1. Introduction

① Project Purpose and Background: This project was undertaken to apply the knowledge learned in the seven weeks. The objective was to practice practical implementation based on the lessons covered.

② Goal: To develop a basic search engine that retrieves sentences similar to the user's query.

2. Requirements

① Preprocess sentences within the search target and store them in a list.

② Receive an input English string (query) from the user and preprocess it

③ Calculate the similarity between the query and sentences within the search target(Similarity is based on the count of the same “word.")

④ Rank the sentences based on similarity

⑤ Output the top 10 ranked sentences to the user from the ranked sentences

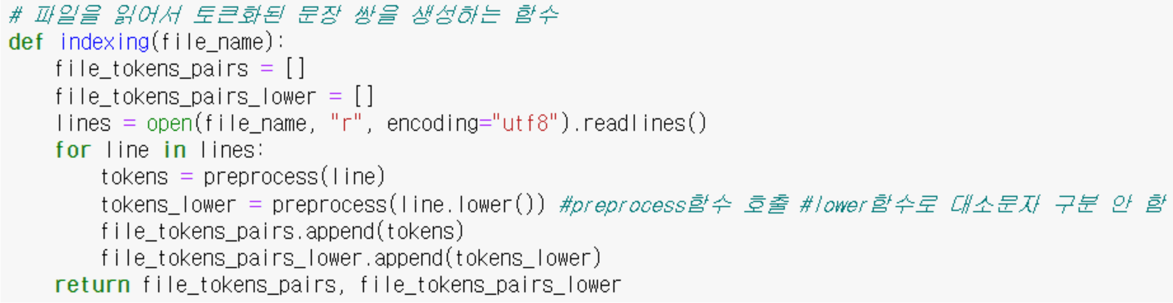
3. Design and Implementation

Implementation Details: (Detailed explanation of how each requirement was implemented, including relevant code snippets)

① User requirements: The system should be capable of searching for sentences similar to the user's query.

② Functional Requirements:

- Preprocess sentences within the search target and store them in a list.



```
# 파일을 읽어서 토큰화된 문장 쌍을 생성하는 함수
def indexing(file_name):
    file_tokens_pairs = []
    file_tokens_pairs_lower = []
    lines = open(file_name, "r", encoding="utf8").readlines()
    for line in lines:
        tokens = preprocess(line)
        tokens_lower = preprocess(line.lower()) #preprocess함수 호출 #lower함수로 대소문자 구분 안 함
        file_tokens_pairs.append(tokens)
        file_tokens_pairs_lower.append(tokens_lower)
    return file_tokens_pairs, file_tokens_pairs_lower
```

  - ■ Input:

    - · input: a file name with its path for search targets

  - ■ Result:

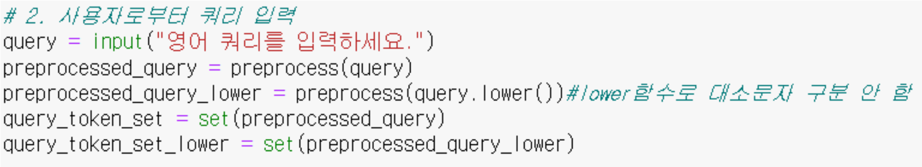
        - return value: a set of tokens for each sentence in the file and A set of lowercase-only tokens for each sentence in a file

    - ■ Explanation:

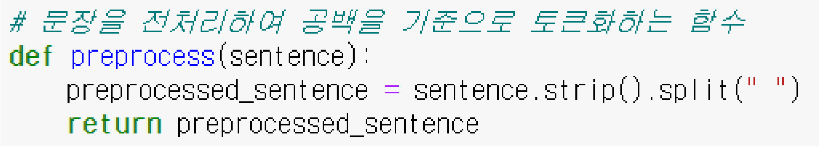
        - Read files in the path as a function “readlines”

        - Save the read file to the variables "file_tokens_pairs" and "file_tokens_pairs_lower" as a recurring statement (lower is lowercase)

- ● Receive an input English string (query) from the user and preprocess it.



```
# 2. 사용자로부터 쿼리 입력
query = input("영어 쿼리를 입력하세요.")
preprocessed_query = preprocess(query)
preprocessed_query_lower = preprocess(query.lower())#lower함수로 대소문자 구분 안 함
query_token_set = set(preprocessed_query)
query_token_set_lower = set(preprocessed_query_lower)
```



```
# 문장을 전처리하여 공백을 기준으로 토큰화하는 함수
def preprocess(sentence):
    preprocessed_sentence = sentence.strip().split(" ")
    return preprocessed_sentence
```

    - ■ Input: Value entered by the user

    - ■ Result: Functions that preprocess sentences to tokenize them based on spaces

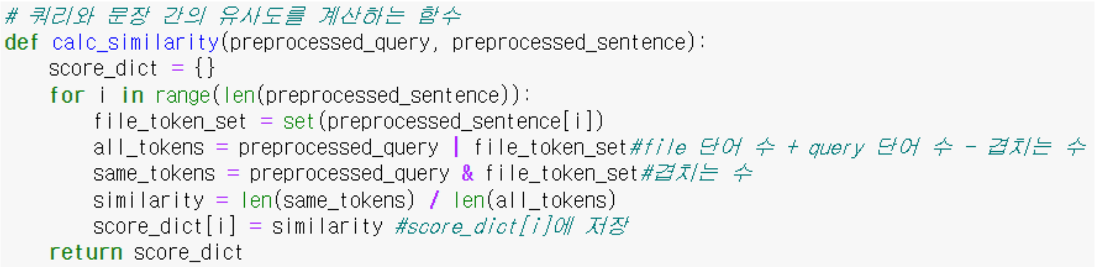
    - ■ Explanation:

        - ◆ Save user input to variable "query"

        - ◆ Call function "preprocess"

        - ◆ Save preprocessed queries in lowercase and normal cases, respectively

- ● Calculate the similarity between the query and sentences within the search target(Similarity is based on the count of the same “word.")



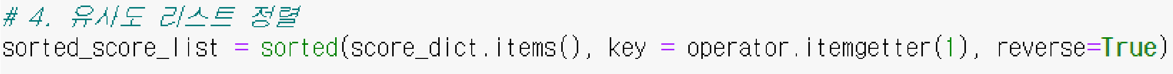
```
# 쿼리와 문장 간의 유사도를 계산하는 함수
def calc_similarity(preprocessed_query, preprocessed_sentence):
    score_dict = {}
    for i in range(len(preprocessed_sentence)):
        file_token_set = set(preprocessed_sentence[i])
        all_tokens = preprocessed_query | file_token_set#file 단어 수 + query 단어 수 - 겹치는 수
        same_tokens = preprocessed_query & file_token_set#겹치는 수
        similarity = len(same_tokens) / len(all_tokens)
        score_dict[i] = similarity #score_dict[i]에 저장
    return score_dict
```

    - ■ Input:

        - ◆ Preprocessed search queries in lower case and preprocessed search target statements in lower case

    - Result:
        - Dictionary with file ID and similarity scores for queries and sentences
    - Explanation:
        - all_token = Number of word in file + Number of word in query - Overlapping Count
        - same_token = Overlapping Count

- Rank the sentences based on similarity



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

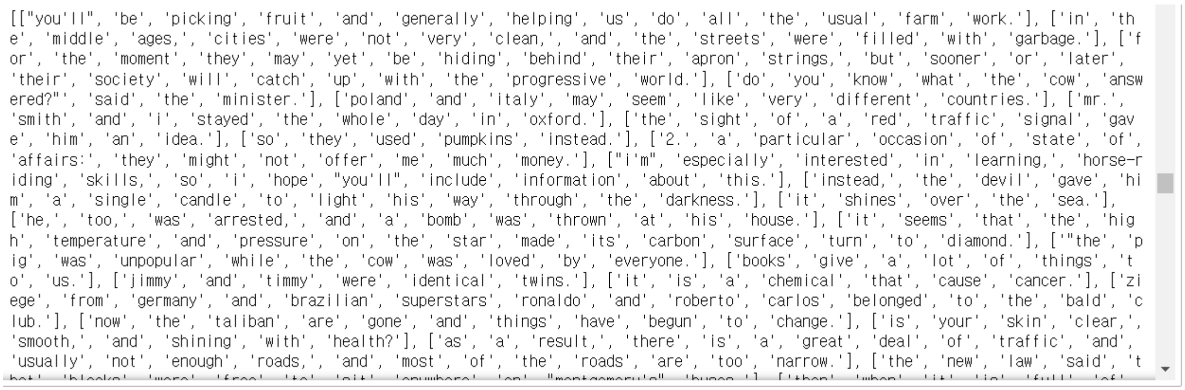
    - Input: score_dict
    - Result: sorted_score_list
    - Explanation: Sort in order of similarity larger

- Output the top 10 ranked sentences to the user from the ranked sentences
    - Input: This is not function. There is no input
    - Result: Output the top 10 ranked sentences to the user from the ranked sentences
    - Explanation: Output in order of "rank", "Index", "score", and "sentence"

4. **Testing • (Testing : Testing involved verifying if the system provided the expected results based on user input.)**

① Test Results for Each Functionality: (Description of test cases and corresponding screenshots for each requirement)

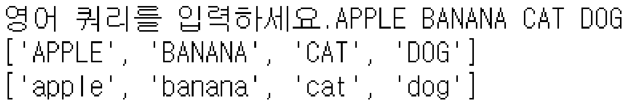
· Indexing



```
[["you'll", 'be', 'picking', 'fruit', 'and', 'generally', 'helping', 'us', 'do', 'all', 'the', 'usual', 'farm', 'work.'], ['in', 'th
e', 'middle', 'ages,', 'cities', 'were', 'not', 'very', 'clean,', 'and', 'the', 'streets', 'were', 'filled', 'with', 'garbage.'], ['f
or', 'the', 'moment', 'they', 'may', 'yet', 'be', 'hiding', 'behind', 'their', 'apron', 'strings,', 'but', 'sooner', 'or', 'later',
'their', 'society', 'will', 'catch', 'up', 'with', 'the', 'progressive', 'world.'], ['do', 'you', 'know', 'what', 'the', 'cow', 'answ
ered?"', 'said', 'the', 'minister.'], ['poland', 'and', 'italy', 'may', 'seem', 'like', 'very', 'different', 'countries.'], ['mr.',
'smith', 'and', 'i', 'stayed', 'the', 'whole', 'day', 'in', 'oxford.'], ['the', 'sight', 'of', 'a', 'red', 'traffic', 'signal', 'gav
e', 'him', 'an', 'idea.'], ['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-r
iding', 'skills,', 'so', 'i', 'hope', "you'll", 'include', 'information', 'about', 'this.'], ['instead,', 'the', 'devil', 'gave', 'hi
m', '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', 'hig
h', 'temperature', 'and', 'pressure', 'on', 'the', 'star', 'made', 'its', 'carbon', 'surface', 'turn', 'to', 'diamond.'], ['"the', 'p
ig', 'was', 'unpopular', 'while', 'the', 'cow', 'was', 'loved', 'by', 'everyone.'], ['books', 'give', 'a', 'lot', 'of', 'things', 't
o', 'us.'], ['jimmy', 'and', 'timmy', 'were', 'identical', 'twins.'], ['it', 'is', 'a', 'chemical', 'that', 'cause', 'cancer.'], ['zi
ege', 'from', 'germany', 'and', 'brazilian', 'superstars', 'ronaldo', 'and', 'roberto', 'carlos', 'belonged', 'to', 'the', 'bald', 'c
lub.'], ['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', 't
hat', 'blocks', 'were', 'free', 'to', 'sit', 'anywhere', 'on', "montgomery's", 'buses.'], ['then', 'when', 'it', 'is', 'full', 'of',
```

- Receive an input English string (query) from the user and preprocess it

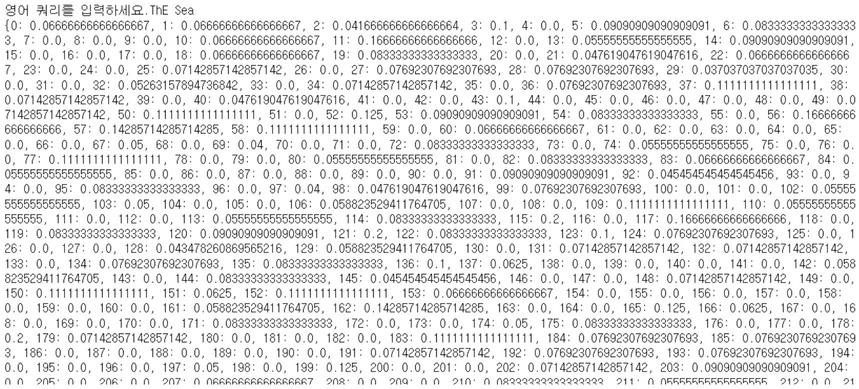
A.



```
영어 쿼리를 입력하세요.APPLE BANANA CAT DOG
['APPLE', 'BANANA', 'CAT', 'DOG']
['apple', 'banana', 'cat', 'dog']
```

- Calculate the similarity between the query and sentences within the search target(Similarity is based on the count of the same “word.")

A.

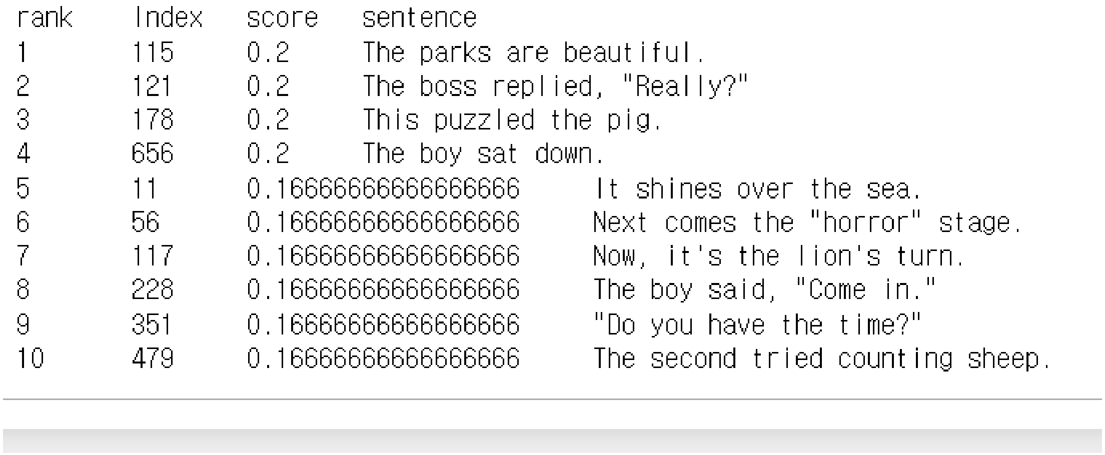


```
영어 쿼리를 입력하세요.ThE Sea
{0: 0.06666666666666667, 1: 0.06666666666666667, 2: 0.041666666666666664, 3: 0.1, 4: 0.0, 5: 0.09090909090909091, 6: 0.0833333333333333
3, 7: 0.0, 8: 0.0, 9: 0.0, 10: 0.06666666666666667, 11: 0.16666666666666666, 12: 0.0, 13: 0.05555555555555555, 14: 0.09090909090909091,
15: 0.0, 16: 0.0, 17: 0.0, 18: 0.06666666666666667, 19: 0.08333333333333333, 20: 0.0, 21: 0.047619047619047616, 22: 0.0666666666666666
7, 23: 0.0, 24: 0.0, 25: 0.07142857142857142, 26: 0.0, 27: 0.07692307692307693, 28: 0.07692307692307693, 29: 0.037037037037037035, 30:
0.0, 31: 0.0, 32: 0.05263157894736842, 33: 0.0, 34: 0.07142857142857142, 35: 0.0, 36: 0.07692307692307693, 37: 0.1111111111111111, 38:
0.07142857142857142, 39: 0.0, 40: 0.047619047619047616, 41: 0.0, 42: 0.0, 43: 0.1, 44: 0.0, 45: 0.0, 46: 0.0, 47: 0.0, 48: 0.0, 49: 0.0
7142857142857142, 50: 0.1111111111111111, 51: 0.0, 52: 0.125, 53: 0.09090909090909091, 54: 0.08333333333333333, 55: 0.0, 56: 0.16666666
666666666, 57: 0.14285714285714285, 58: 0.1111111111111111, 59: 0.0, 60: 0.06666666666666667, 61: 0.0, 62: 0.0, 63: 0.0, 64: 0.0, 65:
0.0, 66: 0.0, 67: 0.05, 68: 0.0, 69: 0.04, 70: 0.0, 71: 0.0, 72: 0.08333333333333333, 73: 0.0, 74: 0.05555555555555555, 75: 0.0, 76: 0.
0, 77: 0.1111111111111111, 78: 0.0, 79: 0.0, 80: 0.05555555555555555, 81: 0.0, 82: 0.08333333333333333, 83: 0.06666666666666667, 84: 0.
05555555555555555, 85: 0.0, 86: 0.0, 87: 0.0, 88: 0.0, 89: 0.0, 90: 0.0, 91: 0.09090909090909091, 92: 0.045454545454545456, 93: 0.0, 9
4: 0.0, 95: 0.08333333333333333, 96: 0.0, 97: 0.04, 98: 0.047619047619047616, 99: 0.07692307692307693, 100: 0.0, 101: 0.0, 102: 0.05555
555555555555, 103: 0.05, 104: 0.0, 105: 0.0, 106: 0.058823529411764705, 107: 0.0, 108: 0.0, 109: 0.1111111111111111, 110: 0.05555555555
555555, 111: 0.0, 112: 0.0, 113: 0.05555555555555555, 114: 0.08333333333333333, 115: 0.2, 116: 0.0, 117: 0.16666666666666666, 118: 0.0,
119: 0.08333333333333333, 120: 0.09090909090909091, 121: 0.2, 122: 0.08333333333333333, 123: 0.1, 124: 0.07692307692307693, 125: 0.0, 1
26: 0.0, 127: 0.0, 128: 0.043478260869565216, 129: 0.058823529411764705, 130: 0.0, 131: 0.07142857142857142, 132: 0.07142857142857142,
133: 0.0, 134: 0.07692307692307693, 135: 0.08333333333333333, 136: 0.1, 137: 0.0625, 138: 0.0, 139: 0.0, 140: 0.0, 141: 0.0, 142: 0.058
823529411764705, 143: 0.0, 144: 0.08333333333333333, 145: 0.045454545454545456, 146: 0.0, 147: 0.0, 148: 0.07142857142857142, 149: 0.0,
150: 0.1111111111111111, 151: 0.0625, 152: 0.1111111111111111, 153: 0.06666666666666667, 154: 0.0, 155: 0.0, 156: 0.0, 157: 0.0, 158:
0.0, 159: 0.0, 160: 0.0, 161: 0.058823529411764705, 162: 0.14285714285714285, 163: 0.0, 164: 0.0, 165: 0.125, 166: 0.0625, 167: 0.0, 16
8: 0.0, 169: 0.0, 170: 0.0, 171: 0.08333333333333333, 172: 0.0, 173: 0.0, 174: 0.05, 175: 0.08333333333333333, 176: 0.0, 177: 0.0, 178:
0.2, 179: 0.07142857142857142, 180: 0.0, 181: 0.0, 182: 0.0, 183: 0.1111111111111111, 184: 0.07692307692307693, 185: 0.0769230769230769
3, 186: 0.0, 187: 0.0, 188: 0.0, 189: 0.0, 190: 0.0, 191: 0.07142857142857142, 192: 0.07692307692307693, 193: 0.07692307692307693, 194:
0.0, 195: 0.0, 196: 0.0, 197: 0.05, 198: 0.0, 199: 0.125, 200: 0.0, 201: 0.0, 202: 0.07142857142857142, 203: 0.09090909090909091, 204:
0.0, 205: 0.0, 206: 0.0, 207: 0.06666666666666667, 208: 0.0, 209: 0.0, 210: 0.08333333333333333, 211: 0.05555555555555555, 212: 0.0, 21
```

- Rank the sentences based on similarity

- Output the top 10 ranked sentences to the user from the ranked sentences

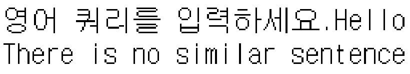
A.



```
rank    Index   score   sentence
1       115     0.2     The parks are beautiful.
2       121     0.2     The boss replied, "Really?"
3       178     0.2     This puzzled the pig.
4       656     0.2     The boy sat down.
5       11      0.16666666666666666     It shines over the sea.
6       56      0.16666666666666666     Next comes the "horror" stage.
7       117     0.16666666666666666     Now, it's the lion's turn.
8       228     0.16666666666666666     The boy said, "Come in."
9       351     0.16666666666666666     "Do you have the time?"
10      479     0.16666666666666666     The second tried counting sheep.
```

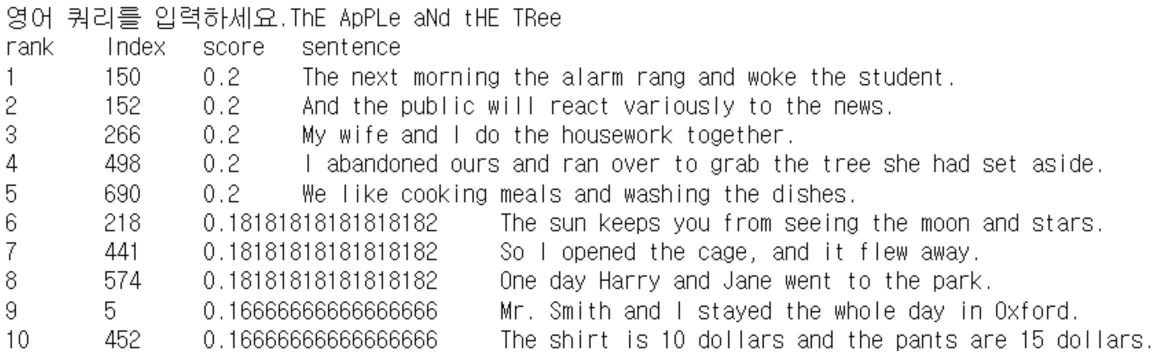
② Final Test Screenshot:: ([Comprehensive screenshot demonstrating the overall functionality of the program)

A. If there is no similar sentence

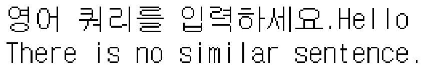


```
영어 쿼리를 입력하세요.Hello
There is no similar sentence.
```

B. If there is a similar sentence



```
영어 쿼리를 입력하세요.ThE ApPLe aNd tHE TRee
rank    Index   score   sentence
1       150     0.2     The next morning the alarm rang and woke the student.
2       152     0.2     And the public will react variously to the news.
3       266     0.2     My wife and I do the housework together.
4       498     0.2     I abandoned ours and ran over to grab the tree she had set aside.
5       690     0.2     We like cooking meals and washing the dishes.
6       218     0.18181818181818182     The sun keeps you from seeing the moon and stars.
7       441     0.18181818181818182     So I opened the cage, and it flew away.
8       574     0.18181818181818182     One day Harry and Jane went to the park.
9       5       0.16666666666666666     Mr. Smith and I stayed the whole day in Oxford.
10      452     0.16666666666666666     The shirt is 10 dollars and the pants are 15 dollars.
```



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

5. **Results and Conclusion**

① 1. Result: Functioned search engine programs and implemented all case-insensitive requirements

② 2. Conclusion: I'm sorry that I can't distinguish special characters