Search engine project report

204201 김리하

1. Introduction

I created a simple search engine based on what I learned in Python class.

2. Requirements

A search system that outputs sentences in order of high similarity when the user enters a query.

3. Design and Implementation

```
# 전처리 함수
def preprocess(sentence): # 주어진 문장을 리스트로 저장
    preprocessed_sentence = sentence.strip().split(" ")
    return preprocessed_sentence
```

Result: Create and return a list of words

Description: A function that preprocesses a sentence and creates a list of words separated by spaces.

```
# 인덱싱
def indexing(file_name): # 주어진 파일에서 각 라인을 읽어와서 전처리된 토큰 리스트로 저장하는 리스트를 생성
    file_tokens_pairs = []
    lines = open(file_name, "r", encoding="utf8").readlines()
    for line in lines:
        tokens = preprocess(line)
        file_tokens_pairs.append(tokens)
    return file_tokens_pairs
```

Input: The file where the line was entered.

Result: Save each line to a list

Description: Preprocess each line and store it in the token list.

Save this list back to the list and return it.

```
    # 유사도 측정 함수
def calc_similarity(preprocessed_query, preprocessed_sentences):
    score_dict = {}
    for i in range(len(preprocessed_sentences)):    # 각 문장 유사도를 계산
        # 시작: 대소문자 구분 없는 토큰 셋을 만들기 위한 코드
        sentence = preprocessed_sentences[i]
        query_str = ' '.join(preprocessed_query).lower()
        sentence_str = ' '.join(sentence).lower()
        preprocessed_query = set(preprocess(query_str))
        preprocessed_sentence = preprocess(sentence_str)
        # 끝: 대소문자 구분 없는 토큰 셋을 만들기 위한 코드

        file_token_set = set(preprocessed_sentence)
        all_tokens = preprocessed_query | file_token_set
        same_tokens = preprocessed_query & file_token_set
        # 유사도를 계산하고 딕셔너리에 저장
        similarity = len(same_tokens) / len(all_tokens)
        score_dict[i] = similarity

    return score_dict  # 딕셔너리 반환
```

Input: preprocessed_query, preprocessed_setences = preprocessed list

Result: Calculate similarity and store in dictionary

Description: Calculates the similarity between queries and sentences, stores the results in a dictionary, and returns them.

```
# 5. 결과 출력
if sorted_score_list[0][1] == 0.0:
    print("There is no similar sentence.")
else:
    print("rank", "Index", "score", "sentence", sep = "₩t")
    rank = 1

    # 최대 10위까지 출력
    for i, score  in sorted_score_list:
        print(rank, i, score, ' '.join(file_tokens_pairs[i]), sep = "₩t")
        if rank == 10:
            break

        # 0점인 결과 출력 X
        if sorted_score_list[i][1] == 0.0:
            break

        rank = rank + 1
```

Input: sorted_score_list – A list where the similarity scores for each sentence are sorted and stored.

Result: Output sentences and scores with similarity up to the top 10

If the score is 0, it is not output.

If there is no similar sentence, output as none.

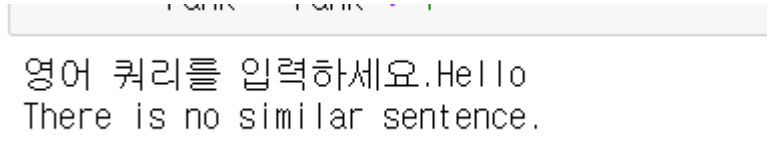
Description: If the score of the first item in the sorted list is 0, it is determined that there are no similar sentences and is not output.

Otherwise, the ranking and number sentences of sentences other than 0 points from 1st place up to 10th place are output.

4. Test

```
# 인덱싱
def indexing(file_name): # 주어진 파일에서 각 라인을 읽어와서 전처리된 토큰 리스트로 저장하는 리스트를 생성
    file_tokens_pairs = []
    lines = open(file_name, "r", encoding="utf8").readlines() # 파일을 읽어서 각 라인을 리스트로 저장
    for line in lines:
        tokens = preprocess(line)
        file_tokens_pairs.append(tokens)
        print(tokens)
    return file_tokens_pairs  # 전처리된 토큰 리스트로 이루어진 리스트를 반환
```

```
['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', 'skills,', 'so', 'I', 'hope', "you'll", 'include', 'information', 'about', 'this.']
['Instead,', 'the', 'devil', 'gave', 'him', 'a', 'single', 'candle', 'to', 'light', 'his', 'way', 'through', 'the', 'darkness.']
['It', 'shines', 'over', 'the', 'sea.']
['He,', 'too,', 'was', 'arrested,', 'and', 'a', 'bomb', 'was', 'thrown', 'at', 'his', 'house.']
['It', 'seems', 'that', 'the', 'high', 'temperature', 'and', 'pressure', 'on', 'the', 'star', 'made', 'its', 'carbon', 'surface', 'turn', 'to', 'diamond.']
['"The', 'pig', 'was', 'unpopular', 'while', 'the', 'cow', 'was', 'loved', 'by', 'everyone.']
['Books', 'give', 'a', 'lot', 'of', 'things', 'to', 'us.']
['Jimmy', 'and', 'Timmy', 'were', 'identical', 'twins.']
['It', 'is', 'a', 'chemical', 'that', 'cause', 'cancer.']
['Ziege', 'from', 'Germany', 'and', 'Brazilian', 'superstars', 'Ronaldo', 'and', 'Roberto', 'Carlos', 'belonged', 'to', 'the', 'bald', 'club.']
['Now', 'the', 'Taliban', 'are', 'gone', 'and', 'things', 'have', 'begun', 'to', 'change.']
['Is', 'your', 'skin', 'clear,', 'smooth,', 'and', 'shining', 'with', 'health?']
['As', 'a', 'result,', 'there', 'is', 'a', 'great', 'deal', 'of', 'traffic', 'and', 'usually', 'not', 'enough', 'roads,', 'and', 'most', 'of', 'the', 'roads', 'are', 'too', 'narrow.']
['The', 'new', 'law', 'said', 'that', 'blacks', 'were', 'free', 'to', 'sit', 'anywhere', 'on', "Montgomery's", 'buses.']
```




```
영어 쿼리를 입력하세요.good
rank    Index   score   sentence
1       473     0.1111111111111111      They said that Tom was a very good actor.
2       175     0.09090909090909091     He thought that the two Cs would look good in advertising.
3       255     0.08333333333333333     Children's minds have more information than before, but is more information good for them?
4       259     0.08333333333333333     One child answered, "My dream is to go to a very good school."
5       539     0.07142857142857142     He talked to the class for half an hour about the importance of good behavior.
6       322     0.0625  Some doctors in London say a little humor each day can be good medicine for depression.
7       603     0.058823529411764705    It is also a time to study the Quran and do good deeds such as helping the poor.
8       513     0.05555555555555555     Anonymity in cyberworld can certainly be a good thing in some ways, but if it is misused, it can be very offensive.
9       714     0.05    In some ancient European religions, there were 12 good gods and one evil god; the evil god was called the 13th god.
```

5. Results and Conclusion

Result: You have completed a simple search engine.

What I felt: It was difficult because there were many parts that were difficult to understand and there were frequent errors. I realized my shortcomings.