정규표현식 보충자료

문제 2-3 (근처 카페, 맛집, 주변 식당) 등 궁금한 부분 추출하기

문제 2-3에서 찾고자 하는 부분은 문제 2-1과 문제 2-2에서 추출한 정보들 사이에 있습니다.

예를 들면, '홍대역 카페 어디가 괜찮아?' 라는 문장에서 우리가 추출하고자 하는 정보는 '카페' 입니다.

그리고 '카페' 라는 정보는 '홍대역' 과 '어디가 괜찮아?' 사이에 존재합니다.

다른 예시를 들어봅시다. '신촌역 괜찮은 식당 어떤 거 있니?' 라는 문장에서 우리가 원하는 정보는 '괜찮은 식당' 이죠. 그리고 이 정보는 '신촌역' 과 '어떤 거 있니?' 사이에 존재합니다.

문제 2-1 의 정보들은 '-역' 과 같은 형태로 끝나고,

문제 2-2 의 정보들은 '어떤, 어디, 좀' 의 세 가지 경우 중 하나로 시작한다는 점을 잘 활용하면

아래와 같은 정규 표현식을 세울 수 있습니다.

```
p1 = re.compile("역 (.+?) 어떤")
p2 = re.compile("역 (.+?) 어디")
p3 = re.compile("역 (.+?) 좀")

result = p1.findall(text) + p2.findall(text) + p3.findall(text)
print(result)
# ['괜찮은 식당', '주변 식당', '근처에 카페', '카페', '맛집', '카페', '근처 카페', '맛집', '주변 식당', '유명한 맛집', '주변 음식점', '주변 카페']
```

만약 findall 함수를 사용하면서 '|' 기호를 사용하면 다음과 같은 결과를 얻을 수 있습니다.

```
p = re.compile("역 (.+?) 어떤|역 (.+?) 어디|역 (.+?) 좀")
result = p.findall(text)
print(result)

# [('', '', '근처 카페'),
#  ('', '', '맛집'),
#  ('', '', '주변 식당'),
#  ('', '', '유명한 맛집'),
#  ('괜찮은 식당', '', ''),
#  ('', '카페', ''),
#  ('주변 식당', '', ''),
#  ('근처에 카페', '', ''),
#  ('', '맛집', ''),
#  ('', '카페', ''),
#  ('', '', '주변 음식점'),
#  ('', '', '주변 카페')]
```

아무래도 원했던 결과와는 약간 다른 결과가 도출된 모습입니다.

이처럼 findall 함수에 grouping 과 or 을 함께 적용하다보면 결과가 복잡해질 수 있어서 상당한 주의를 요합니다.

그래서 첫 번째 방식처럼 아예 정규표현을 나누거나 전후방탐색 방식을 이용한다면 보다 직관적으로 정보를 추출할 수 있습니다. 전방탐색을 이용한 코드는 다음과 같습니다.

```
p = re.compile("역 (.+) (?=어떤|어디|좀)")
result = p.findall(text)
print(result)

# ['근처 카페', '맛집', '주변 식당', '유명한 맛집', '괜찮은 식당', '카페',
# '주변 식당', '근처에 카페', '맛집', '카페', '주변 음식점', '주변 카페']
```

문제 2-4 질문 바꾸기

```
#역 이름 바꾸기

p = re.compile(".+역")
result = p.sub('안암역',text)
print(result)
```

문제 2-1 에서 사용했던 정규표현식을 활용합니다.

여기선 편의상 “.+역" 표현을 사용했습니다.

이후에 re.sub 함수를 통해 역명을 바꿔줍니다.

```
#질문 바꾸기

p = re.compile("((어떤|어디|좀).+)")
result = p.sub('알려주세요',text)
print(result)
```

마찬가지로 문제 2-2에서 사용했던 정규표현식을 활용합니다.

“어떤.+|어디.+|좀.+” 을 사용해도 좋고, “(어떤|어디|좀).+” 도 좋습니다.

```
#질문 바꾸기2

# 답1
p = re.compile("역 (.+) (?=어떤|어디|좀)")
result = p.sub('역 레스토랑 ',text)
print(result)
print()

# 답2
p = re.compile("(?<=역 )(.+)(?= 어떤| 어디| 좀)")
result = p.sub('레스토랑',text)
print(result)
```

앞선 문제에서 활용된 전방표현을 활용하면 이 문제도 쉽게 해결할 수 있습니다.

동일한 정규표현식을 사용하고, 해당 표현으로 매치된 내용을 ‘역 레스토랑 ‘으로 바꿔줌으로써

비슷한 질문들을 손쉽게 만들어 낼 수 있습니다. (#답1)

‘역 ‘ 부분을 후방탐색을 통해 나타낼 수도 있습니다. (#답2)

문제2-5

```
#역 이름 조합

p = re.compile(".+역")
station = set(p.findall(text))

for one in station:
  print(one,'--------')

  result = p.sub(one,text)
  print(result)
```

이제 역 이름을 모으는 과정은 익숙하리라 생각합니다.

“.+역" 패턴과 re.findall 함수를 활용해 역들을 전부 찾은 후, set 함수를 통해 중복을 제거해줍니다.

그렇게 만들어 낸 역들의 집합에 대해 반복문을 돌며 역명을 바꿔줍니다.

물론 내용을 바꿀 때 쓰는 함수는 re.sub 함수입니다.

```
#다양한 조합으로
p = re.compile("어떤.+|어디.+|좀.+")
result = set(p.findall(text))

for one in result:
  print(one,'--------')
  answer = p.sub(one,text)
  print(answer)
```

직전에는 역이름만 바꿔주었다면 이제는 이어지는 질문 부분을 바꿔주는 내용입니다.

패턴 부분만 다르고 나머지는 동일합니다.

```
#다양한 조합으로
p = re.compile("(?<=역 )(.+)(?= 어떤| 어디| 좀)")
result = p.findall(text)
print(result)
print()

for one in result:
  print(one,'--------')
  result = p.sub(one,text)
  print(result)
```

마지막으로 역이름과 질문 부분 사이에 들어가는 내용을 바꿔볼 수도 있습니다.

마찬가지로 패턴 부분만 다르고 나머지는 동일합니다.