Simple Search Engine

231135 이나경

1. Introduction

- Project Purpos and Background: This project was undertaken to apply the knowledge in the seven weeks. The objective was to pracitce practical implementation based on the lessons covered. Practice on how to handle files and sentences

- Goal: To develop a basic search engine that retrieves sentences similar to the user's query.

2. Requirements

- User requirements: The system should be capable of searching for sentences similar to the user's query.

- Funtional Requirments: (Same as previously listed.) preprocessing, indexing, calc_similarity, ranking, output Top 10

3. Design and Implementation

1. preprocessing

```
def preprocess(sentence):
    preprocessed_sentence=sentence.strip().split(" ")
    return preprocessed_sentence
```

- input: sentence: Sentence you want to preprocess

- result: return value: preprocessed_sentence.

Spaces are removed and sentences divided based on spacing are given.

- Explanation: Use strip() to remove spaces on both sides of the entered sentence and Divide sentences using spaces using split()

2. indexing

```
# 인덱싱을 진행합니다.
def indexing(file_name):
    file_tokens_pairs=[]
    # file의 문장들 전부를 읽기 형식으로 가져옵니다.
    lines=open(file_name,"r",encoding="utf8").readlines()
    for line in lines:
        tokens=preprocess(line) # 문장 line에 대해 전처리 진행
        file_tokens_pairs.append(tokens)
    return file_tokens_pairs
```

- input: file_name = It is jhe-koen-dev.en.txt in this case. file_tokens_pairs = List for storing tokens

- result: return value: file_tokens_pairs

A set of tokens for each sentence in a file

- Explanation: Declare lsit, import sentences in read format. Take the sentences one by one, preprocess them, and add the result to file_tokens_pairs.

3. calc_similarity

```
def calc_similarity(preprocessed_query,preprocessed_sentences)
    # preprocessed_query = query_token_set, preprocessed_sente
    score_dict={}
    for i in range(len(preprocessed_sentences)): # 파일의 문장
        sentences_set = set(preprocessed_sentences[i]) # set으
        sentence = preprocessed_sentences[i] # 원형 저장
        query_str = ' '.join(preprocessed_query).lower() # 대소
        sentence_str = ' '.join(sentence).lower()
        preprocessed_query = set(preprocess(query_str)) # 전처
        preprocessed_sentence = preprocess(sentence_str)

        file_token_set = set(preprocessed_sentence)
        all_tokens = preprocessed_query | file_token_set
        same_tokens = preprocessed_query & file_token_set
        # 같은 tokens의 수와 모든 토큰을 비교하여 유사도 측정
        similarity = len(same_tokens) / len(all_tokens)
        score_dict[i] = similarity
    return score_dict
```

- input: preprocessed_query = query_token_set, preprocessed_sentences = file_tokens_pairs

- result: return value: score_dict

Returns the result of measuring the similarity of all tokens.

- Explanation: Declares a dictionary score_dict to store the measured similarity. Repeat as much as preprocessed_sentences to measure the similarity of sentences.

Save sentence_set as a set and use lower() to change it into lowercase letters so that it can be compared without case-sensitive cases.

Measure the similarity by comparing the number of the same tokens with the number of all tokens, and store the results.

4. ranking

```
# 4. 유사도 리스트를 정렬합니다.
sorted_score_list = sorted(score_dict.items(), key = operator.itemgetter(1), reverse=True)
```

- input: score_dict = the result of measuring the similarity of all tokens.

- result: List of sorted similarities

- Explanation: Sort the items in the dictionary.

5. output Top 10

```
# 5. 결과를 출력합니다.
if sorted_score_list[0][1] == 0.0:
    print("There is no similar sentence.")
else:
    print("rank", "Index", "score", "sentence", sep = "₩t")
    rank = 1
    for i, score  in sorted_score_list:
        print(rank, i, score, ' '.join(file_tokens_pairs[i]), sep = "₩t")
        if rank == 10:
            break
        rank = rank + 1
```

- input: sorted_score_list = Sorted Similarity Figures

- result: 10 highest similarity results output

- Explanation: Repeating sorted_score_list, outputting information and adding 1 to rank each time output, ending the program when rank turns 10.

4. Testing

1. preprocessing

```
['Some', 'years', 'later,', 'the', 'real', 'murderer', 'was', 'discovered.']
['Many', 'of', 'the', "world's", 'priceless', 'artworks', 'have', 'been', 'dama
ged', 'or', 'destroyed', 'by', 'warfare.']
['In', 'Vermont,', 'someone', 'had', 'to', 'walk', 'with', 'a', 'red', 'flag',
'to', 'warn', 'that', 'a', 'car', 'was', 'coming.']
['He', 'was', 'surprised', 'to', 'find', 'so', 'few', 'people', 'in', 'the', 's
treet,', 'but', 'thought', 'that', 'this', 'was', 'because', 'he', 'was', 'so',
'much', 'earlier', 'than', 'usual.']
['So', 'small', 'talk', 'helps', 'people', 'decide', 'if', 'they', 'want', 't
o', 'get', 'to', 'know', 'each', 'other', 'better.']
['"How', 'did', 'you', 'manage', 'to', 'become', 'so', 'rich', 'without', 'e-ma
il', 'and', 'e-commerce?']
['When', 'you', 'look', 'in', 'the', 'mirror,', 'what', 'do', 'you', 'see?']
['One', 'group', 'thought', 'of', 'a', 'relaxing', 'scene', 'like', 'a', 'blu
e', 'sky', 'or', 'a', 'beach.']
['This', 'method', 'is', 'called', '*acupuncture.']
['Strong', 'believers', 'may', 'not', 'even', 'swallow', 'their', 'own', 'saliv
a.']
['Wild', 'animals', 'go', 'to', 'sleep', 'and', 'stay', 'asleep', 'all', 'winte
```

2. indexing

```
r', 'apron', 'strings,', 'but', 'sooner', 'or', 'later', 'their', 'society',
ill', 'catch', 'up', 'with', 'the', 'progressive', 'world.']
['Do', 'you', 'know', 'what', 'the', 'cow', 'answered?"', 'said', 'the', 'mi
ter.']
['Poland', 'and', 'Italy', 'may', 'seem', 'like', 'very', 'different', 'coun
es.']
['Mr.', 'Smith', 'and', 'I', 'stayed', 'the', 'whole', 'day', 'in', 'Oxford.
['The', 'sight', 'of', 'a', 'red', 'traffic', 'signal', 'gave', 'him', 'an',
dea.']
['So', 'they', 'used', 'pumpkins', 'instead.']
['2.', 'a', 'particular', 'occasion', 'of', 'state', 'of', 'affairs:', 'They
'might', 'not', 'offer', 'me', 'much', 'money.']
["I'm", 'especially', 'interested', 'in', 'learning,', 'horse-riding', 'skil
s,', 'so', 'I', 'hope', "you'll", 'include', 'information', 'about', 'this.']
['Instead,', 'the', 'devil', 'gave', 'him', 'a', 'single', 'candle', 'to', '
ht', 'his', 'way', 'through', 'the', 'darkness.']
['It', 'shines', 'over', 'the', 'sea.']
```

3. calc_similarity

```
0.0
0.0
0.0
0.0
0.0
0.0
0.1
0.0
0.0
0.08333333333333333
0.043478260869565216
0.0
0.0625
0.058823529411764705
0.0
0.0
0.06666666666666667
```

4. ranking

```
{0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0, 5: 0.0, 6: 0.0, 7: 0.0, 8: 0.0, 9: 0.
0, 10: 0.0, 11: 0.0, 12: 0.0, 13: 0.0, 14: 0.0, 15: 0.0, 16: 0.0, 17: 0.1, 18:
0.0, 19: 0.0, 20: 0.08333333333333333, 21: 0.043478260869565216, 22: 0.0, 23:
0.0625, 24: 0.058823529411764705, 25: 0.0, 26: 0.0, 27: 0.06666666666666667, 2
8: 0.0, 29: 0.0, 30: 0.0, 31: 0.1111111111111111, 32: 0.0, 33: 0.0, 34: 0.0, 3
5: 0.0, 36: 0.0, 37: 0.0, 38: 0.0, 39: 0.0, 40: 0.0, 41: 0.0, 42: 0.0, 43: 0.0,
44: 0.0, 45: 0.125, 46: 0.0, 47: 0.0, 48: 0.0, 49: 0.0, 50: 0.0909090909090909
1, 51: 0.09090909090909091, 52: 0.0, 53: 0.0, 54: 0.0, 55: 0.05555555555555555,
56: 0.0, 57: 0.0, 58: 0.0, 59: 0.0, 60: 0.0, 61: 0.0, 62: 0.0, 63: 0.0, 64: 0.
0, 65: 0.05555555555555555, 66: 0.0, 67: 0.0, 68: 0.0, 69: 0.037037037037037035
5, 70: 0.07142857142857142, 71: 0.0, 72: 0.0, 73: 0.08333333333333333, 74: 0.0,
75: 0.07142857142857142, 76: 0.0, 77: 0.09090909090909091, 78: 0.08333333333333
333, 79: 0.0, 80: 0.05, 81: 0.0, 82: 0.07142857142857142, 83: 0.0, 84: 0.0, 85:
0.0, 86: 0.0, 87: 0.0, 88: 0.0, 89: 0.0, 90: 0.0, 91: 0.0, 92: 0.0, 93: 0.0, 9
4: 0.0, 95: 0.0, 96: 0.05263157894736842, 97: 0.0, 98: 0.043478260869565216, 9
9: 0.0, 100: 0.0, 101: 0.0, 102: 0.0, 103: 0.0, 104: 0.0, 105: 0.0, 106: 0.0526
31578947368421, 107: 0.125, 108: 0.0, 109: 0.0, 110: 0.0, 111: 0.07692307692307
693, 112: 0.07692307692307693, 113: 0.0, 114: 0.0, 115: 0.0, 116: 0.0, 117: 0.0,
```

5. output Top 10

rank	index	score	sentence
1	679	0.6	My name is Mike.
2	526	0.3333333333333333	Bob is my brother.
3	538	0.3333333333333333	My hobby is traveling.
4	453	0.2857142857142857	My mother is sketching them.
5	241	0.25	My father is running with So-ra.
6	336	0.25	My family is at the park.
7	212	0.2222222222222222	My sister Betty is waiting for me.
8	505	0.2	My little sister Annie is five years old.
9	190	0.16666666666666666	It is Sunday.
10	314	0.16666666666666666	This is Washington.

- Final Test Screenshot

1) If there is no similar sentence

```
영어 쿼리를 입력하세요.Hello
There is no similar sentence.
```

2) If there is a similar sentence

```
영어 쿼리를 입력하세요.Hello My name is
```

rank	Index	score	sentence
1	679	0.6	My name is Mike.
2	526	0.3333333333333333	Bob is my brother.
3	538	0.3333333333333333	My hobby is traveling.
4	453	0.2857142857142857	My mother is sketching them.
5	241	0.25	My father is running with So-ra.
6	336	0.25	My family is at the park.
7	212	0.2222222222222222	My sister Betty is waiting for me.
8	505	0.2	My little sister Annie is five years old.
9	190	0.16666666666666666	It is Sunday.
10	314	0.16666666666666666	This is Washington.

5. Results and conclusions

1. Project Results: Created a simple search engine.

2. What I felt: It was amazing to make a small search engine, and it was very hard, but I felt like I had a lot to gain.