

# 2018학년도 2학기 언어와 컴퓨터

## 제12강 정규표현식 실습

박수지

서울대학교 인문대학 언어학과

2018년 10월 31일 수요일

## 오늘의 목표

- 1 알고리즘을 읽고 파이썬으로 프로그램을 작성할 수 있다.
- 2 re 모듈을 사용하여 파이썬에서 정규표현식을 처리할 수 있다.
- 3 ELIZA를 파이썬에서 작동하게 할 수 있다.

# nltk

## Natural Language ToolKit

### 설치 방법

VSCode 터미널에서 실행할 것

**Windows** `pip install -U nltk`

**Mac OS** `sudo pip3 install -U nltk`

### 좋은 책

<https://www.nltk.org/book>

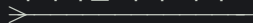
# nltk

## Natural Language ToolKit

### 예시: 단어 & 문장 분리

```
>>> import nltk
>>> sentence = 'The quick brown fox jumps over the lazy dog.'
>>> nltk.word_tokenize(sentence)
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the',
 'lazy', 'dog', '.']
>>> text = "Hello! How are you? I'm fine, thank you. And you?"
>>> nltk.sent_tokenize(text)
['Hello!', 'How are you?', "I'm fine, thank you.", 'And you?']
```

기계학습에서 계속...



# MaxMatch

## 목표

- 띄어쓰기가 없는 문장을 단어의 목록으로 만든다.
- 단어 하나하나의 길이는 가능한 한 길게 한다.  
**단어** 어휘부에 있는 것

## 예시

这些学生会游泳 [这些, 学生, 会, 学生会, 游泳]

- 1 这些/学生会/游泳 — MaxMatch의 선택
- 2 这些/学生/会/游泳 — 사실 이쪽이 잘된 단어 분리

# MaxMatch

- 1 문장이 빈 문자열이면 빈 목록을 반환한다.
- 2 문장의 첫 문자부터 시작하는 문자열을 긴 것부터 읽으면서:
  - a 현재 단계의 문자열이 사전에 있으면:
    - i 부분문자열을 단어 목록에 추가하고
    - i 나머지 부분에 대하여 [2]부터 다시 시작한다.
  - b 사전에 없으면:
    - ii 문자열의 길이를 줄여서 [2-a]로 되돌아간다.
- 3 가장 짧은 문자까지 읽었는데도 사전에 없다면:
  - a 첫 번째 문자를 단어 목록에 추가하고
  - b 두 번째 문자부터 시작하는 문자열에 대하여 [2]부터 다시 시작한다.

# MaxMatch

```
function MAXMATCH(sentence, dictionary) returns word sequence W

if sentence is empty
    return empty list
for  $i \leftarrow \text{length}(\text{sentence})$  downto 1
    firstword = first  $i$  chars of sentence
    remainder = rest of sentence
    if InDictionary(firstword, dictionary)
        return list(firstword, MaxMatch(remainder, dictionary) )

    # no word was found, so make a one-character word
    firstword = first char of sentence
    remainder = rest of sentence
    return list(firstword, MaxMatch(remainder, dictionary) )
```

**Figure 2.11** The MaxMatch algorithm for word segmentation.

pseudocode를 파이썬 소스코드로 번역하기

# MaxMatch

## pseudocode

```
if sentence is empty  
    return empty list
```

실습 코드 maxmatch.py

## Python

```
if not sentence:  
    return []
```

하지 마세요...

```
if sentence is "":  
    return []
```



# re 모듈 사용하기

## 시작: Raw string

```
>>> print('s\tu') # '\t'는 탭 문자
s    u
>>> print(r's\tu') # '\t'는 '\'와 't'의 연쇄
s\tu
>>> print(fr'\a{2+3}')
```

# f-string과 결합

```
\a5
```

# re 모듈 사용하기

## 오늘의 함수

`re.compile()` 문자열 → 정규표현식 패턴 객체

`re.match()` 패턴, 문자열 → 매치

`re.search()` 패턴, 문자열 → 매치

`re.findall()` 패턴, 문자열 → 매치된 문자들의 리스트

`re.sub()` 패턴, 대체할 것, 문자열 → 매치가 대체된 문자열

# re 모듈 사용하기

## 정규표현식 패턴 객체 만들기

### 전공 찾기

시작: [가-힣]+대학 [가-힣]+학과

- 인문대학 국어국문학과
- 사회과학대학 인류학과

수정: [가-힣]+대학 [가-힣]+학[과부]

- 자연과학대학 수리과학부

수정: [가-힣]+대학 [가-힣]+학?[과부]

- 음악대학 작곡과

수정: ([가-힣]+대학 )?[가-힣]+학?[과부]

- 자유전공학부

# re 모듈 사용하기

## 정규표현식 패턴 객체 만들기

### 전공 찾기

```
>>> pattern = re.compile(r'([가-힣]+대학 )?[가-힣]+학?[과부]')
>>> pattern
re.compile('([가-힣]+대학 )?[가-힣]+학?[과부]')
>>> type(pattern)
<class 're.Pattern'>
>>> dir(pattern)
['__class__', '__copy__', '__deepcopy__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'findall', 'finditer', 'flags', 'fullmatch', 'groupindex', 'groups', 'match', 'pattern', 'scanner', 'search', 'split', 'sub', 'subn']
```

# re 모듈 사용하기

## 문자열에서 패턴 검색하기

### 전공 찾기

```
>>> string = '서울대학교 자연과학대학 수리과학부 이한솔'
>>> print(pattern.search(string))
<re.Match object; span=(6, 18), match='자연과학대학 수리과학부'>
>>> print(re.search(pattern, string)) # 위와 같음
<re.Match object; span=(6, 18), match='자연과학대학 수리과학부'>
```

### 관찰

`help(re.match)`와 `help(re.search)`로 두 함수의 차이를  
알아보자.

# re 모듈 사용하기

## 문자열에서 패턴 검색하기

### 전공 찾기

```
>>> pattern
re.compile('([가-힣]+대학 )?[가-힣]+학?[과부]')
>>> match = pattern.search(string)
>>> print(match.group()) # 매치되는 부분 전체
자연과학대학 수리과학부
>>> print(match.group(1)) # 괄호 속에 있던 것
자연과학대학
```

### 관찰

`pattern = re.compile(r'(:[가-힣]+대학 )?[가-힣]+학?[과부]')`로 바꾼 뒤 다시 시도해 보자.

# re 모듈 사용하기

문자열에서 패턴과 매치되는 모든 것의 리스트 반환하기

## 신문명 찾기

```
>>> pattern = re.compile(r'([가-힣]+(?:일보|신문))')
>>> string = """조선일보가 ..... 덧붙였다."""
>>> print(pattern.search(string).group())
조선일보
>>> print(pattern.findall(string))
['조선일보', '조선일보', '경향신문', '경향신문', '조선일보', '경향신문', '문화일보', '문화일보', '조선일보', '조선일보']
```

# re 모듈 사용하기

패턴과 매치되는 부분을 다른 것으로 바꾸기

## 숫자 추상화

```
>>> string = '2018년 10월 31일 11시'
>>> print(string)
2018년 10월 31일 11시
>>> print(re.sub(r'[0-9]+', '#', string))
#년 #월 #일 #시
```



# re 모듈 사용하기

패턴과 매치되는 부분을 다른 것으로 바꾸기

## 지역번호 형식 바꾸기

```
>>> string = '02-880-2206'
>>> print(string)
02-880-2206
>>> print(re.sub(r'(0[02-9][0-9]*)-([0-9]+-[0-9]+)', \
... r'(\1)\2', string))
(02)880-2206
```

## 관찰

- 두 번째 인자의 \1과 \2가 무엇을 가리키는가?
- 두 번째 인자를 raw string으로 쓰지 않으면 어떻게 되는가?

# ELIZA

## 기원

Joseph Weizenbaum (1966)

“ELIZA: A Computer Program For the Study of **Natural Language Communication** Between Man And Machine”

- 《마이 페어 레이디》(1964)의 원작 《피그말리온》(1913)의 주인공  
⇒ 인간의 교육으로 대상의 언어 능력을 개선할 수 있다.

## 일라이자 효과

컴퓨터의 행동에 무의식적으로 인격을 부여하는 것

<http://www.hani.co.kr/arti/opinion/column/734697.html>

# ELIZA

## 작동 과정

- 1 입력된 텍스트를 읽고 키워드가 있는지 탐색한다.
- 2 키워드와 연관된 규칙에 따라 문장을 변형하여 출력한다.

## 예시

I am very unhappy these days

⇒ How long have you been very unhappy these days?

**키워드** “I am”

**분해 규칙** I am BLAH

**변형 규칙** How long have you been BLAH

BLAH의 의미를 이해할 필요가 없다.

# ELIZA

## 예시

It seems that you hate me

⇒ What makes you think i hate you

**키워드** “you”, “me”

**분해 규칙** 0 YOU 0 ME

0 임의의 개수의 단어 → 정규표현식 (.\*)

**변형 규칙** WHAT MAKES YOU THINK I 3 YOU

3 분해된 결과의 세 번째 요소 → 캡처 그룹 \1

# ELIZA

## 예시

```
>>> import re
>>> pattern = re.compile(r'.* YOU (.*) ME')
>>> message = 'IT SEEMS THAT YOU HATE ME'
>>> match = pattern.match(message)
>>> response = f'WHAT MAKES YOU THINK I {match.group(1)} YOU'
>>> print(response)
WHAT MAKES YOU THINK I HATE YOU
```

# ELIZA 만들기

실습 코드 `eliza.py`

## 실행 예시

```
PLEASE TELL ME YOUR PROBLEM
> Femmes are all alike.
IN WHAT WAY
> They are always bugging us about something or other.
CAN YOU THINK OF A SPECIFIC EXAMPLE
> Well, my girlfriend made me come here.
YOUR GIRLFRIEND MADE YOU COME HERE.
> She says I'm depressed much of the time.
I AM SORRY TO HEAR THAT YOU ARE DEPRESSED
```

더 생각해 볼 것

`re.split()` 함수에 대하여 알아보자.

다음 주에 배울 것

데이터 시각화, 벡터, 조건부확률, 베이즈 정리, 회귀분석 등등

참고

《밑바닥부터 시작하는 데이터 과학》 3-8장, 11장