SAMSUNG

Samsung Innovation Campus

Artificial Intelligence Course

Together for Tomorrow!
Enabling People
Education for Future Generations

Chapter 7.

Natural Language Processing and Language Models for Text Mining

Artificial Intelligence Course

Chapter Description

Chapter objectives

- ✓ Process input text from sources of various text formats in order to extract high quality information.
- ✓ Structure language and derive patterns by natural language processing. and evaluate and analyze these results to utilize them for real world applications.

Chapter contents

- ✓ Unit 1. Text Mining
- ✓ Unit 2. Text Preprocessing
- ✓ Unit 3. Language Model
- ✓ Unit 4. Natural Language Processing with Keras

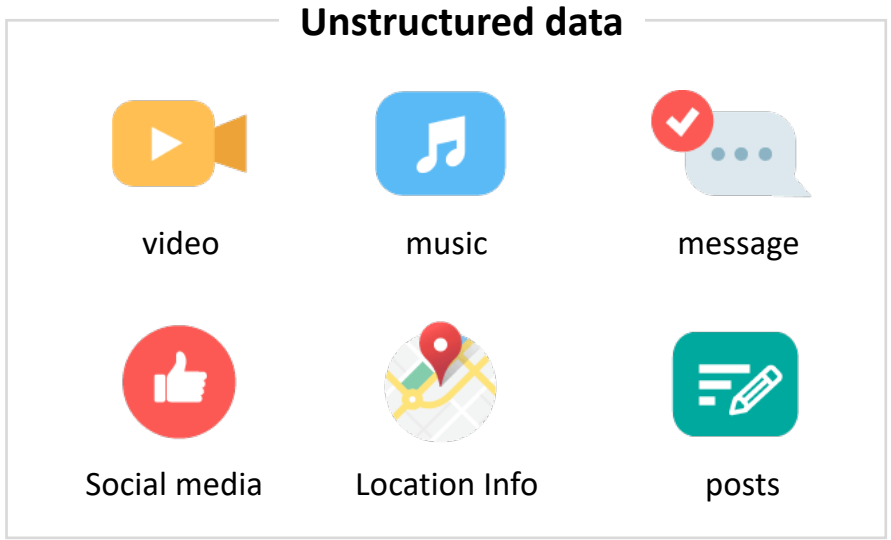
Unit 1.

Text Mining

Text Mining

What is unstructured data?

- Data that is not yet structured. It does not have a defined data model (structure)
- Documents, videos, or audios that have a large amount of data but with varying structures and forms.
- Books, journals, documents, metadata, health records, audio, video, analog data, images, files, and also e-mail messages, webpages, word-processor documents are all composed of unstructured texts.

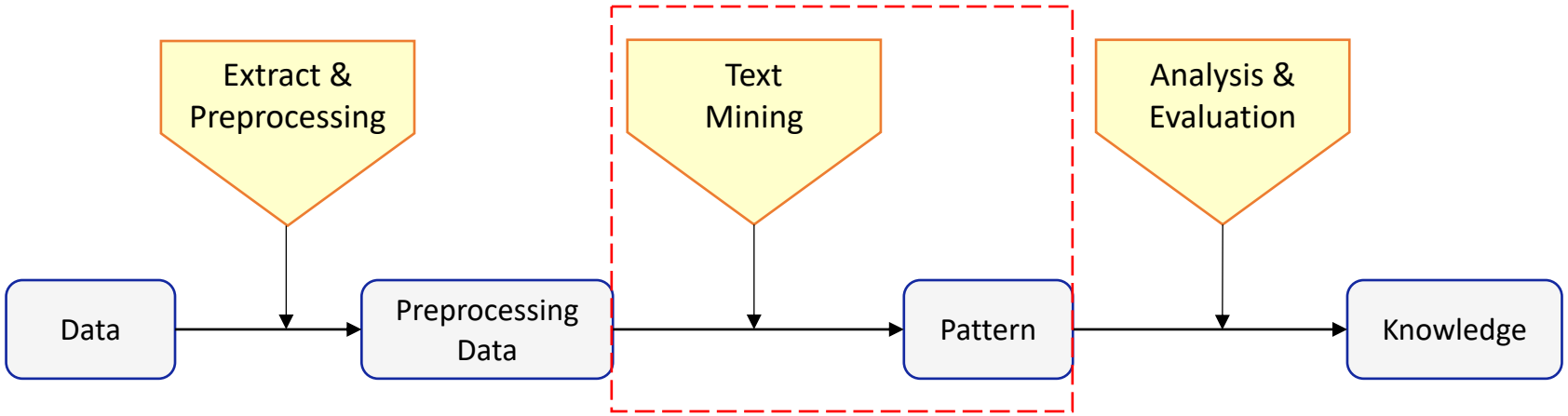


Analyzing unstructured data

- Manually enter tags into metadata to structure texts.
- Skilled data structuring based on text-mining uses a method that creates a tag so that a word in the text and a part in the speech correspond.
- Uses software that builds structures processable by machines.
- Analysis that enables semantic deduction from texts, syntax, and other small or large patterns. This analysis uses algorithms that inspect all internal structure of human communication in word unit by forming them into linguistic, auditory, visual structure.

What is Text Mining??

- Text mining or text analytics is technology that extracts useful information from unstructured text data
- To be more specific, it is finding practical patterns from large amount of document data by applying mechanical algorithm and statistical techniques.



Text mining vs data mining

- Data mining extracts useful and valuable patterns from structured data.
- In contrast, text mining extracts named-entities, patterns or information on word-sentence relationships from unstructured data composed of natural language

Text Mining Application Example in Real World

- **Used for marketing:** Corporations collect and analyze posts on twitter that mention their brand names or messages with the customers.
  - They examine if users mention certain topics in certain time; if users have positive or negative connotations; how keywords change within time; if customers' keywords changed before and after a campaign; if a promotion generated a word of mouth; if a certain group of customers react; and etc.
  - With this information, corporations can closely monitor marketing activities and control reputation management and build competitive strategies against competing brands based on feedback monitoring.
- **Supporting data for various industries:** It can support data from fields such as politics, environment and medicine, and business areas such as manufacturing, facilities, and marketing.
- **Factories predicting breakdown**: Factories can predict facilities breakdown from documents recorded by facilities maintenance workers.
- **Checking product reviews**: Customers have access to information of a product's performance or issues from product analysis or reviews.
- **News analysis**: It can show popular issues within time flow and discover experts of a certain field by analyzing topics of news reports or speech statements.

Ejemplos:

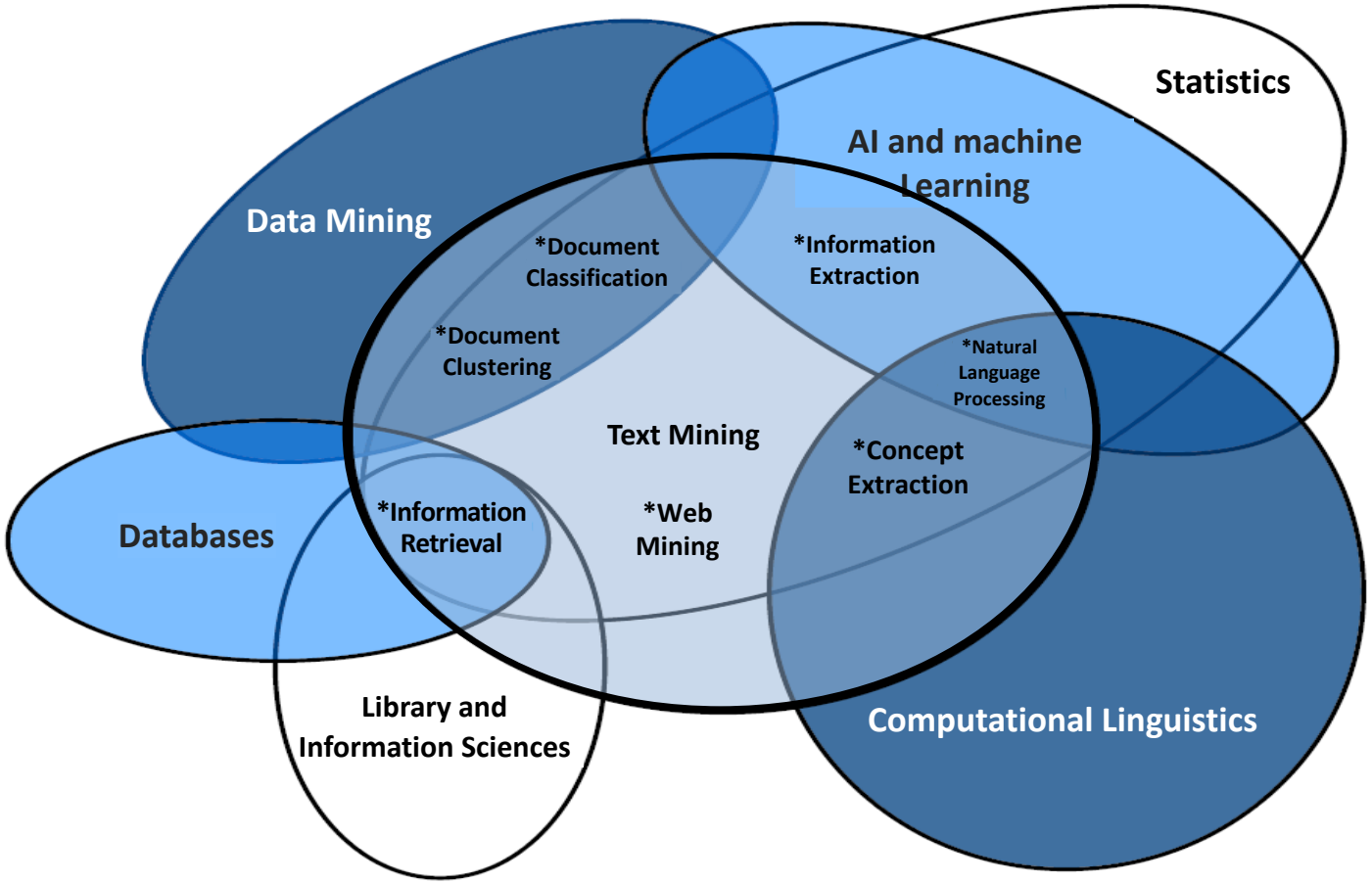
Text mining vs Natural Language Processing (NLP) (1/3)

- **NLP is a field of AI that handles communication.** It enables machines to generate and analyze human language (generate and understand natural language)
  - NLP can process many types of voice including slangs, dialects and grammatical errors. Machine learning constructs the foundation for this methodology. Applications of NPL are search engine, AI chatbots, grammar correction apps etc.
- On the other hand, **text mining is a sub-category of data mining science.** Data mining science includes data search, data mining and machine learning methods.
- More than 80% of organizations worldwide utilize textual information. **NLP recognizes text and voice while text mining evaluates the quality of text.**

Text mining vs Natural Language Processing (NLP) (2/3)

- Different tools are used for text mining and NLP.
  - In order to construct high-quality NLP system, you must be proficient in neural network, deep learning and NLTK.
  - Text mining system is a technique like Levenshtein Distance, Cosine similarity or Feature Hashing.
  - You must be familiar with text processing programming language and statistical models like Perl or Python.
- **NLP offers understanding** of explained emotions and grammatical structure, and detect intent behind a text.
  - This assists fluent translation of text to another language.
- Meanwhile, **text mining discovers relationship** among words in the text.
  - It analyzes frequency of word use and patterns.

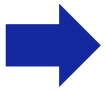
Text Mining vs Natural Language Processing (NLP) (3/3)



Text mining algorithms and their topics (1/3)

- Text mining handles topics listed on the right table by applying various algorithms listed on the left table. (1/2)

Algorithm	Area
Naïve bayes	Document classification
Conditional random fields	Information extraction
Hidden Markov models	Information extraction
K-means	Clustering
Singular value decomposition (SVD)	Document classification
Logistic regression	Document classification
Decision trees	Document classification



Topics	Practice Area
Keyword search	Search and information retrieval
Inverted index	Search and information retrieval
Document clustering	Document Clustering
Document similarity	Document Clustering
Feature selection	Document classification
Sentiment analysis	Document classification Web mining
Dimensionality reduction	Document classification
eDiscovery	Document classification

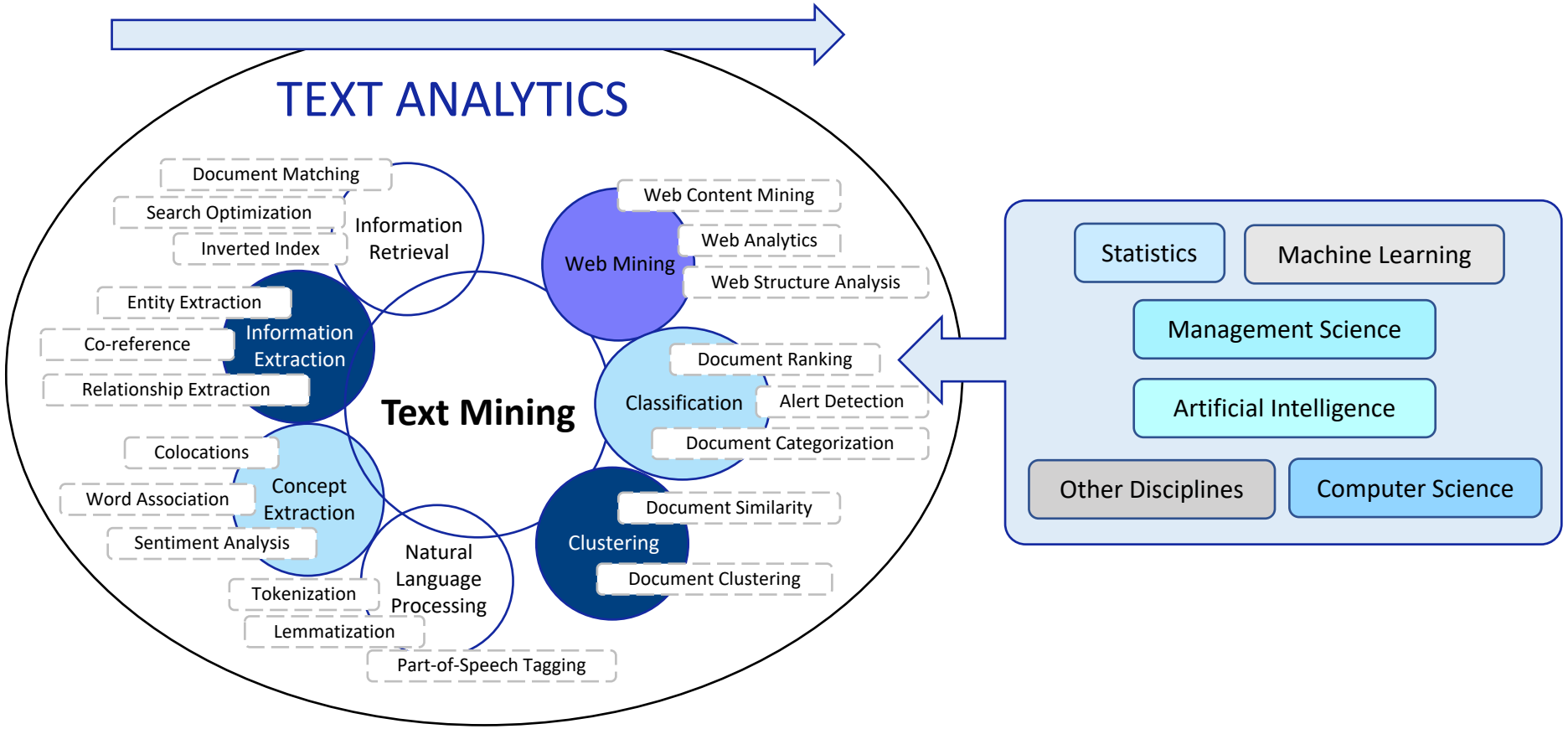
Text mining algorithms and their topics (2/3)

- Text mining handles topics listed on the right table by applying various algorithms listed on the left table.

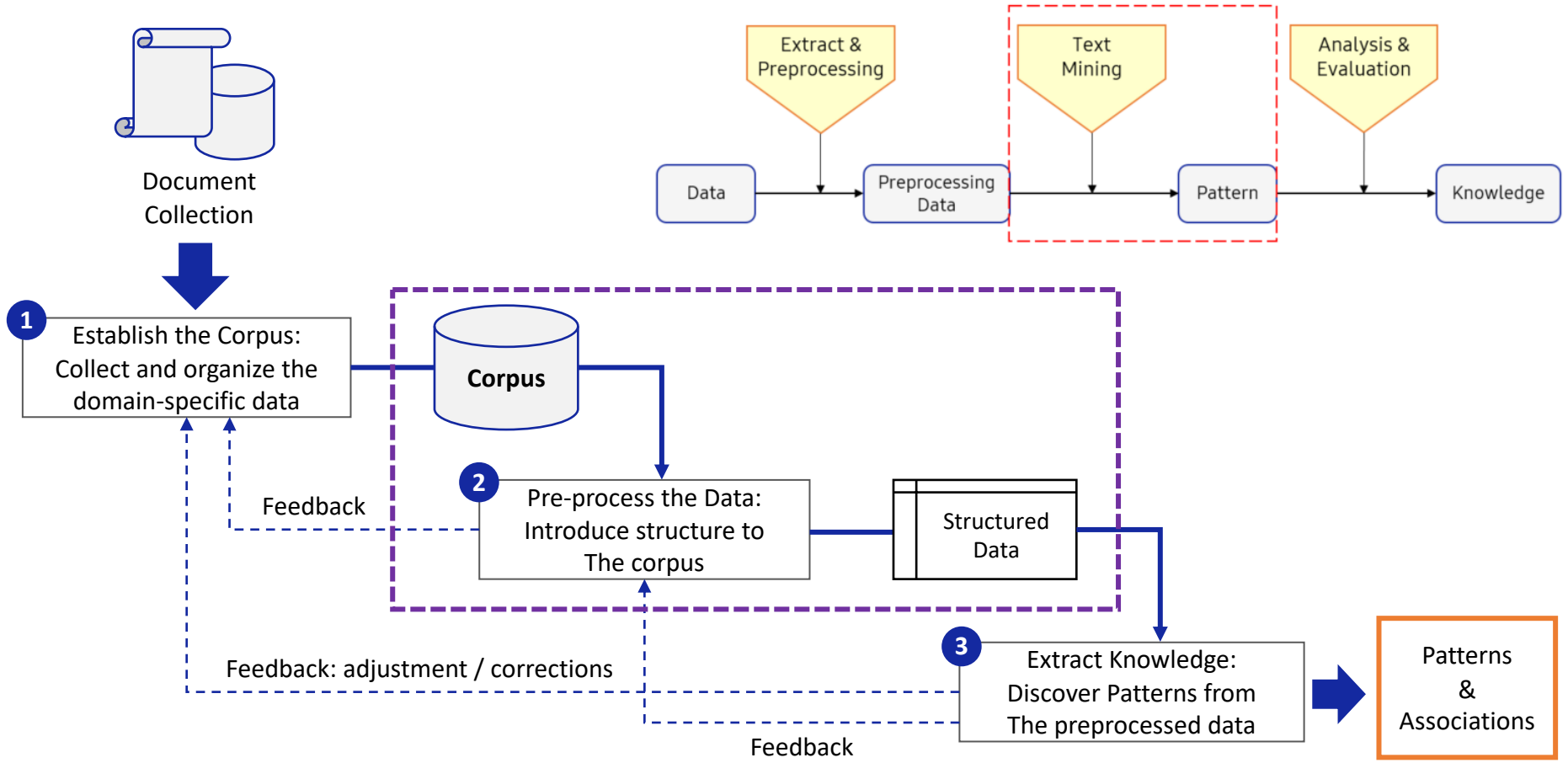
Algorithm	Area
Neural network	Document classification
Support vector machines	Document classification
MARSplines	Document classification
Link analysis	Concept extraction
k-nearest neighbors	Document classification
Word clustering	Concept extraction
Regression	classification

Topic	Practice Area
Web crawling	Web mining
Link analytics	Web mining
Entity extraction	Information extraction
Link extraction	Information extraction
Part of speech tagging	Natural language processing
Tokenization	Natural language processing
Question answering	Natural language processing Search and information retrieval
Topic modeling	Concept extraction
Synonym identification	Concept extraction

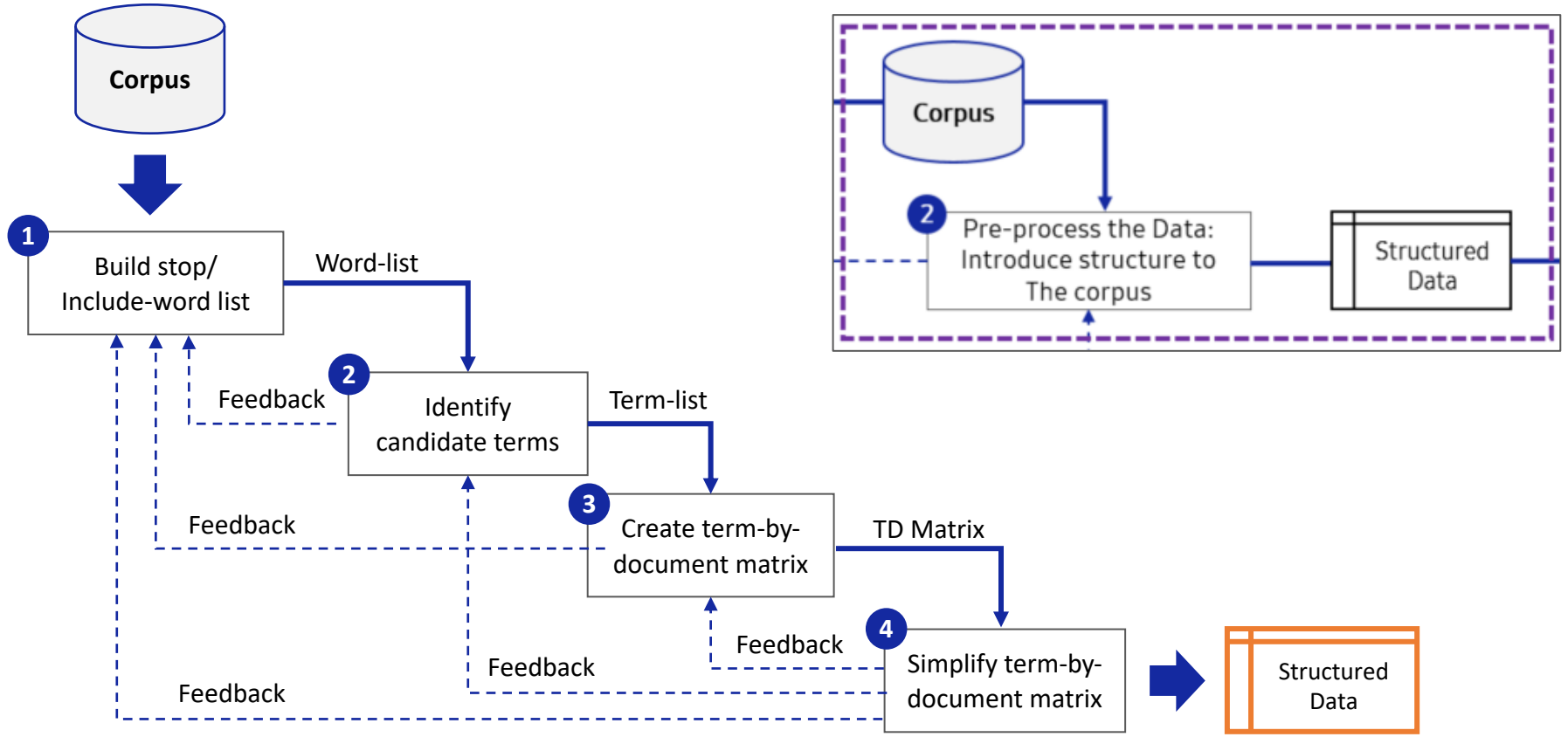
Text mining algorithms and their topics (3/3)



Basic Procedure for Text Mining



Basic Procedure for Text Mining



Unit 1.

Text Mining

Collecting Data from Different Resources

- Collecting text is a process of establishing the plan for collecting data and gathering data suitable for the task's objective and characteristics.
  - Collecting text is an important process that determines the quality of text analysis services.
  - You make a detailed plan after reviewing time period, cost, possibility of personal information infringement, and inclusion of data categories meeting the objective. According to the plan, you perform pre-test and proceed with data collection from different resources.

Collecting Data from Different Resources

Collecting text is a process of establishing the plan for collecting data and gathering data suitable for the task's objective and characteristics.

- Various collecting techniques are used according to data type and features. Main techniques are listed below.

Technique	Features	Data Type
Crawling	- Collects web documents and information on the web, such as social media, news and web information - Follows URL link and collect repetitively	Web document
Scraping	- Collects information from a single website (or document)	Web document
FTP	- Transmits and receives files from internet servers using TCP/IP protocol - Considers using SFTP for reinforced security - Considers constructing exclusive network for linked servers	File
Open API	- Offers data collecting method with an open API that allows easy access to service, information and data.	Real time data
RSS	- XML based content distribution protocol that allows sharing up-to-date web-based information	Content

Text Data from Websites

EjemplosCollectingData.ipynb

- Download a webpage with the **Requests** library
  - This is OK when the exact URL is known.
  - If log-in is required, use the Selenium library instead.
  - HTML content without parsing.

Ex

In [1]:

```
1 import requests as rq
2 res = rq.get("https://en.wikipedia.org/wiki/Machine_learning")
3 res.status_code
4 print(res.text)
```

Line 1-3

- If the status code is 200, then OK.

Parsing HTML with the **BeautifulSoup4** library:

- The downloaded HTML should be parsed in order to access the desired content.

Ex

In [1]:

```
1 import requests, bs4
2 res = reqquests.get("https://en.wikipedia.org/wiki/Machine_learning")
3 soup = bs4.BeautifulSoup(res.text, 'html.parser')
4 x=soup.find_all('p')
5 text =' '
6 for i in range(len(x)):
7 text += x[i].text.strip() +'\n'
8 print(text)
```

Line 1-3

- Returns a BeautifulSoup object.

Parsing HTML with the **BeautifulSoup4** library:

- The downloaded HTML should be parsed in order to access the desired content.

Ex

In [1]:

```
1 import requests, bs4
2 res = reqquests.get("https://en.wikipedia.org/wiki/Machine_learning")
3 soup = bs4.BeautifulSoup(res.text, 'html.parser')
4 x=soup.find_all('p')
5 text =' '
6 for i in range(len(x)):
7 text += x[i].text.strip() +'\n'
8 print(text)
```

Line 1-4

- Get all the paragraphs.

Parsing HTML with the BeautifulSoup4 library:

- The downloaded HTML should be parsed in order to access the desired content.

Ex

In [1]:

```
1 import requests, bs4
2 res = reqquests.get("https://en.wikipedia.org/wiki/Machine_learning")
3 soup = bs4.BeautifulSoup(res.text, 'html.parser')
4 x=soup.find_all('p')
5 text =' '
6 for i in range(len(x)):
7 text += x[i].text.strip() +'\n'
8 print(text)
```

Line 1-7

- Join all the text contents.

Parsing HTML with the BeautifulSoup4 library:

- The downloaded HTML should be parsed in order to access the desired content.

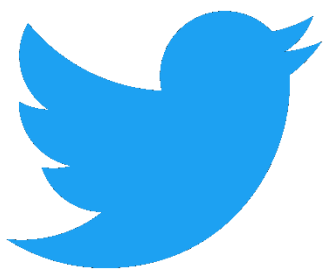
Ex The example from the previous slide produces an output as shown below.

> Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.[1][2]:2 Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.
>
> Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.[3][4] In its application across business problems, machine learning is also referred to as predictive analytics.

Text Data From Twitter

About Twitter:

- Social networking, news and microblogging.
- Users post and interact through short messages known as tweets.
- Used for spreading news and ideas, promotion, building relationship, etc.
- Twitter API allows access to the features of Twitter without having to go through the website.
- Source of text data.



Reading Tweets from Twitter:

- Apply for access following the steps below:

1) Go to https://developer.twitter.com/
2) Create a developer account and then log in.
3) In the Dashboard, press the Create an app button.
4) Fill out the form and then press the Create button.
5) Click on the Keys and tokens tab.

   ⇒ The “Consumer API key” and “Consumer API secret key” are already available.

6) In order to generate the “Access token” , “Access token secret”, click on the Create my access token button.
7) Generate also the “Bearer Token”

Desde mayo de 2003 las cuentas de desarrollador gratuitas (Free) solo permiten postear

Reading Tweets from Twitter:

Ex Using the Tweepy library to fetch tweets: **API Twitter v1.1**

```
1 #!pip install colab_env
2 #!pip install tweepy
3 import colab_env
4 import tweepy
5 import os
6
7 # Key Tokens stored in vars.env (it's a good practice not to have keys in the coc
8 my_consumer_key = os.getenv("API_KEY")
9 my_consumer_secret = os.getenv("API_KEY_SECRET")
10 my_access_token = os.environ["ACCESS_TOKEN"]
11 my_access_secret = os.environ["ACCESS_TOKEN_SECRET"]
12
13 #Initialize the token
14 auth = tweepy.OAuthHandler(my_consumer_key, my_consumer_secret)
15 auth.set_access_token(my_access_token,my_access_secret)
16
17 my_keyword = "Trump"
18 n_tweets = 100
19 tweets = []
```

Line 8 ~ 11

- Authentication Tokens are available in the user twitter dashboard. It’s a good practice to saved them in an environment file (vars.env) or in a config file (config.py)

Reading Tweets from Twitter:

Ex Using the Tweepy library to fetch tweets: **API Twitter v1.1**

```
23 #Searching tweets
24 for status in tweepy.Cursor(api.search, q= my_keyword + " -filter:retweets",
25                              lang="en", result_type="recent").items(n_tweets):
26   tweets.append(status.text)
27
28 # create array of tweet information: username, tweet id, date/time, text
29 dataset=[]
30 for tweet in tweets:
31     tweet_data = {'user_name':tweet.user.screen_name, 'text':tweet.text}
32     dataset.append(tweet_data)
33
34
```

Line 24 ~ 26

- Cursor class can be used to iterate through the fetched tweets.

Reading Tweets from Twitter:

Ex A simple output:

Eric McCormack backtracks call to 'blacklist' Hollywood Trump donors https://t.co/r64ILsB5Mp https://t.co/5gCWP9aStI

If you don't understand people's pulse, Certainly politics is not for you.

Trump and Modi both are elected becaus… https://t.co/PQR7L4JJsz

@JohnBerman have you heard Trump's idea on how to get rid of Dorian/hurricanes in general? I hear when he suggested… https://t.co/Sde1ieROWO

"Dare We Dream of the End of the G.O.P.?" by BY MICHELLE GOLDBERG https://t.co/xf9ggrWEPc

Afghanistan President Ashraf Ghani to visit Washington to meet Donald Trump | 2019-09-06 https://t.co/dNJsyIbLI2

Times of Middle East: Ideal storm: Media pound Trump about ‘Sharpie-gate’ hurricane map https://t.co/Hwv6p6nwpP @DaPathanGuy @TimesofMEast

Times of Middle East: Trump’s Hurricane Strategy Tops This Week’s Internet Information Ro https://t.co/QX0yrghNDO @DaPathanGuy @TimesofMEast

Times of Middle East: Google’s Hellish 3 A long time, Trump’s Tariff Hold off, and Add https://t.co/M9xKV2ARIc @DaPathanGuy @TimesofMEast

Times of Middle East: Trump Aims to Privatize FANNIE and FREDDIE… https://t.co/OuBJp6STQ0 @DaPathanGuy @TimesofMEast

Trump shows old map with Alabama in Dorian’s forecast path https://t.co/Ihkq7yuax6

Times of Middle East: Trump Administration Expedites Obstacle to California on https://t.co/h03hooqRzQ @DaPathanGuy @TimesofMEast

Reading Tweets from Twitter:

https://www.tweepy.org/

Ex Using the **Tweepy** library to fetch tweets: **API Twitter v2 (Essential Users)**

```
1 !pip install colab_env
2 !pip install tweepy --upgrade
3
4 import os
5 import colab_env
6 import tweepy
7
8 #Calling API with app-only authentication
9 client = tweepy.Client(bearer_token=os.getenv("BEARER_TOKEN"))
10
11 #Searching tweets
12 #The query means: search the tweets with key word 'covid' thar are not retweets written in spanish
13 #By default it fetches 10 tweets. Some parameters:
14 #tweet_fields: to get additional fields, e.g. lang
15 #expansions: to expand the query with information of other objects related with the tweet, e.g. user
16 tweets = client.search_recent_tweets(query='covid -is:retweet lang:es',
17                                      tweet_fields=['lang'],
18                                      expansions='author_id')
```

02_search_tweets.ipynb

Line 9

- App-only authentication (Bearer Token) – read-only access to public information. Not require an authentication user.

Reading Tweets from Twitter:

Ex Using the **Tweepy** library to fetch tweets: **API Twitter v2 (Essential Users)**

```
19 i=1
20 for tweet in tweets.data:
21     #You can get the user information using the api
22     user = client.get_user(id=tweet.author_id)
23     print(str(i) + ' ' + str(tweet.author_id) + ' ' + user.data.username + ' ' + user.data.name)
24     i+=1
25     print(tweet.text + ' LANG: ' + tweet.lang +'\n')
26
27 # You also can get the user information expanding the query (expansions='author_id')
28 users = {u["id"]: u for u in tweets.includes['users']}
29
30 for tweet in tweets.data:
31     if users[tweet.author_id]:
32         user = users[tweet.author_id]
33         print(user.name)
34
```

Line 9

- App-only authentication (Bearer Token) – read-only access to public information. Not require an authentication user.

Reading Tweets from Twitter:

Ex A sample output

```
1 602566807 cesgaraleix César G. Aleixandre
Una fase más.... Vacunan con la cuarta dosis de covid al personal sanitario en la CV https://t.co/HOmZCGD8Tz
a través de @levante_emv

2 227070206 elorbe_ Periodico El Orbe
Retrocede temor popular al Covid-19 y la población ya no fue al "Mes del Testamento" + Durante los dos años
de pandemia fue el period... https://t.co/8dz6izPmqZ

3 1219007900582137856 skynexito Skynex Acho
Dios hay gente que sigue pensando en el covid no me lo puto explico

4 829348585 horanvoir maria
queria tomar vacina contra covid de novo

5 1380641162445385733 NoticiacriptoC Noticiacripto.com 🟢
AYER | China🇨🇳 comenzó a aplicar su política de "cero #COVID" con ametralladoras en el aeropuerto
Xishuangbanna en Yunnan. La gente gritaba "¿nos vas a matar a todos?" https://t.co/6tSLBFXYo6

6 1371366029138530305 embargenqatar Embajada Argentina en Qatar
📣 PROTOCOLO SANITARIO COVID-19 DE QATAR PARA EL MUNDIAL FIFA #Qatar2022 🇶🇦 @roadto2022es @fifaworldcup_es
@MOFAQatar_ES https://t.co/eTevspy5Ov

7 3062451 linker Linker Mortis
Nuevo COVID? el mundo esta loco, loco https://t.co/N7C4kIlU8s
```

Unit 1.

Text Mining

String Manipulation

Useful functions and methods for string manipulation:

Function	Explanation
x.lstrip()	Remove space on the left.
x.rstrip()	Remove space on the right.
x.strip()	Remove space on both sides.
x.replace(str1, str2)	Replace the substring str1 by str2.
x.count(str)	Number of occurrences of str in x.
x.find(str)	Find the sub-string str. Returns -1 if not found.
x.index(str)	Find the sub-string str. Throws an error if not found.
y.join(str_list)	Concatenate the elements of str_list using y as separator.
x.split(y)	Break up a string using y as separator.
x.upper()	Convert x into uppercase.
x.lower()	Convert x into lowercase.
len(x)	Returns the string length.

Useful functions and methods for string manipulation:

- Used for making string patterns.
- Useful for recognizing and processing complex string patterns ⇒ pre-processing of text data.
- More powerful than the combinations of the usual string functions or methods.
- Supported by many languages including Python.

Regular Expression

45. ex_0501.ipynb

- Metacharacters:
  - Characters with special meanings in the regular expressions.

```
. ^ $ * + ? { } [ ] \ | ( )
```

  - Can be used to construct patterns.
  - More information can be found at: https://docs.python.org/3/howto/regex.html

  - You can practice actual regular expressions in the website: https://regex101.com/

Metacharacters: [ ]

- Can enclose a set of characters as match pattern.
- Any character can be enclosed by the [ ].
- For example, "[abc]" matches with any pattern containing "a" or "b" or "c" character.

RegEx	String	Match?	Explanation
"[abc]"	"a"	Yes	"a" is in the string.
"[abc]"	"before"	Yes	"b" is in the string.
"[abc]"	"dude"	No	There is neither "a" nor "b" nor "c" in the string.

Metacharacters: [ ]

- We can use a hyphen "-" to indicate a range of characters.

  Ex "[0-5]" is the same as "[012345]"

  Ex "[0-9]" means the entire set of number digits.

  Ex "[a-d]" is the same as "[abcd]".

  Ex "[a-zA-Z]" means the entire set of alphabet letters both uppercase and lowercase.

Metacharacters: [^]

- Characters that are not in the enclosed set will be matched.
- " ^ " has to be the first character within the square brackets.

RegEx	String	Match?	Explanation
"[^abc]"	"a"	No	In the string, there is no other character than "a" or "b" or "c".
"[^abc]"	"before"	Yes	There are characters other than "a" or "b" or "c" in the string.
"[^abc]"	"dude"	Yes	There are characters other than "a" or "b" or "c" in the string.

Metacharacters: [ ] and [^]

- There are shorthand expressions as following:

  → "\w" is the same as "[a-zA-Z0-9_]"

  → "\W" is the same as "[^a-zA-Z0-9_]"

  → "\d" is the same as "[0-9]"

  → "\D" is the same as "[^0-9]"

  → "\s" means white space character.

  → "\S" means non-white space character.

Metacharacters: **Dot .**

- Dot matches with any character.
- "\." is the dot as a character (not a metacharacter).

RegEx	String	Match?	Explanation
"a.b"	"aab"	Yes	"a" in the middle of the string matches with the dot.
"a.b"	"a0b"	Yes	"0" in the middle of the string matches with the dot.
"a.b"	"abc"	No	There is no character in between "a" and "b".

Metacharacters: *

- Pattern that repeats the preceding character for any number of times (including 0).

RegEx	String	Match?	Explanation
"ca*t"	"ct"	Yes	"a" does not appear.
"ca*t"	"cat"	Yes	"a" appears once.
"ca*t"	"caaat"	Yes	"a" is repeated three times.

Metacharacters: +

- Pattern that repeats the preceding character at least once or more times.

RegEx	String	Match?	Explanation
"ca+t"	"ct"	No	"a" does not appear.
"ca+t"	"cat"	Yes	"a" appears once.
"ca+t"	"caaat"	Yes	"a" is repeated three times.

Metacharacters: ?

- Pattern where the preceding character does not appear or appears just once.

RegEx	String	Match?	Explanation
"ca?t"	"ct"	Yes	"a" does not appear.
"ca?t"	"cat"	Yes	"a" appears once.
"ca?t"	"caat"	No	"a" is repeated twice (more than once).

Metacharacters: {m}

- Pattern where the preceding character is repeated *m* times.

RegEx	String	Match?	Explanation
"ca{2}t"	"ct"	No	"a" does not repeat twice.
"ca{2}t"	"cat"	No	"a" does not repeat twice.
"ca{2}t"	"caat"	Yes	"a" is repeated exactly twice.

Metacharacters: {m, n}

- Pattern where the preceding character is repeated from *m* to *n* times.

RegEx	String	Match?	Explanation
"ca{2,5}t"	"cat"	No	"a" is repeated less than two times.
"ca{2,5}t"	"caat"	Yes	"a" is repeated twice.
"ca{2,5}t"	"caaaaaat"	No	"a" is repeated more than five times.

Metacharacters: ^

- Pattern after the ^ matches with the beginning of a string or text.
- Not the same meaning as the first hat character within the square brackets "[^]"

RegEx	String	Match?	Explanation
"^Life"	"Life is boring"	Yes	"Life" pattern is found at the beginning of the string.
"^Life"	"My Life is boring"	No	"Life" pattern is not found at the beginning of the string.

Metacharacters: $

- Pattern before the $ matches with the end of a string or text.

RegEx	String	Match?	Explanation
“Python$”	“Python is easy”	No	“Python” pattern is not found at the end of the string.
“Python$”	“You need Python”	Yes	“Python” pattern is found at the end of the string.

Metacharacters: |

- Used to join patterns by the logical or.
- More than two patterns can be concatenated by the logical or.

RegEx	String	Match?	Explanation
“love|hate”	“I love you”	Yes	“love” pattern found in the string.
“love|hate”	“I hate him”	Yes	“hate” pattern found in the string.
“love|hate”	“I like you”	No	Neither “love” nor “hate” pattern found in the string.

Matching group patterns:

match()	Determine if the RE matches at the beginning of the string.
search()	Scan through a string, looking for any location where this RE matches.
findall()	Find all substrings where the RE matches, and returns them as a list.
finditer()	Find all substrings where the RE matches, and returns them as an iterator.

match() and search() return None if no match can be found. If they're successful, a match object instance is returned, containing information about the match: where it starts and ends, the substring it matched, and more.

Replacing group patterns:

sub(pattern, repl, string, counts=0, flags=0)	Return the string obtained by replacing the leftmost non-overlapping occurrences of pattern in string by the replacement repl

Matching group patterns:

45. ex_0503.ipynb

- We can group patterns by enclosing with ( ).

Ex

```
1 import re
2 my_regex = re.compile("([0-9]+)[^0-9]+([0-9]+)")
3 m = my_regex.search("Anna is 15 years old and John is 12 years old.")
4 print("Group 0 (Full pattern): " + m.group(0))
5 print("Group 1: " + m.group(1))
6 print("Group 2: " + m.group(2))
```

```
Group 0 (Full pattern): 15 years old and John is 12
Group 1: 15
Group 2: 12
```

- In the example, an equivalent regular expression is: my_regex = re.compile("(\d+)\D+(\d+)")

```
1 import re
2 my_regex = re.compile("(\d+)\D+(\d+)")
3 m = my_regex.search("Anna is 15 years old and John is 12 years old.")
4 for i in range(0,len(m.groups())+1):
5   print("Group " + str(i) + ": " + m.group(i))
```

```
Group 0: 15 years old and John is 12
Group 1: 15
Group 2: 12
```

Matching group patterns:

- We can group patterns by enclosing with ( ).

Ex "Extract only the phone number"

```
1 import re
2 my_regex = re.compile("(\D+)((\d+)\D+(\d+)\D+(\d+))")
3 m = my_regex.search("John 010-1234-5678 Maria 023-4567-8900")
4 print("Phone number: " + m.group(2))
```

```
Phone number: 010-1234-5678
```

```
1 import re
2 my_regex = re.compile("(\D+)((\d+)\D+(\d+)\D+(\d+))")
3 m = my_regex.finditer("John 010-1234-5678 Maria 023-4567-8900")
4 for match in m:
5   print("Phone number of " + match.group(1).strip() + " is " + match.group(2).strip())
```

```
Phone number of John is 010-1234-5678
Phone number of Maria is 023-4567-8900
```

Matching group patterns:

- We can group patterns by enclosing with ( ).

Ex "Hilde de phone number"

```
1 # Hide the phone number.
2 reg_ex = re.compile("(\D+)(\d+)\D+(\d+)\D+(\d+)")
3 m = reg_ex.search("John Wheeler 010-1234-5678")
4 print((m.group(1)).strip() + " " + m.group(2) + "-****-****")
```

```
John Wheeler 010-****-****
```

```
1 # Hide the phone number with sub.
2 string= re.sub('\d','*',"John Wheeler 010-1234-5678")
3 print(string)
```

```
John Wheeler ***-****-****
```

Unit 1.

Text Mining

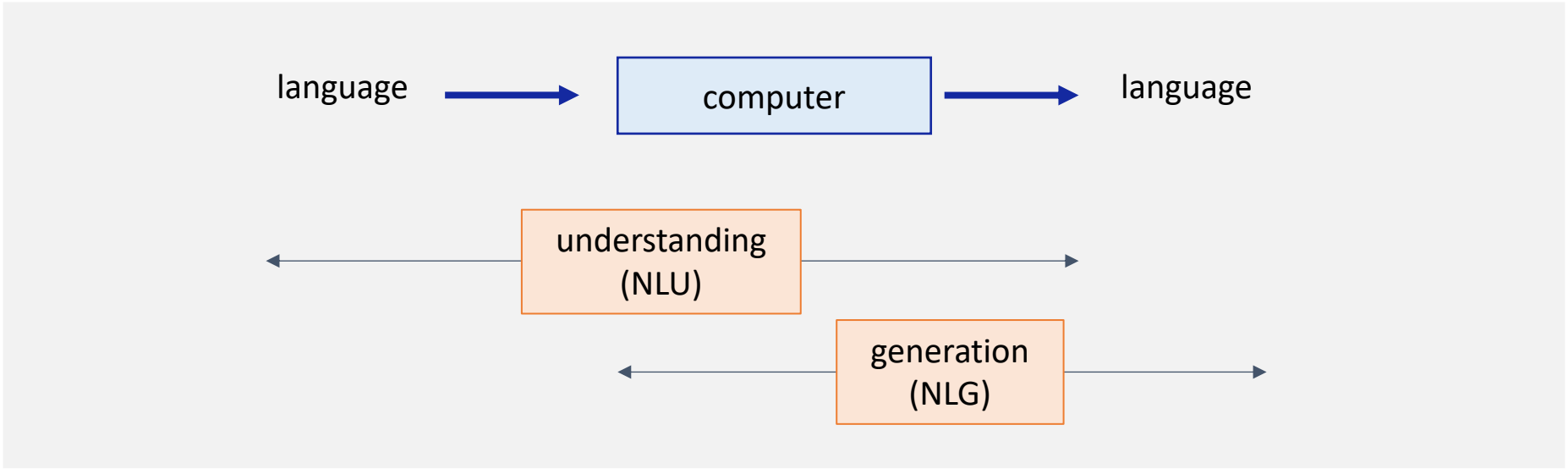
Natural Language Processing

What the Natural Language Processing (NLP) is:

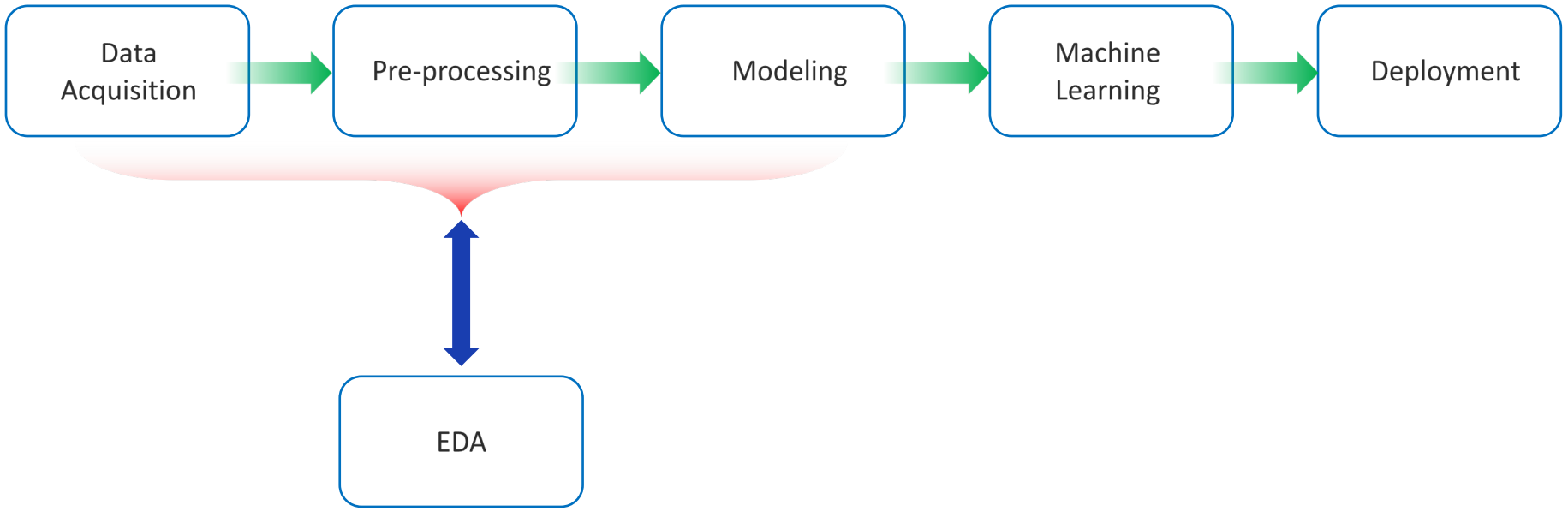
Natural language refers to language people naturally use in their daily lives.
Natural Language Processing (NLP) is an academic field that enables computer to understand and generate natural language.

- Extracts features from text data in order to classify, summarize, cluster and do sentiment analysis.
- Intersection of linguistics and AI
- Based on statistical models
- Different from the way humans understand the language
- Requires transformation into a structured model

A narrow meaning of Natural Language Processing is the processing mechanism used by programs using natural language as input and output.



NLP Workflow:



Difficulties of Natural Language Processing

- NLP is receiving much attention and is widely used, but the procedure is complex.
- The performance of machine translation has improved, but machine translation still often produces awkward translation results.
- NLP is complex because the input data is a not numerical value but human language. The input of human language makes the data processing extremely complex and uncertain.
    - Even the same words have possibility of various interpretations depending on context. This is called linguistic ambiguity.
    - Like idioms that take a different meaning once various words assemble, there is always an exception to how a phrase or words or morphemes construct a sentence.
    - Since language is flexible and open to expansion, modeling language always entails uncertainty. Also, as time goes by, new words are created, and some words become unused.

Difficulties of Natural Language Processing

Research paradigm of NLP: (1) Rule-based approach

- The rule-based approach defines beforehand the grammatical rules of the language, and processes natural language based on those rules.

  - It decides the part of speech (POS) for a given word based on linguistic phenomena rather than statistical methods.
  - In the sentence below, it grasps the meaning of the sentence with the first verb, and figures out the object of the instruction and its subject matter with 'to' or 'that.'

    "*Send a message to Michael that I will be late for meeting.*"

- The problem with the rule-based system is that it is impossible to establish the rules prior.
- Currently, it is only used within combination of other methods because grammatical rules cannot be entirely neglected in language processing.

Research paradigm of NLP: (2) Statistics-based approach

- The statistics-based approach decides the part of speech (POS) by computing lexical probabilities and contextual probabilities within reference to a huge volume of dictionaries to eliminate ambiguity of POS.

  - Lexical probability is probability that a certain POS applies to a word. This can be expressed as P (POS | word) in mathematical form.
  - Contextual possibility is possibility that a certain POS of a word will show with the POS of the next word. This, mathematically is P (POS | POS)
  - Then it labels the POS that produces highest result of the multiplication of linguistic probability and contextual probability as the most appropriate for words of semantic ambiguity

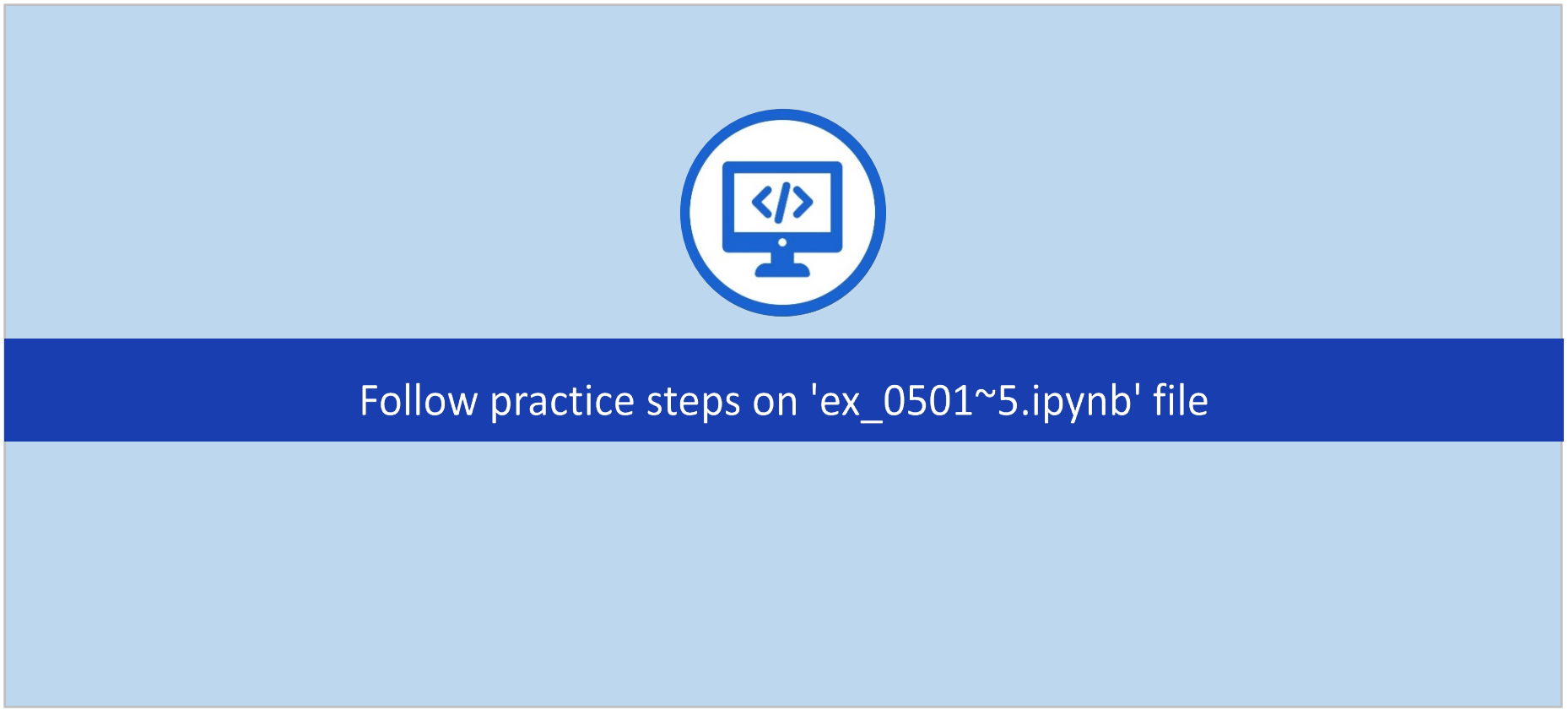
- There has been much progress as computers analyze sentences much faster with improvement in performance, but human intervention is still necessary.
- Such issues are being tackled by deep learning techniques nowadays.

Research paradigm of NLP: (3) Deep Learning-based approach

- While statistical analysis is based on peripheral analysis such as frequency of word appearance, Deep Learning approach enables in-depth analysis based on composite connection among data.

  - NLP models that understand sentence or overall context of sentence could have been created after deep learning based NLP was initiated.
  - If you create an artificial neural network component that connects with all parts of a sentence, this variable, after learning, contains information of the whole sentence. The accuracy is constantly increasing.

Coding Exercise #0501~0508



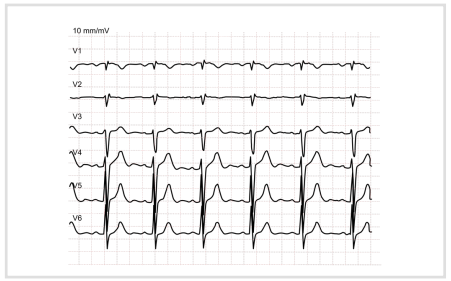
Unit 1.

Text Mining

Sequential Data

Data of temporal property is widely used around the world. (1/2)

- For example, there are stock prices, human voice, or ECG signals.
- These data have sequences. You must utilize this feature and temporal information in order to obtain high performance from sequential data.



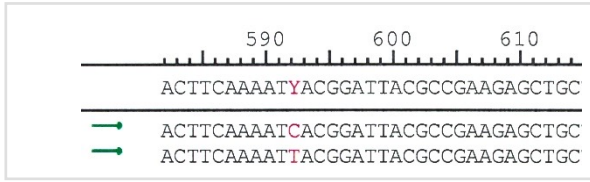
(a) ECG signal

(b) Stock price

(c) Vocal signal

Climbing down did I see, the flower I couldn't see on the way up

(d) sentence



(e) DNA sequence

Data of temporal property is widely used around the world. (2/2)

- Temporal data is dynamic because they change overtime, and it generally has variable length.
  - Recurrent neural network, which you will study later, is a learning model that effectively processes such temporal data.
- Lately, it can process extremely long patterns that occur in daily lives. For example, machine translator can now translate sentences of more than 30 words, while only 10 words was the maximum in the past.
- To translate a long sentence, it is needed to understand context between two words that are far apart. This is called long-term dependency.
  - Since standard RNN cannot fully process long-term dependency, LSTM, which supplemented selective memory function to standard RNN, is widely used. Selective memory is the capacity to discern memory for long-term and short-term.

Unit 1.

Text Mining

Corpus

Corpus:

- Refers to a set of text data subject to analysis.

  a) Raw corpus: text data stored in a data base.

  b) Tagged corpus: text data where words and phrases have been labeled according to a model.

```
( (S (`` ``)
    (S-TPC-2
      (NP-SBJ-1 (PRP We) )
      (VP (MD would)
        (VP (VB have)
          (S
            (NP-SBJ (-NONE- *-1) )
            (VP (TO to)
              (VP (VB wait)
                (SBAR-TMP (IN until)
                  (S
                    (NP-SBJ (PRP we) )
                    (VP (VBP have)
                      (VP (VBN collected)
                        (PP-CLR (IN on)
                          (NP (DT those) (NNS assets) )))))))))))
    (, ,) (`` ``)
    (NP-SBJ (PRP he) )
    (VP (VBD said)
      (S (-NONE- *T*-2) ))
    (. .) ))
```