

코테코치(CTCH)

20222592 권민정

INDEX

1. 프로젝트 개요
2. 간략한 기능 계획
3. 사용 API
4. 구현 결과

1. 프로젝트 개요

- 코딩테스트를 요구하는 기업들이 증가하면서 개발자로 취업하기 위해 코딩테스트를 준비하는 사람들 증가
- 혼자서 학습을 진행하면서 올바른 방향으로 학습하고 있는지에 대한 불안감 존재

-> 이러한 문제 해소를 위해 AI 기반 코딩테스트 코치를 개발

채용 전형

[Recruiting Process]

- 서류검토 > **라이브 코딩테스트 및 1차 직무면접** > 온라인 인성검사 및 2차 컬처핏 면접(레퍼런스체크) > 처우 및 입사일 합의 > 최종입사
- * 2차 컬처핏 면접은 최종 합격이 아니며, 연봉 및 근로조건 합의가 된 이후 최종 입사가 확정됩니다.
- * 인성검사는 2차 컬처핏 면접의 참고자료로 1차 직무면접 합격 후 온라인으로 응시하시게 됩니다.(약 40분)

2. 간략한 기능 계획

- 1) 사용자가 작성한 코드 피드백
- 2) 사용자가 원하는 알고리즘을 기반으로 문제 추천
- 3) 사용자가 어려워하는 문제에 대해 힌트 제공

```

31 def __init__(self):
32     self.file = None
33     self.fingerprints = set()
34     self.logdupes = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.log"),
39                         "a")
40         self.file.seek(0)
41         self.fingerprints.update(self._get_fingerprints())
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool("DEBUG", False)
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)

```

3. 사용 API

- ChatGPT API

-> Model : gpt-4o

-> 입력값 : 문제 링크, 사용자가 작성한 코드 파일(txt), 문제 추천을 원하는 링크 (ex. 백준, leetcode)

```
from openai import OpenAI

client = OpenAI(api_key = key)
response = client.chat.completions.create(model='gpt-4o',
                                          messages = [{ 'role': 'user',
                                                         'content': content}],
                                          temperature = 0.1)

print(response.choices[0].message.content)
```

4. 구현 결과 - 코드 피드백

- 코드 평가, 개선점, 공부할 부분, 개선된 코드 등의 피드백 제공
- ChatGPT 4.0이 더 정확한 답변을 제공하기에 해당 모델 사용
- 답변의 일관성을 제공하고자 temperature, top_p를 작게 설정

1. 사용자가 푼 코딩 문제 피드백

```
2288'  
三 : {code}, 취업을 위해 코딩테스트를 준비하고 있는데 문제에 대해 내가 작성한 코드를 보고 간결하게 평가하고 어떤 공부가 필요한지 말해줘.
```

4. 구현 결과 – 문제 추천

- 코딩테스트를 위해 알아야 하는 알고리즘 14개를 기반으로 문제 추천
- 사용자로부터 원하는 알고리즘과 문제 개수를 입력 받아 작업 수행
- 난이도 조절 가능

2. 사용자가 원하는 알고리즘을 기반으로 문제 추천

```
In [9]: types = '배열, 연결리스트, 스택, 큐, 덱, 스택의 활용, BFS, DFS, 재귀, 백트래킹, 시뮬레이션, 정렬, 다이나믹 프로그래밍, 그리디'.split(',')

for i in range(1,8):
    print(f'{i}. {types[i-1]}'.ljust(14, ' '), f'{i+7}. {types[i+6]}'.ljust(15, ' '))

user_want_type = int(input('어떤 알고리즘 문제를 원하시나요?'))
problem_number = int(input('몇 문제를 추천받고 싶으신지 말씀해주세요.'))
problem_level = input('난이도 선택 : 상, 중, 하')

content2 = f'{types[user_want_type-1]}과 관련된 코딩테스트 문제 중 난이도가 {problem_level}인 문제 {problem_number}개를 https://www.acn
```

1. 배열	8. DFS
2. 연결리스트	9. 재귀
3. 스택	10. 백트래킹
4. 큐	11. 시뮬레이션
5. 덱	12. 정렬
6. 스택의 활용	13. 다이나믹 프로그래밍
7. BFS	14. 그리디

4. 구현 결과 – 어려운 문제 힌트 제공

- 사용자가 풀기 어려워 하는 문제를 입력받아 힌트 제공
- max_tokens를 500으로 제한
-> 이러한 제약 때문인지 반말을 사용함
- 실제 GPT가 제공한 힌트가 문제 풀이에 많은 도움이 됨

3. 사용자가 풀기 어려운 문제가 있을 경우 약간의 힌트 제공

```
difficult = 'https://www.acmicpc.net/problem/1312'
content3 = f'''문제 링크 : {difficult}를 푸는 데 어려움이 있어. 어떤 방식으로 풀어야 하는지 간단하게 힌트를 알려줘.
절대 답을 말하면 안돼.'''

response = client.chat.completions.create(model='gpt-4o',
                                           messages = [{ 'role': 'user',
                                                         'content': content3}],
                                           temperature = 0.1,
                                           top_p=0.1,
                                           max_tokens=500)

print(response.choices[0].message.content)
```

물론, 문제를 푸는 데 도움이 될 수 있는 힌트를 제공할게.

문제 1312는 A를 B로 나눈 몫의 소수점 아래 K번째 숫자를 구하는 문제야. 이 문제를 해결하기 위해서는 실제로 나눗셈을 수행하면서 소수점 아래 자릿수를 하나씩 구해나가는 방법을 사용할 수 있어.

1. A를 B로 나누어 몫을 구하고, 나머지를 계산해.
2. 나머지에 10을 곱해서 다음 자릿수를 구할 준비를 해.
3. 이 과정을 K번 반복해서 K번째 소수 자릿수를 찾아.

이 방법을 통해 소수점 아래 K번째 숫자를 직접 구할 수 있을 거야. 나눗셈을 수작업으로 하는 과정을 생각해보면 도움이 될 거야. 화이팅!

THANK YOU

20222592 권민정