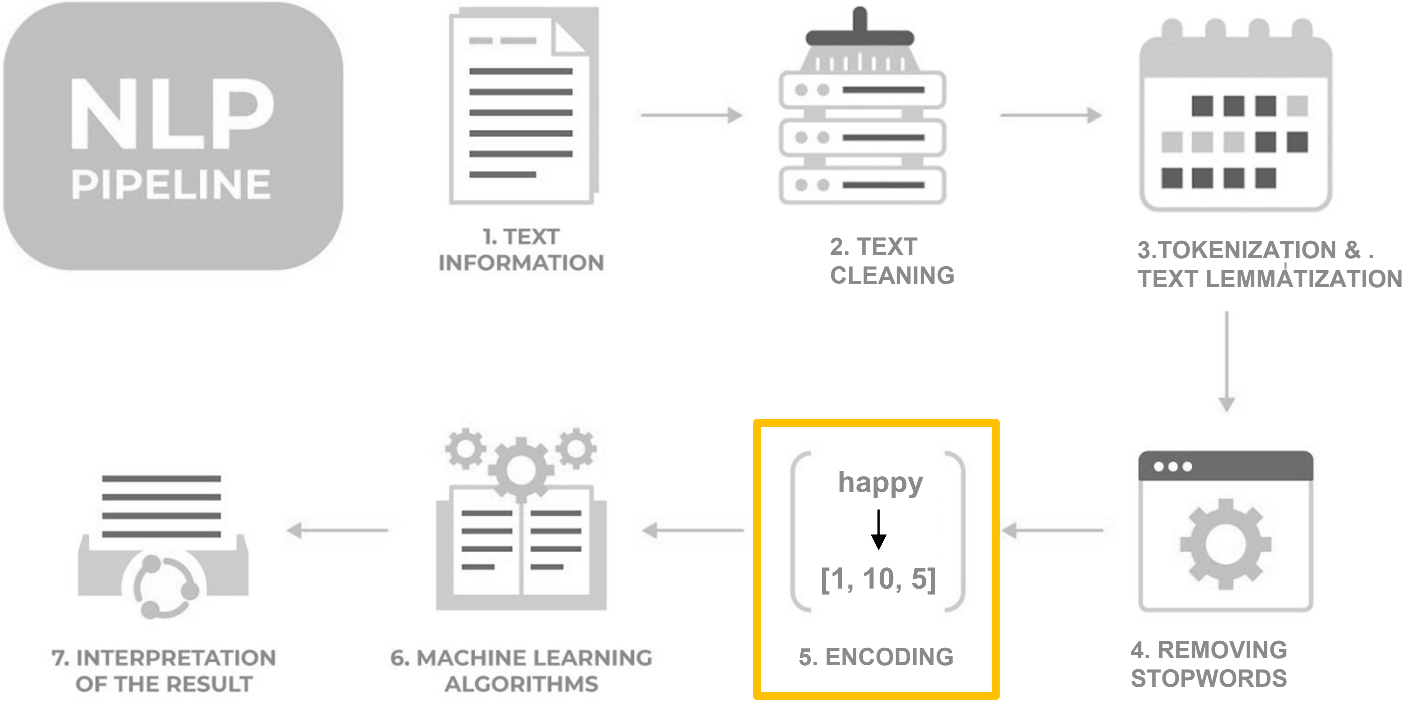
```
['nlp', ' \n', 'natural', 'language', 'processing', 'nlp', 'short', 'broadly', 'define', 'automatic', 'manipulation',
'natural', 'language', 'like', 'speech', 'text', 'software', '\n', 'study', 'natural', 'language', 'processing', 'aro
und', '50', 'year', 'grow', 'field', 'linguistic', 'rise', 'computer', '\n', 'post', 'discover', 'natural', 'languag
e', 'processing', 'important', '\n', 'read', 'post', 'know', ' ', 'natural', 'language', 'different', 'type', 'datu
m']
```

Removed stopwords



```
Removed word:  {'of', 'and', 'in', 'why', 'the', 'how', 'out', 'as', 'what', 'this', 'for', 'have', 'you', 'other',
'after', 'be', 'or', 'will', 'by', 'than', 'with', 'it', 'from', 'so', 'more'}
```

NLP preprocessing

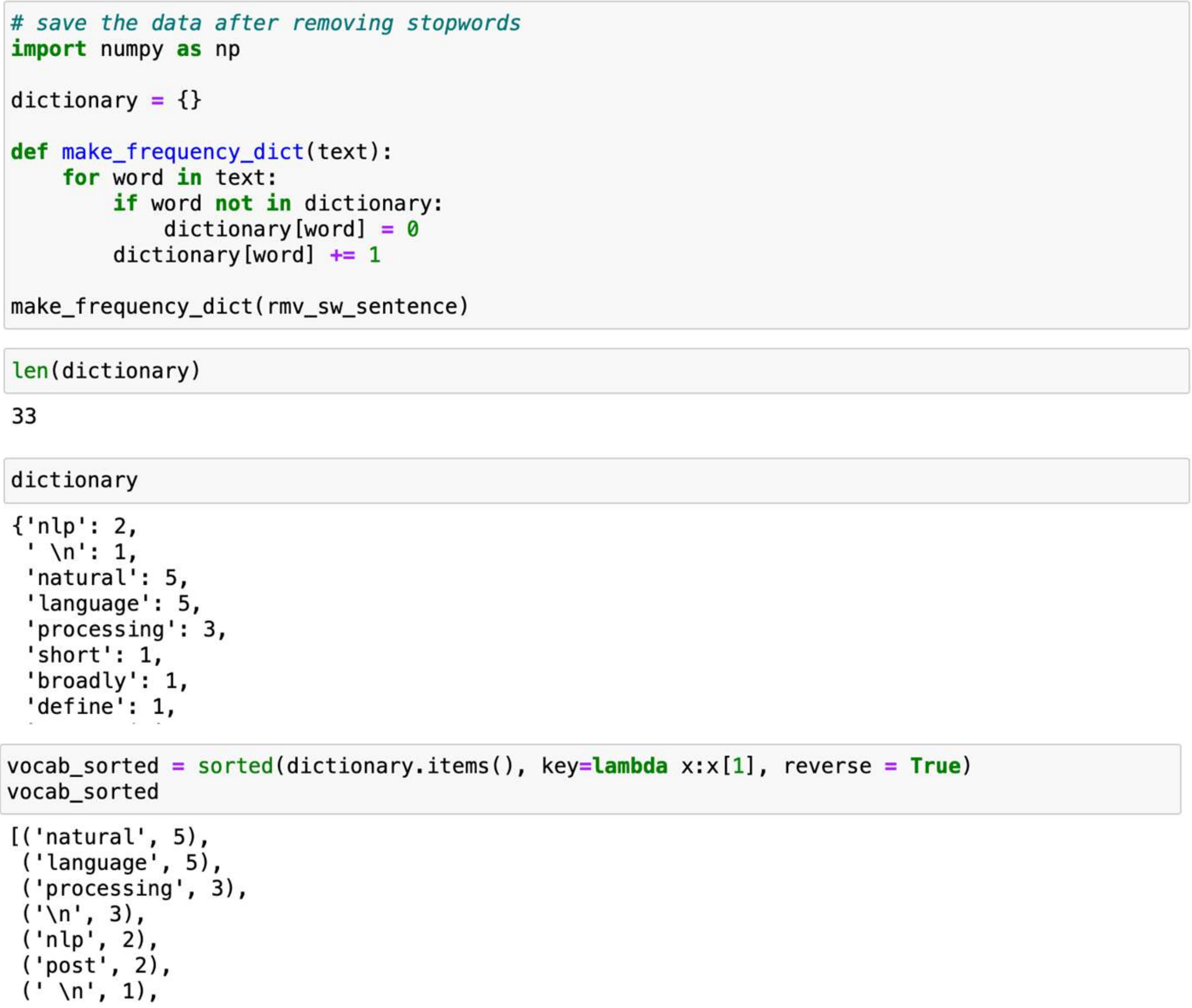

NLP
PIPELINE
1. TEXT INFORMATION
2. TEXT CLEANING
3.TOKENIZATION & . TEXT LEMMATIZATION
4. REMOVING STOPWORDS
happy
[1, 10, 5]
5. ENCODING
6. MACHINE LEARNING ALGORITHMS
7. INTERPRETATION OF THE RESULT

5. Encoding

- Integer Encoding
  - What is integer encoding?
    - Mapping unique integers to words
  - How to?
    - Make a frequency-based dictionary
      - Step1. Make a frequency dictionary
      - Step2. Make a dictionary based on frequency
      - Step3. Add 'OOV' index for unknown words (OOV: Out of Vocabulary)
    - Encoding the words

5. Encoding – a frequency based dictionary

- Step1. Make a frequency dictionary



```
# save the data after removing stopwords
import numpy as np

dictionary = {}

def make_frequency_dict(text):
    for word in text:
        if word not in dictionary:
            dictionary[word] = 0
        dictionary[word] += 1

make_frequency_dict(rmv_sw_sentence)
```

```
len(dictionary)
```

```
33
```

```
dictionary
```

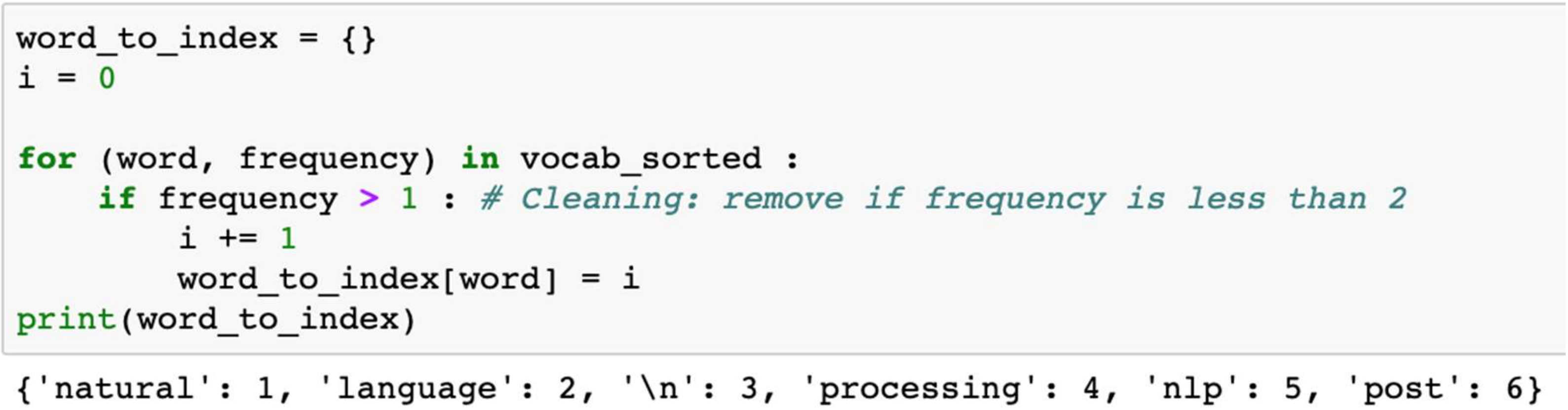
```
{'nlp': 2,
 ' \n': 1,
 'natural': 5,
 'language': 5,
 'processing': 3,
 'short': 1,
 'broadly': 1,
 'define': 1,
```

```
vocab_sorted = sorted(dictionary.items(), key=lambda x:x[1], reverse = True)
vocab_sorted
```

```
[('natural', 5),
 ('language', 5),
 ('processing', 3),
 ('\n', 3),
 ('nlp', 2),
 ('post', 2),
 (' \n', 1),
```

5. Encoding – a frequency based dictionary

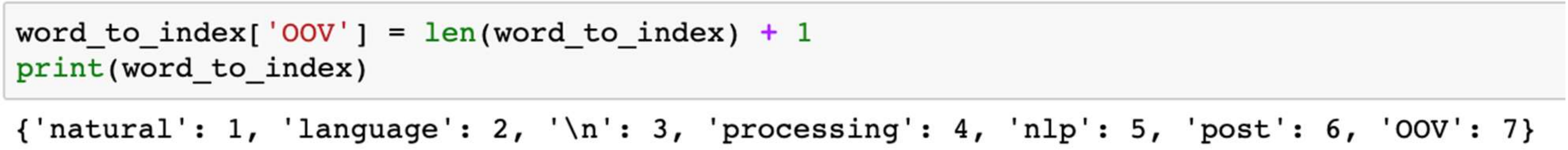
- Step2. Make a dictionary based on frequency



```
word_to_index = {}
i = 0

for (word, frequency) in vocab_sorted :
    if frequency > 1 : # Cleaning: remove if frequency is less than 2
        i += 1
        word_to_index[word] = i
print(word_to_index)
```

```
{'natural': 1, 'language': 2, '\n': 3, 'processing': 4, 'nlp': 5, 'post': 6}
```

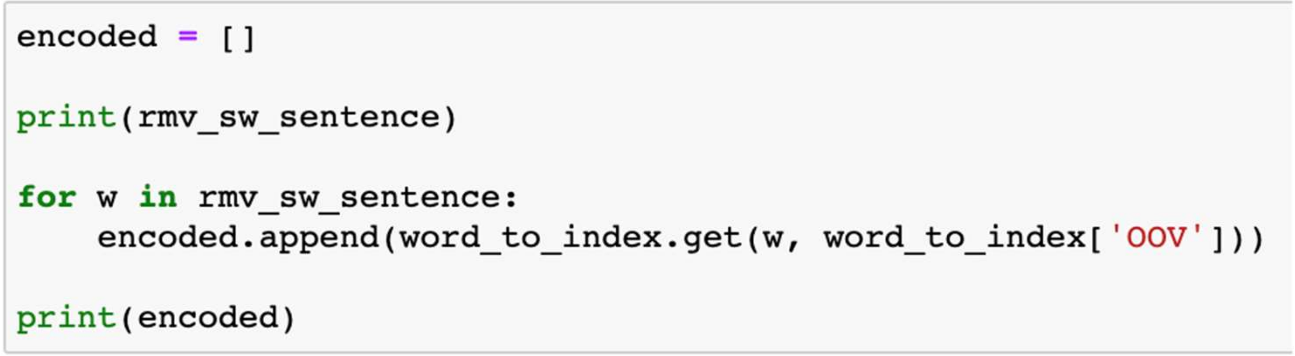
- Step3. Add ‘OOV’ index for unknown words



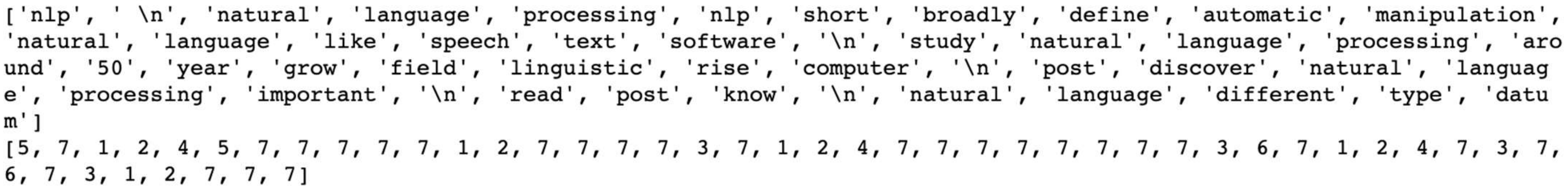
```
word_to_index['OOV'] = len(word_to_index) + 1
print(word_to_index)
```

```
{'natural': 1, 'language': 2, '\n': 3, 'processing': 4, 'nlp': 5, 'post': 6, 'OOV': 7}
```

5. Encoding – Encoding the words



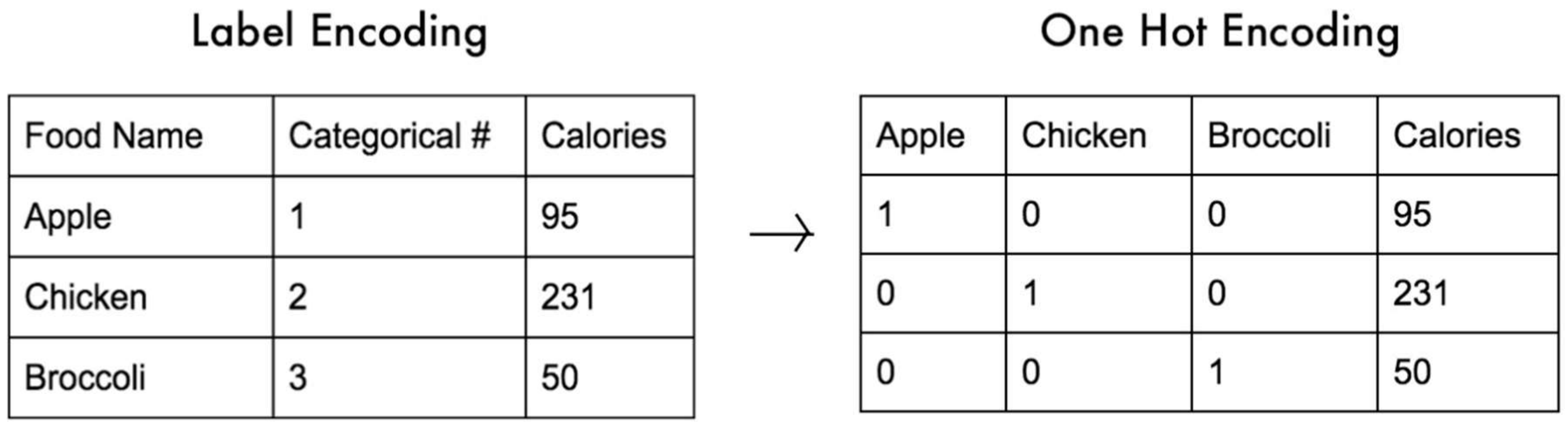
```
encoded = []

print(rmv_sw_sentence)

for w in rmv_sw_sentence:
    encoded.append(word_to_index.get(w, word_to_index['OOV']))

print(encoded)
```



```
['nlp', ' \n', 'natural', 'language', 'processing', 'nlp', 'short', 'broadly', 'define', 'automatic', 'manipulation',
'natural', 'language', 'like', 'speech', 'text', 'software', '\n', 'study', 'natural', 'language', 'processing', 'aro
und', '50', 'year', 'grow', 'field', 'linguistic', 'rise', 'computer', '\n', 'post', 'discover', 'natural', 'languag
e', 'processing', 'important', '\n', 'read', 'post', 'know', '\n', 'natural', 'language', 'different', 'type', 'datu
m']
[5, 7, 1, 2, 4, 5, 7, 7, 7, 7, 7, 1, 2, 7, 7, 7, 7, 3, 7, 1, 2, 4, 7, 7, 7, 7, 7, 7, 7, 7, 3, 6, 7, 1, 2, 4, 7, 3, 7,
6, 7, 3, 1, 2, 7, 7, 7]
```

5. Encoding

- One-hot encoding
  - Problem in Integer encoding
    - Giving the higher numbers higher weights.
  - What is one-hot encoding?
    - A method to quantify categorical data
    - Producing a vector with length equal to the number of categories in the data set.



Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

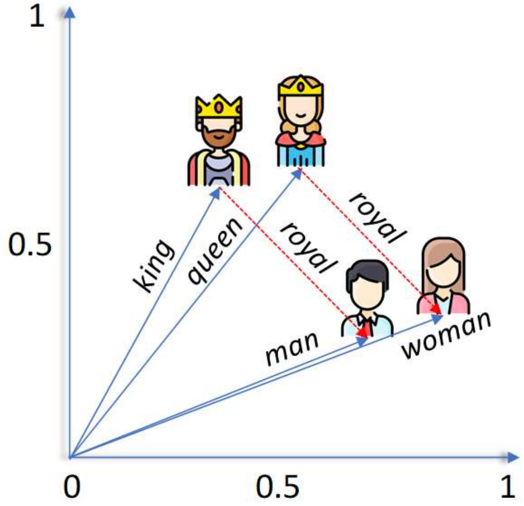
→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Word Embedding

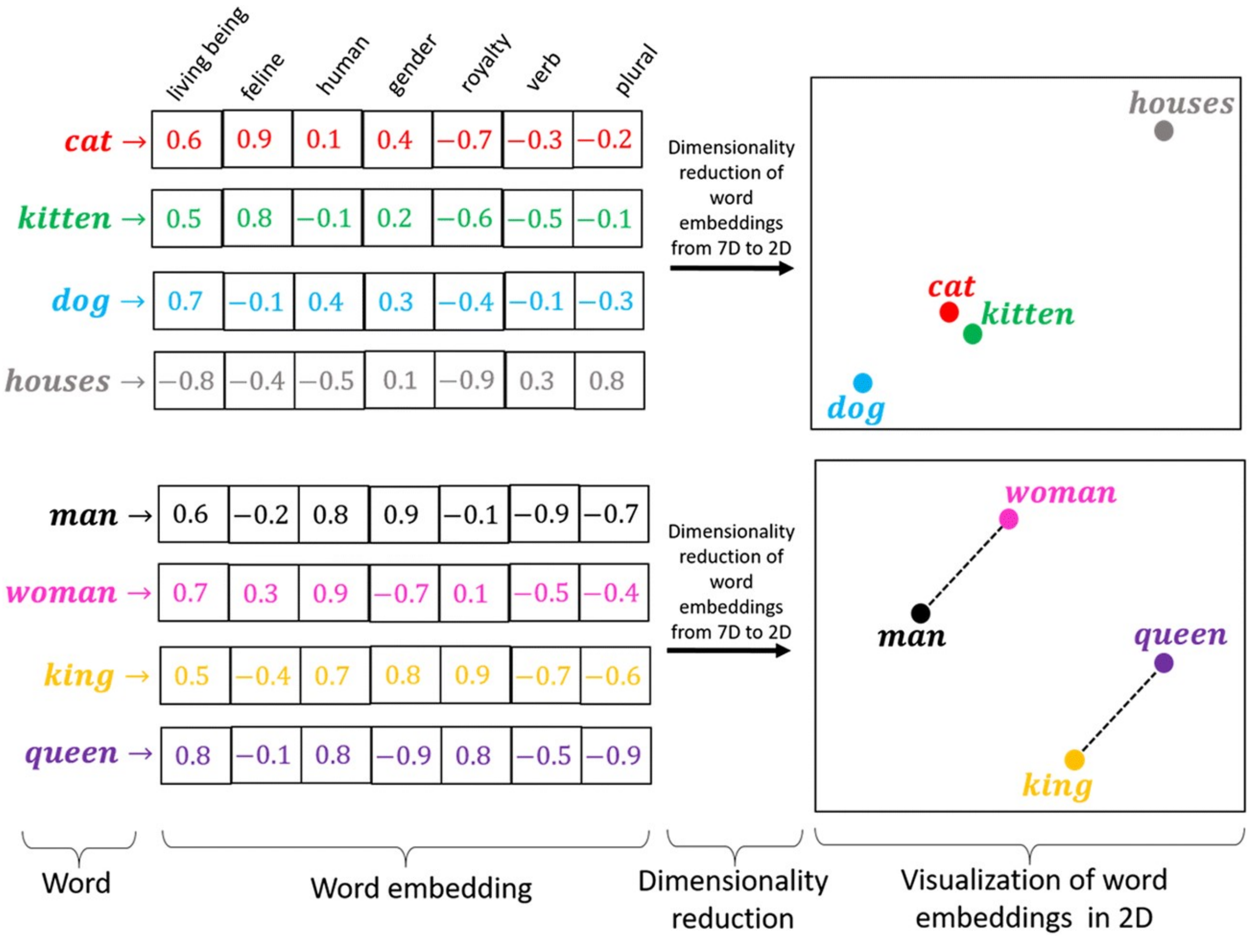
- Word Embedding
  - Problem in One-hot encoding
    - Vector length = the number of words in the dictionary
    - Losing the context of the sentence
  - What is word embedding?
    - Vector representations of a particular word
    - Capturing context of a word in a document, semantic and syntactic similarity, relation with other words
  - Word embedding type
    - Frequency based embedding
    - Prediction based embedding
      - *Example : Word2Vec*



https://towardsdatascience.com/deep-learning-for-nlp-word-embeddings-4f5c90bcdab5

NLP preprocessing

- Word Embedding



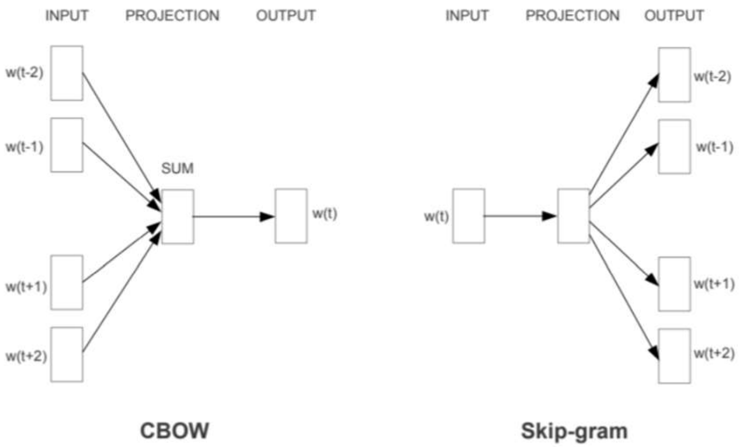
	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

https://medium.com/@hari4om/word-embedding-d816f643140

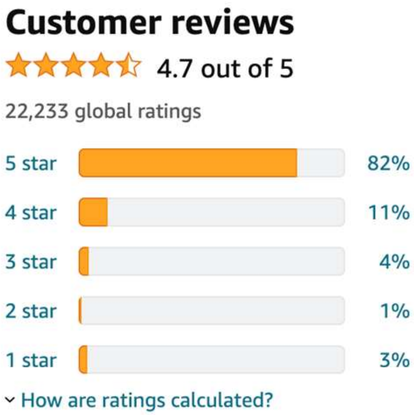
NLP preprocessing

- Word Embedding
  - Prediction based embedding
    - CBOW(continuous bag of words)
      - the distributed representations of context (or surrounding words) are combined to **predict the word in the middle**.
    - Skip-gram
      - the distributed representation of the input word is used to **predict the context**.



https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314

Exercise (HW8)



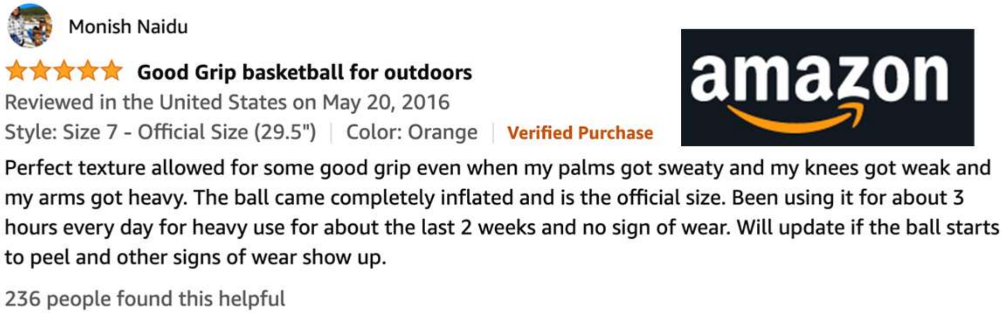
Customer reviews

★★★★☆ 4.7 out of 5

22,233 global ratings

5 star	82%
4 star	11%
3 star	4%
2 star	1%
1 star	3%

› How are ratings calculated?



Monish Naidu

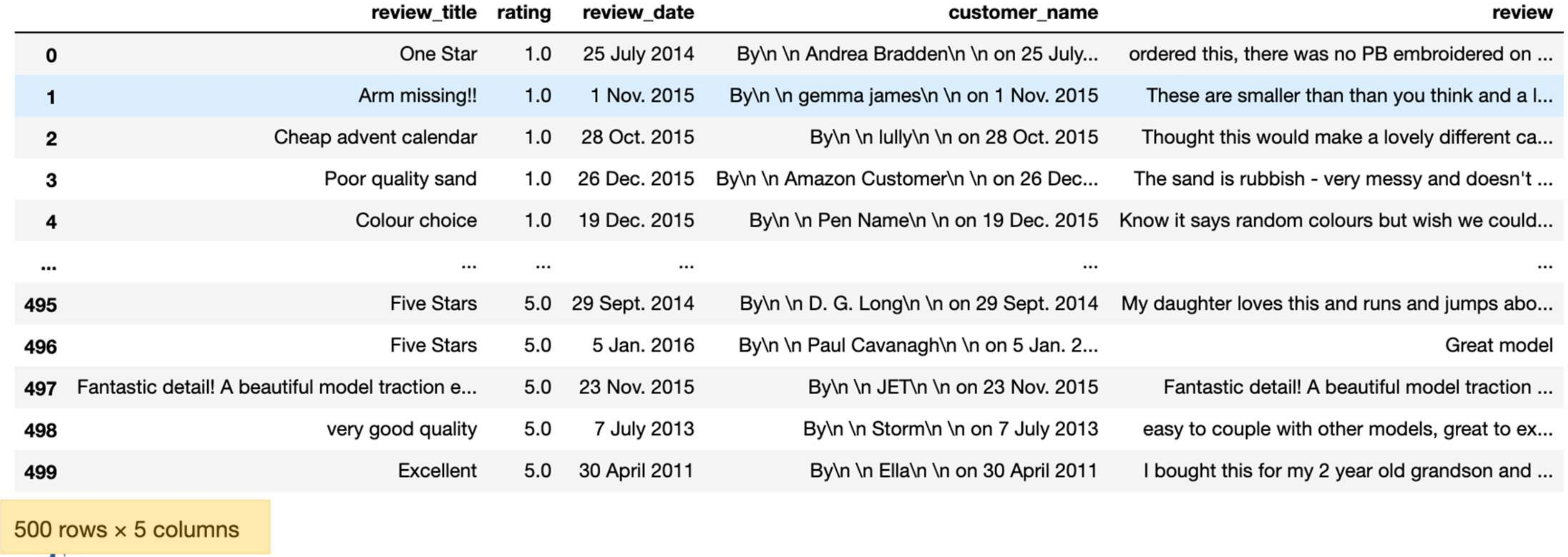
★★★★★ **Good Grip basketball for outdoors**

Reviewed in the United States on May 20, 2016

Style: Size 7 - Official Size (29.5") | Color: Orange | Verified Purchase

Perfect texture allowed for some good grip even when my palms got sweaty and my knees got weak and my arms got heavy. The ball came completely inflated and is the official size. Been using it for about 3 hours every day for heavy use for about the last 2 weeks and no sign of wear. Will update if the ball starts to peel and other signs of wear show up.

236 people found this helpful



	review_title	rating	review_date	customer_name	review
0	One Star	1.0	25 July 2014	By\n \n Andrea Bradden\n \n on 25 July...	ordered this, there was no PB embroidered on ...
1	Arm missing!!	1.0	1 Nov. 2015	By\n \n gemma james\n \n on 1 Nov. 2015	These are smaller than than you think and a l...
2	Cheap advent calendar	1.0	28 Oct. 2015	By\n \n lully\n \n on 28 Oct. 2015	Thought this would make a lovely different ca...
3	Poor quality sand	1.0	26 Dec. 2015	By\n \n Amazon Customer\n \n on 26 Dec...	The sand is rubbish - very messy and doesn't ...
4	Colour choice	1.0	19 Dec. 2015	By\n \n Pen Name\n \n on 19 Dec. 2015	Know it says random colours but wish we could...
...	...	...	...	...	...
495	Five Stars	5.0	29 Sept. 2014	By\n \n D. G. Long\n \n on 29 Sept. 2014	My daughter loves this and runs and jumps abo...
496	Five Stars	5.0	5 Jan. 2016	By\n \n Paul Cavanagh\n \n on 5 Jan. 2...	Great model
497	Fantastic detail! A beautiful model traction e...	5.0	23 Nov. 2015	By\n \n JET\n \n on 23 Nov. 2015	Fantastic detail! A beautiful model traction ...
498	very good quality	5.0	7 July 2013	By\n \n Storm\n \n on 7 July 2013	easy to couple with other models, great to ex...
499	Excellent	5.0	30 April 2011	By\n \n Ella\n \n on 30 April 2011	I bought this for my 2 year old grandson and ...

500 rows × 5 columns

Exercise – Part0

Load ’amazon_train_df.csv’ data.

1. Read in csv file and clean the data
2. Change the name of columns
3. Remove HTML (if necessary)
4. Remove punctuation and Replacing with lower case
5. Lemmatization + Tokenization
6. Remove stopwords

Exercise – Part1

- print the most 5 frequent words for each review data from ’amazon_train_df.csv’ (10 points)

- output format.

  review 1: Z, K, C, D, E

  review 2: S, F, G, B, M

  review 3: …

  review 4: …

  review 5: T, X, P, Z, K

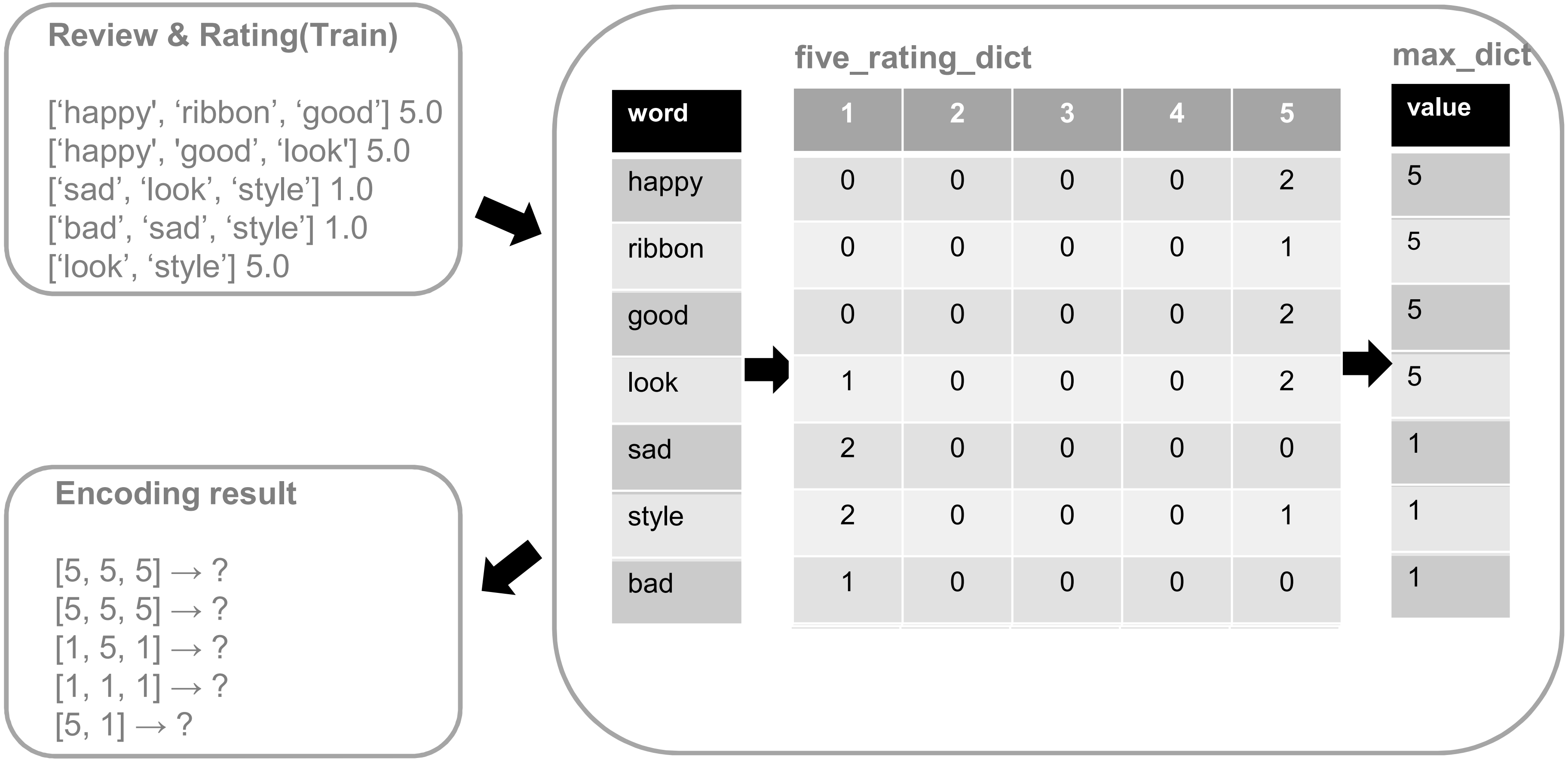
Exercise – Part2

- **v1** : Understanding the meaning of word through a frequency-based dictionary
  - Make a word-to-index dictionary from the train data set (5 points)

- **v2** : Using the word-to-index dictionary we made above,
  - Make a word-to-rating dictionary (5 points)

Exercise

- Make a simple word-to-rating dictionary



Review & Rating(Train)

[‘happy', ‘ribbon’, ‘good’] 5.0
[‘happy', 'good’, ‘look'] 5.0
[‘sad’, ‘look’, ‘style’] 1.0
[‘bad’, ‘sad’, ‘style’] 1.0
[‘look’, ‘style’] 5.0

five_rating_dict

max_dict

word	1	2	3	4	5	value
happy	0	0	0	0	2	5
ribbon	0	0	0	0	1	5
good	0	0	0	0	2	5
look	1	0	0	0	2	5
sad	2	0	0	0	0	1
style	2	0	0	0	1	1
bad	1	0	0	0	0	1

Encoding result

[5, 5, 5] → ?
[5, 5, 5] → ?
[1, 5, 1] → ?
[1, 1, 1] → ?
[5, 1] → ?

Exercise – Part4

- **v3: Using the word-to-rating dictionary**
  1. Load ‘’amazon_test_df.csv’ data
     - Do the same process of Part 0
  2. Encode the test review data using the word-to-index dictionary (integer encoding)
  3. Predict the rating of test review (5 points)
     - How are you going to decide(predict) the rating value?
  4. Suggest how to evaluate your predicted result. (5 points)
  5. Suggest how to improve your result. (Bonus 5 points)

Encoding result

[5, 5, 5] → ?
[5, 5, 5] → ?
[1, 5, 1] → ?
[1, 1, 1] → ?
[5, 1] → ?

Exercise – Extra Assignment

- **Run the chatbot program and get results with your own text inputs.**

  (bonus 5 points)

- **Run the text2bible program and get results with your own text inputs.**

  (bonus 5 points)