

뇌 병변 및 발달장애인을 위한 피해 신고 웹프로그램

권석현¹, 고상현², 백종록³, 장지욱⁴, 지도교수 신연순⁵

¹동국대학교 일본학과, ²동국대학교 바이오환경과학과, ³동국대학교 교육학과,

⁴동국대학교 경찰행정학부, ⁵동국대학교 컴퓨터공학과

sukhyun205@gmail.com, sanghyun123452@gmail.com, supporterleo@naver.com,

swjw28@dgu.ac.kr, ysshin@dongguk.edu

Damage reporting web program for people with brain lesions and developmental disabilities

Suk-hyun Kwon¹, Sang-hyun Ko², Jong-rok Baek³,

Ji-wook Jang⁴, Yeon-soon Shin⁵

요 약

본 논문은 폭행, 인권침해 등 특수한 상황에 부딪친, 의사소통이 힘든 뇌 병변 및 발달장애인을 위한 상담 챗봇을 소개한다. KoGPT를 활용하여 질문리스트를 작성하고 데이터베이스를 생성하였다. Github face_recognition을 통한 이미지 학습을 통해 사용자를 인증하고 NGINX, redis, Django, daphne를 사용하여 챗봇의 아키텍처를 만들었다. 이를 통해 장애인이 장소와 시간에 구애받지 않고 직접 신고하여 빠른 보호와 행정 조치를 받을 수 있다. 또한 상담원 인력 부족 문제에도 도움을 줄 수 있다.

Abstract

This paper introduces counseling chatbot for people with brain lesions and developmental disabilities who face special situations such as assault and human rights violations. A question list was created and a database was created using KoGPT. We authenticated the user through image learning through Githubface_recognition and created the chatbot architecture using NGINX, redis, Django, and daphne. Through this, disabled people can report directly regardless of place and time and receive quick protection and administrative measures. It can also help with the problem of a shortage of counselors.

Keywords

distributed graph database, recommender systems, collaborative filtering, cloud computing

1. 서 론

현재 국내 발달장애인의 비율은 2010년 7.0%에서 2020년 9.4%로 증가해, 10%에 육박할 만큼 증가추세에 있다. [1] 이와 함께 장애인 학대 신고 건수는 매년 증가하고 있다. 그러나 장애인 스스로 본인의 피해 상황을 신고하는 것에 제약이 많은 것이 현실이다. 2021년 장애인 학대 의심 사례 2,461건 중 발달장애인의 직접 신고는 167건으로 약 7%에 불과하다. [2] 실제 학대 피해자의 약 40%가 지인 또는 복지시설 종사자만 학대가 발생했을 경우 피해를 신고할 수 있는 수단이 매우 제한적일 수밖에 없다. 그뿐만 아니

라 장애인 피해상담 인원의 부족과 코로나로 인한 비대면 상황은 발달장애인의 학대 신고와 조사를 어렵게 하고 있다. 이에 비대면 상황에서 장애인이 주체적으로 신고할 수 있는 웹 프로그램을 만들고자 한다. 먼저 발달장애인에 대한 이해가 필요하기에 서울시 장애인 의사소통 권리위원회, 서울특별시 장애인권익 옹호 기관과 면담하여 발달장애인의 언어 수준, AAC (보완·대체 의사소통), 필요한 서비스 등에 대한 자문을 구했다. 자문 내용과 조사 내용을 바탕으로 챗봇을 통한 장애인의 직접 신고 웹프로그램을 구현하기로 했다. 장애인이 이해하기 쉬운 글과 문장으로 질문을 생성하고 장애인 신원 파악을 위한 안면인식을 위

해 딥러닝 방식을 조사하고 사용했다.

II. 선행사례

장애인 스스로 신고할 수 있는 수단에 관한 연구는 크게 진전된 바가 없다. 본 조는 그 이유를 본 프로젝트를 주로 사용하게 될 발달장애인이 스스로 피해를 당했다고 인지하는 것이 어렵다는 것으로 보고 있다. 하지만 장애인 학대 현황 보고서에 따르면 2019년부터 2021년까지 장애인 본인에 의한 신고 건수는 꾸준히 증가하고 있음을 확인했다. [1],[2],[3]

본 조의 프로젝트를 주로 사용할 발달장애인은 AAC (Augmentative and Alternative Communication) 보완·대체 의사소통 사용법에 대한 학습을 받는다. AAC는 단어와 이미지 상자를 통해 단어 카드를 조합해 단어와 음성을 통해 상대방과의 의사소통을 보조해주는 프로그램이다.

늘어나는 장애인 신고자의 필요에 맞추어 본 조는 발달장애인에게 친숙한 AAC를 활용해 발달장애인 친화적 신고 챗봇을 제작했다. 챗봇을 통해 장애인은 자신의 피해 상황을 신속하게 관련 기관으로, 면담자의 도움 없이 신고함으로써 관련 인력이 부족한 현 상황을 타개함과 동시에 학대 피해를 제때 대응할 수 있게 될 것이라고 본다. 챗봇 제작을 위해 한국어 언어 모델인 KoGPT 2와 대화형 학습모델을 활용했다. [4] 초안에서는 Konlpy를 활용해 면담자와 장애인의 1:1 온라인 면담에서 면담자가 입력한 내용을 형태소 분석을 통해 의도를 파악하고 자동으로 장애인이 이해하기 쉬운 내용으로 변형하고자 했으나 실용성에 문제가 있을 것이라는 의견에 따라 챗봇 제작으로 계획을 변경했다. [5]

활용한 데이터는 모두 오픈소스로 상업적 용도를 제외한 경우에 한 해 사용할 수 있다. 본 조는 장애인의 피해 상황 파악 시 질문하는 순서와 화법 [6],[7] 그리고 판례집에서 [8] 실제 사례 등을 고려하여 질문을 새롭게 작성하고 이를 KoGPT 2에 적용하여, 출력된 질문을 참고하여 챗봇이 출력할 질문리스트를 생성했으며, 챗봇의 흐름과 출력하는 내용에 대해서는 본 조의 프로젝트에 특화된 알고리즘을 새롭게 작성하여 적용하였다.

II. 발달장애인이 주체적으로 신고할 수 있는 웹프로그램

3.1 프로젝트 목표

폭행, 인권침해 등 특수한 상황에 부딪친, 의사소통이 힘든 뇌병변 및 발달장애인이 직접 피해를 신고할 수 있도록 하는 것이 본 프로젝트의 목표이다.

따라서 이해력이 부족한 발달장애인의 정확한 답변을 이끌어내기 위해, 그림1의 이해하기 쉬운 그림 이미지로 구성된 AAC 보완·대체 의사소통 프로그램을 활용하였다.



그림 1 보완·대체 의사소통 프로그램

Fig. 1 AAC (Augmentative and Alternative Communication)

3.2 피해 신고를 위한 질문리스트 작성

한국어를 사전적, 문맥적으로 이해하고 사용자가 원하는 결과값을 보여주는 한국어 AI 언어 모델 KoGPT[4]를 활용하여 질문리스트를 작성하였다. 피해를 신고하는 장애인의 폭행 사건을 조건으로 하여 질문 예시를 출력하였고, 처음에는 출력되는 예시가 같은 답변을 반복하거나 기존의 조건에서 떨어져 소설이 작성되거나 뉴스 형태로 내용이 바뀌는 문제가 발생했다. 이는 kogpt-2가 뉴스, 위키백과 등을 기반으로 학습된 것이 원인으로 보였다. 이를 해결하기 위해 github에서 찾은 오픈소스 대화형 학습모델 데이터셋과 판례집 및 읽기 쉬운 자료 작성법을 참고하여 조건을 추가하였고, 이를 바탕으로 그림 2와 같은 질문리스트를 작성할 수 있었다.

A	B	A	B
1 Q		13 어디서	(tag_people)한테 어디서 맞았나요?
2 누가	누가 때렸는지 말해줄 수 있나요?	14	(tag_people)(이)가 당신을 때린 장소가 어디예요?
3	누가 당신을 때렸나요?	15	(tag_people)(이)가 때린 장소가 어디인지 기억나요?
4	누가 때렸는지 말해주세요.	16	어디서 (tag_people)한테 맞았나요?
5	누가 때렸는지 기억이 나면 말해주세요.	17	(tag_people)(이)가 어디서 때렸는지 알아요?
6	때린 사람이 누구인지 기억나요?	18	어디서 맞았는지 기억나요?
7	때린 사람이 누구인지 아니요?	19	어디서 맞았는지 말해주세요.
8	때린 사람이 누구인지 말해줄 수 있어요?	20	(tag_people)(이)가 어디서 아프게 했어요?
9	때린 사람을 알려주세요.	21	(tag_people)(이)가 아프게 한 장소가 어디인지 기억나요?
10	누구에게 맞았는지 기억나요?	22	(tag_people)(이)가 때린 장소를 알려주세요.
11	누구에게 맞았나요?	23	...
12	...		

그림 2 kogpt-2학습결과 참조 질문리스트 (누가, 어디서)

Fig. 2 kogpt-2 learning outcome reference question list (who, where)

3.3 질문리스트를 기반으로 챗봇 알고리즘 작성

기본적으로 챗봇이 질문리스트를 기반으로 질문을 하면 피해를 본 장애인의 답변을 받아 저장하는 형식이며, 장애인이 명확하게 질문에 답변했는지 확인하기 위해 label 시스템을 적용했다. label은 0과 1로 구분되며, 먼저 label(=0)의 질문을 통해 장애인이 질문의 내용을 이해하고 있는지 확인하고, 그 후에 label(=1)의 질문을 통해 장애인의 구체적인 피해 상황답변을 이해하기 쉬운 AAC 그림 자료[7]를 통해 답변받아 저장한다. 주요 답변자인 장애인은 각 질문에 대해 [네, 아니요, 잘 모르겠

3.4 안면인식 오픈소스 활용

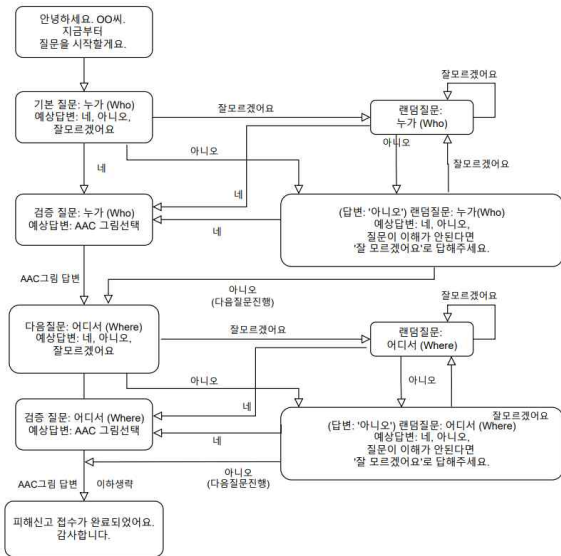


그림 3 피해 신고 다이어그램
Fig. 3 Damage reporting diagram

3.4 웹서버 설계

웹 프레임워크로는 그림4와 같이 Django를 사용해서 채팅 기능을 구현했으며, 웹서버와 Django서버를 연동하기 위해 uwsgi로 django와 nginx를 연결했다. 채팅은 http 프로토콜이 아닌 ws(웹 소켓) 프로토콜을 사용했기 때문에, nginx에서는 리버스 프록시를 사용하여 ws 프로토콜의 경우 uwsgi가 아닌, daphne 서버와 django를 연결하였다. 또한 비동기 통신 연결을 위해 필요한 daphne 서버를 구축하였으며, nginx에서 리버스 프록시 설정을 통해 ws 프로토콜 연결이 오면, daphne 서버가 django를 연결해주도록 설정했다.

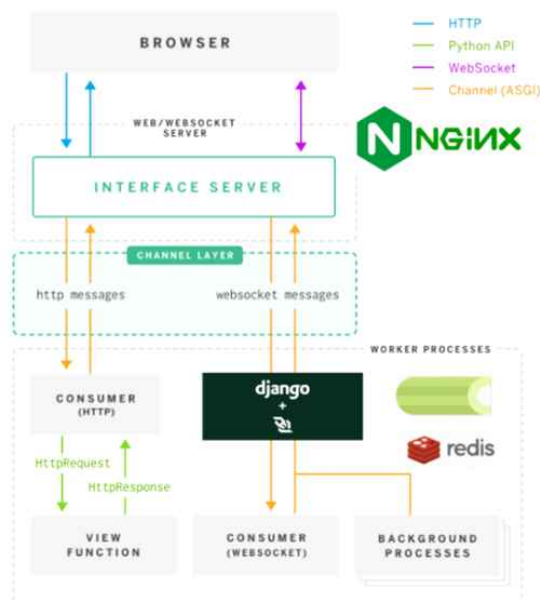


그림 4 웹서버 설계도
Fig. 4 web server blueprint

상담대상자인 장애인을 식별하기 위해 안면인식 인증방식을 도입하였으며 실행화면은 아래 그림 5와 같다.



그림 5 안면인식 오픈소스 활용
Fig. 5 Facial Recognition Open Source Utilization

인면인식을 통한 인증과정에 관해 서술하면 다음과 같다. 먼저 상담대상자인 장애인인 얼굴 사진을 머신러닝을 통해 학습시킨 후, 카메라에 비치는 사람의 얼굴 또는 사진과 학습시킨 데이터를 비교, 대조한다. 만약 등록된 사람이 맞다면 최종적으로 상담을 시작하게 되며, 만약 등록된 사람이 아닌 경우에는 상담프로그램을 실행시킬 수 없다.

3.5 보완·대체 의사소통 프로그램(AAC) 활용

상대적으로 문장의 이해력이 부족한 발달장애인의 명확한 피
해답변을 어떻게 이끌어낼지 서울시 장애인 의사소통 권리증진
센터에 자문한 결과, 그림6의 이해하기 쉽게 직관적인 이미지와
텍스트를 출력해주는 ‘나의 AAC 프로그램을 소개받았으며, 발
달장애인은 아래에 보이는 막대기 및 돌과 같은 폭행 수단, 팔
다리와 같은 신체 부위 등, 폭행당한 수단과 폭행 부위에 대한
구체적인 진술을 직관적인 이미지와 텍스트를 통해 명확하게
답변할 수 있다.

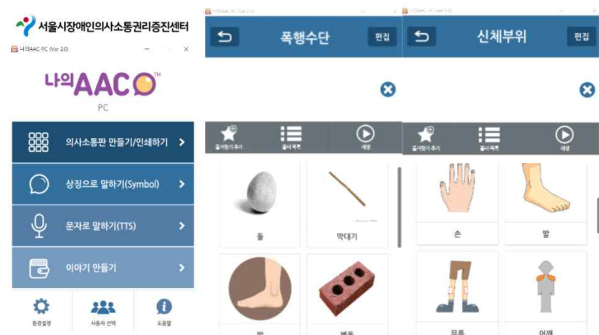


그림 6 나의 AAC 프로그램 (폭행 수단, 신체 부위)
Fig. 6 My AAC program (means of assault, body parts)

IV. 결론

본 논문에서는 뇌 병변 및 발달장애인의 폭행 피해 상황 신고를 위해 여러 가지 질문을 통해 장애인의 피해 사실을 파악할 수 있는 챗봇을 설계하고 개발하

였다. 이를 위해 언어 예측 모델인 KoGPT 2[4]와 보완·대체 의사소통 프로그램인 AAC[7]를 활용하였다. KoGPT 2에서 학습시킨 문장들을 이용하여 새로운 질문을 생성해냈고, 이렇게 생성된 질문들에 대한 대담으로 AAC 그림상징을 사용하였다. 질문을 생성하면서 문맥적, 문법적으로 자연스러운 문장이 되는지를 확인하였고, 장애인들이 더 쉽게 이해하고 답을 할 수 있도록 AAC의 기능을 이용하여 대담을 매끄럽게 이끌어 낼 수 있었다. 또한, 로그인을 통한 인증 절차가 아닌 장애인들이 더 쉽고 편리하게 서비스를 이용할 수 있도록 안면인식 기술을 적용하였다. 상담대상자의 식별을 위해 안면인식 오픈소스를 도입하여 대상자 본인임을 확인한 후 상담이 진행될 수 있도록 하였다. 이렇게 개발된 챗봇을 통해 부족한 상담 인력의 문제를 해결하고 장애인 스스로가 피해를 신고할 수 있다는 가능성을 확인하였다. 현재 장애인 폭행 피해 건수가 증가하고 있는 만큼 해당 기술이 도입될 경우, 더 빠른 신고가 가능할 것으로 기대된다. 또한 직접 신고가 어려운 장애인들에게 스스로 신고를 할 수 있도록 하여 장애인의 자율권을 보다 확장할 수 있을 것으로 예상된다. 추후 더욱 학습된 질문을 통해 피해 당시 상담자의 기분, 심정 등을 파악할 수 있도록 하고, 체계화된 질문 알고리즘을 적용하여 상담자의 대답에 알맞은 질문이 자동으로 생성되도록 구현하는 연구가 필요하다. 또한, 장애인 개개인의 특성에 맞추어 원하는 상담방식을 설정할 수 있는 커스터마이징 된 챗봇 시스템을 구현하는 것을 목표로 한 기술 고도화를 기대할 수 있다.

참 고 문 헌

- [1] 보건복지부, 중앙장애인 권익옹호 기관. “2019 장애인 학대 현황 보고서”, 72. (2019).
- [2] 보건복지부, 중앙장애인 권익옹호 기관. “2020 장애인 학대 현황 보고서”, 92. (2020).
- [3] 보건복지부, 중앙장애인 권익옹호 기관. “2021 장애인 학대 현황 보고서”, 92. (2021).
- [4] <https://github.com/SKT-AI/KoGPT2>. (2022).
- [5] 박은정, 조성준. KoNLPy: 쉽고 간결한 한국어 정보처리 파이프라인 패키지, 제26회 한글 및 한국어 정보처리 학술대회 논문집, 133-136. (2014).
- [6] 류경희, 공정식. 학대 피해 장애인에 대한 수사 전 면담기법에 관한 연구. 피해자학 연구, 25(2), 167-213. (2017).
- [7] 읽기 쉬운 자료개발센터 ‘알다’. “발달장애인이 읽기 쉬운 자료 제작 안내서”. (2018).
- [8] 중앙장애인 권익옹호 기관. “장애인 학대 사건 판례집 2017-2020”. (2021).
- [9] <https://github.com/haven-jeon/KoGPT2-chatbot>. (2022).
- [10] 민서영, “지난해 장애인 학대 전년보다 11.5% 증

가...피해자 10명 중 7명은 발달장애인”, 경향신문, <https://m.khan.co.kr/national/health-welfare/article/202209281200011#c2b> (2022)

[11] 박종진. 자연어 처리를 이용한 감정 스트레스 인지 및 관리 챗봇 개발. 전기학회논문지, 67(7), 954-961. (2018).

부록(코드 첨부)

```
1 import torch
2 from transformers import AutoTokenizer, AutoModelForCausalLM
3
4 tokenizer = AutoTokenizer.from_pretrained(
5     'kakaobrain/kogpt', revision='KoGPT6B-ryan1.5b-float16', # or float32 version: revision='KoGPT6B-ryan1.5b
6     bos_token='[BOS]', eos_token='[EOS]', unk_token='[UNK]', pad_token='[PAD]', mask_token='[MASK]'
7 )
8 model = AutoModelForCausalLM.from_pretrained(
9     'kakaobrain/kogpt', revision='KoGPT6B-ryan1.5b-float16', # or float32 version: revision='KoGPT6B-ryan1.5b
10    pad_token_id=tokenizer.eos_token_id,
11    torch_dtype='auto', low_cpu_mem_usage=True
12 ).to(device='cuda', non_blocking=True)
13 _ = model.eval()
14
15 prompt = '인공지능아, 너는 말을 할 수 있니?'
16 with torch.no_grad():
17     tokens = tokenizer.encode(prompt, return_tensors='pt').to(device='cuda', non_blocking=True)
18     gen_tokens = model.generate(tokens, do_sample=True, temperature=0.8, max_length=64)
19     generated = tokenizer.batch_decode(gen_tokens)[0]
```

코드 1 KoGPT2 불러오기

Code. 1 Load KoGPT2

```
7 with torch.no_grad():
8     tokens = tokenizer.encode(prompt, return_tensors='pt').to(device='cuda', non_blocking=True)
9     gen_tokens = model.generate(tokens, do_sample=True, temperature=0.85, max_length=512)
10    generated = tokenizer.batch_decode(gen_tokens)[0]
11
12 print(generated)
```

폭행을 당한 장애인과의 면담 내역입니다.
누가 때렸는지, 언제 때렸는지, 어디서 맞았는지, 무엇으로 맞았는지, 어디를 맞았는지를 알아내기 위해 한 질문들입니다.
장애인의 답변도 포함되어 있습니다.
Q: 누구한테 맞았는지 기억하십니까?
A: 다리가요.
Q: 왜 다리를 맞았다고 생각하나요?
A: 걸지를 못했으니까.
Q: 몇 살 때부터 그 일이 있었는지 기억하십니까?
A: 국민학교 졸업할 때부터요.
Q: 누가 때렸다고 생각하십니까?
A: 아버지가 때렸다고 생각해요.
Q: 맞을 때 기분은 어땠어요?
A: 아파서요.
Q: 많이 아팠을 텐데요.
A: 아팠는데, 참을 수 있을 정도요?
Q: 다리 맞을 때 다리 안 펴졌어요?
A: 아버지가 안 펴졌어요.
Q: 왜 그렇게 생각하나요?
A: 다리가 너무 아팠으니까요.
Q: 언제부터 아팠어요?
A: 초등학교 입학할 때부터요.
Q: 다리가 아파서 학교를 못 가니까 어땠어요?
A: 저 혼자 앉아 있을 수 없을 정도로 아팠어요.
Q: 누가 다리를 다쳤다고 생각하십니까?
A: 아버지가 그랬어요.

코드 2 KoGPT2 실행 예시

Code. 2 Example of running KoGPT2

```
while True:
    ret, frame = video_capture.read()
    if process_this_frame:
        # 1. RGB
        small_frame = cv2.resize(frame, (8, 8), fx=0.25, fy=0.25)
        rgb_small_frame = small_frame[:, :, 0:3]
        face_locations = face_recognition.face_locations(rgb_small_frame)
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
        face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(brown_face_encodings, face_encoding)
            name = "Unknown"
            face_distances = face_recognition.face_distances(brown_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]
            face_names.append(name)
        process_this_frame = not process_this_frame

    for (top, right, bottom, left), name in zip(face_locations, face_names):
        top += 4
        right += 4
        bottom += 4
        left += 4

        # 2. 얼굴 번호와 이름을 표시
        cv2.rectangle(frame, (left, top), (right, bottom), (255, 0, 0), 2)

        # 얼굴에 표시
        # 3. 얼굴 번호와 이름 표시
        cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (255, 255, 255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(frame, name, (left + 5, bottom - 5), font, 0.5, (255, 255, 255), 1)
        rgb_list.append(name)

    # 화면에 표시
    cv2.imshow('video', frame)

    # 모든 창이 닫힌 상태로 실행을 종료
    if cv2.waitKey(1) <= 0:
        counter = counter(rgb_list)
        if counter.max_count[0][0] == 0:
```

코드 3 안면인식 관련 코드

Code. 3 Facial Recognition Related Code

```
while not ansok:
    time.sleep(3)
    screen = win32gui.FindWindow(None, '나의AAC PC (ver 2.0)')
    bbox = win32gui.GetWindowRect(screen)
    img = ImageGrab.grab(bbox)
    area = (3, 270, 150, 320)
    cr_img = img.crop(area)
    cr_img.show()
    ans = pytesseract.image_to_string(cr_img, lang='kor', config='-c preserve_interword_spaces=1 --oem 3 --psm 7')
    ans = ans.strip()
    print(ans)
```

코드 4 파이테서렉트를 활용한 AAC 앱의 문자 인식

Code. 4 Character Recognition in AAC App Using Pytesseract

```
# 17기
def __who_():
    no_ans = '모르는 사람'
    temp = who_df.query('label == 0')[0].sample(n=5) # csv에서 랜덤으로 랜덤에 5인 질문 5개 추출
    temp = temp.loc[:, 'q'].tolist() # 질문 일련을 리스트로 변환
    pass_cnt = 0 # 0번으로 질문 1번 이상 답할 경우 대상을 모르는 것으로 알고 다음 질문을 넘어감
    # 5개를 묻어 누가 태워는지 알아오기 라는 질문에 2번 4번으로 답하면 어떤 인물이 가려져 있지는 것으로 보고 '어디서' 질문으로 넘어감
    stop_pass = False
    # 다음 label 질문으로 넘어가도 되는지 확인하는 bool

    for q in temp:
        if stop_pass:
            break
        print(q)
        q_pass = False
        # label은 그대로하고 질문만 새롭게 출력하는 것을 확인하는 bool
        while not q_pass:
            ans = input()
            print(ans)
            if ans == '내':
                q_pass = True
                stop_pass = True
            elif ans == '아니요':
                pass_cnt += 1
                if pass_cnt == 2:
                    return no_ans
            elif q == temp[-1]:
                print('가려져 있지 않나 보네요. \n문제를라도 가려져 나면 알려주세요.')
                return no_ans
            print('2번째인 점을 위해서 어떤 사람이 누구인지 확실하게 알 필요가 있어요.\n\n남문이 이해가 안 가시면 잘 모르겠다고 알려주세요.\n\n가려져 있지 않다면 한 번 더 이름을 알려주세요.\n\n모름을 한 번 더 누르면 다음 질문을 할게요.')
            print('물문을 다르게 하세요.')
            q_pass = True
        elif ans == '알못했어':
            print('물문을 다르게 하세요.')
            q_pass = False
        # mac로 답하기 때문에 답변에 오타가 날 경우를 예상했다.
        ...
        elif ans not in expected_Ali:
            print('답변은 네, 아니요, 잘모르겠어요 중 3가지로 해주세요')
            ...
    if q == temp[-1] and stop_pass == False:
        print('가려져 있지 않나 보네요. \n문제를라도 가려져 나면 알려주세요.')
        return no_ans
```

코드 5 질문 알고리즘 예시

Code. 5 Question Algorithm Example