

대한민국 ICT인재를 키우는 힘

- 한이음 ICT멘토링 참여기 2탄

인공지능을 통한 미디어콘텐츠의 다국어자막 번역 서비스 앱개발 프로젝트

글. 강자원 컴퓨터시스템응용기술사, KBS MNC(Media Network Center)팀
강원대학교 안종혁 이현수 오신혁
* 한이음사이트(www.hanium.or.kr)



2004년부터 시작된 ICT멘토링제도 운영 사업은 과학기술정보통신부가 지원하고 정보통신기술진흥센터(現 정보통신기획평가원, IITP)에서 주관하는 인력양성사업으로 대학생(멘티)이 ICT기업전문가(멘토)와 팀을 이루어 프로젝트를 수행함으로써 ICT실무 역량을 향상시키는 대한민국 ICT 분야 대표 멘토링 프로그램이다.

지난달에 한이음 ICT 사업에 대해 간략한 소개가 있었다면 이번 칼럼에서는 지난 12월 19일에 열렸던 한이음 멘토데이 참여 후기와 스마트한 강원대 학생들과 함께 했던 인공지능 프로젝트에 대해 소개하는 시간을 가져보려 한다. 수행 프로젝트에 대한 내용은 학생들이 직접 작성한 결과 보고서를 바탕으로 작성하였다.

한이음 멘토데이

2018년 한이음 멘토데이는 12월 19일 양재동 엘타워에서 열렸다. 한이음 프로젝트는 공식적으로 11월이 되면 마무리가 된다. 한이음 멘토로 활동하게 되면 한이음 멘토데이에 참여할 수 있는 영광의 혜택도 누리게 된다. 올 한해 ICT멘토링 참여 멘토들의 성과를 공유하고 참가자 교류를 통해 인적 네트워크도 형성하는 자리이다. 그리고 올해의 멘토들을 선정해 수상을 한다. 10년 이상을 꾸준히 활동하여 학생들을 지도해온 멘토, 멘티들이 뽑은 올해의 멘토 등 다양한 형태로 멘토



들의 업적을 치하하는 행사를 진행한다. 많은 멘토들이 참여했고 수행했던 프로젝트 중 의미 있었던 경험을 공유하는 자리도 있었다. 다양한 경품행사도 진행되었으나 본인은 아쉽게도 당첨의 기회는 얻지 못하였다.

올해의 한이음 멘토 시상식에서는 올 한해 열성적으로 멘티를 이끈 5명의 멘토를 선정해 정보통신기술진흥센터장상, 한국정보산업연합회장상을 시상하였다. 무엇보다 가장 의미 있는 상은 멘티들이 뽑은 멘토상이었다. 이 상은 멘티와 함께 행사에 참여하여 멘티가 멘토에게 직접 상을 수여한다. 멘티는 우리 멘토가 어떻게 멘토링을 했는지 그리고 어떤 면에서 훌륭한지 이유를 말하였다. 그 이유를 들으며 인상 깊었던 점은 훌륭한 멘토는 멘티들에게 많은 지식을 전달하는 것보다 지쳐있는 멘티들을 위로하고 격려해 추진력을 끌어올리는 것이라 생각했다. 이번 행사를 통해 선배 멘토들의 다양한 경험담은 큰 조언이 되었다.

이번 한이음 ICT멘토링 사업에는 120개 대학 1천800여 명의 멘티가 3대 1의 경쟁률을 뚫고 선발됐다. 신청 프로젝트는 사물인터넷(IoT)이 26%로 가장 많았고 모바일(16%), 인공지능(12%), 빅데이터(9%), 증강·가상현실(5%) 등 순이었다. 필자 또한 12% 안에 속하는 인공지능 프로젝트를 강원대학교 학생들과 수행하였다. 다음은 강원대학교 학생들과 수행한 프로젝트를 소개하려 한다.

프로젝트명 : 인공지능(AI)을 통한 미디어콘텐츠의 다국어자막 번역 서비스 앱개발 프로젝트

팀 명		토마토				
팀 원		이 름	소 속	부서/학과	담당업무	직위/학년
멘 토		강자원	KBS	MNC	프로젝트 지도	감독
멘 티	팀장	안종혁	강원대학교	컴퓨터정보통신	SW 개발	4학년
	팀원1	이현수	강원대학교	컴퓨터정보통신	SW 개발	3학년
	팀원2	오신혁	강원대학교	컴퓨터정보통신	PM & SW 개발	3학년

프로젝트 소개

자연어처리 기술을 바탕으로 영상을 앱에서 재생하면 이를 각종 언어로 번역된 자막을 제공하는 서비스 개발 프로젝트

프로젝트 사용자 시나리오

강원대학교에 재학하고 있는 제임스는 영국에서 온 유학생이다. 제임스의 취미생활은 드라마를 보는 것이다. 한국의 문화를 조금 더 이해하고 학우들과 친해지기 위해 한국 드라마를 보려고 하지만 한국에서 한국어 드라마를 보다 보니 자막을 구하기 힘들다는 불편함이 있다. 제임스는 아직 한국어가 서툴러서 자막이 없으면 내용을 이해하기가 힘들다. 영상의 음성을 자동으로 번역하여 자막으로 출력해주는 프로그램이 있어서 사용해 보았지만 자막 문장에 어색함이 있다는 것을 알았다. 이 불편함을 어떻게 해소해야 할지 고민하던 중 다른 유학생 친구에게서 한국어 영상에 다국어 자막을 넣어주는 다국어 자막 번역 애플리케이션이 있다는 것을 들었고, 다른 프로그램에 비하여 자막 정확도도 높고 이해하기도 쉽다고 들었다. 제임스는 곧바로 그 애플리케이션을 다운받은 후 실행했다. 실행하자 첫 화면에서 원하는 영상을 검색하여 찾을 수 있었다. 검색하여 찾은 영상을 클릭하자 원하는 자막의 언어를 선택하려는 창이 떴다. 제임스는 바로 'English'를 선택하였다. 잠깐의 로딩화면이 뜬 후, 곧바로 자막과 함께 영상이 재생되었다. 놀랍게도 평소 아무리 찾아도 자막을 구하기가 힘들었는데 바로 영어로 된 자막이 나오는 것이었다. 제임스는 호기심에 영어가 아닌 다른 나라의 언어도 클릭해 보았다. 일본어, 중국어뿐만 아니라 불어, 아랍어 등 많은 언어가 지원된다는 것을 알고 감탄했다.

워크플로우 및 역할



미디어 open



음성추출



음성 → 텍스트변환

인공지능
자연어처리텍스트가공
다국어 번역

영상에 다국어자막 제공



현수



종혁



신혁



현수

플레이어 부분은
걱정마! 내가 맡겠어!
파일이나 잘 생성해
주라고!

STT(Speech to Text) 부분은
내가 할게! 그리고 JSON
타입으로 데이터를 넘겨주면
신혁이가 다음을 맡아줘

걱정마! Word Embedding 적용해서
텍스트로 변환된 음성을 vector로
표현하고 패턴 분석을 통한 모델을
학습시키면 된다구! 모델은 나중에
다시 설명해줄게!

처음과 끝은 모두 내가!
안드로이드 개발의
진수를 보여주겠어!

역할 및 수행내용 설명

다국어자막 미디어 플레이어 앱 개발 부문

① 앱 개발 부문 담당자 프로필



현수

이름 : 이현수 (강원대학교 3학년)
담당분야 : 안드로이드기반 플레이어 앱 개발
개발력 : ★★★★★☆
체력 : ★★★★★☆
기술력 : ★★★★★★
기능 : 영상을 불러오고 선택된 자막파일을 연동시켜 플레이어에서 보여주는 기능

② 수행결과 이미지

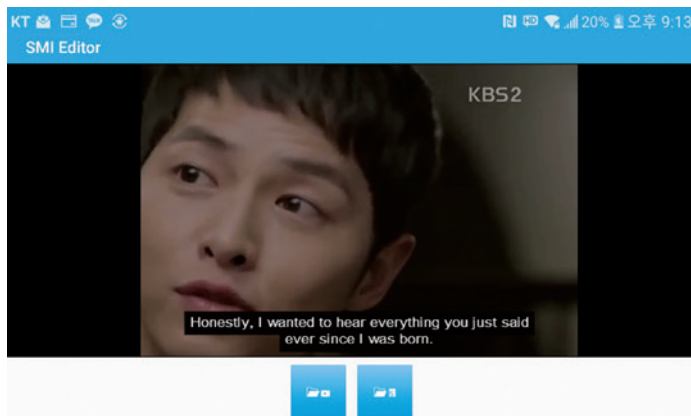


그림 1. 개발 완료된 미디어플레이어 앱 화면

③ 수행내용 중 고군분투 한 부분

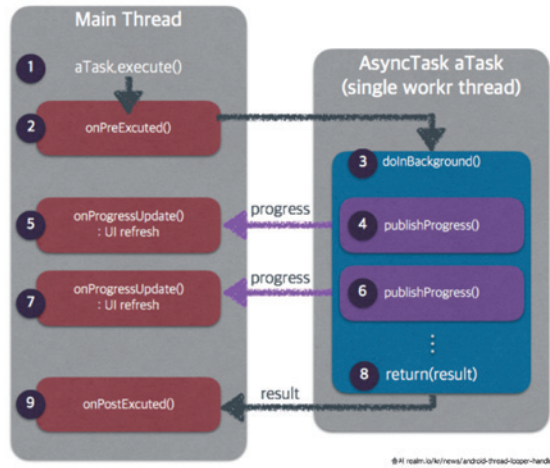


그림 2. AsyncTask 객체의 동작 순서

안드로이드에서 번역 서버로부터 번역된 자막파일을 내려 받기 위해 사용하는 http 통신은 AsyncTask 클래스를 사용하여 구현했다. AsyncTask는 비교적 오래 걸리지 않은 작업에 유용하고, 태스크 캔슬이 용이하며 로직과 UI 조작이 동시에 일어나야 할 때 매우 유용하게 사용된다는 장점이 있어 프로젝트에서 요구하는 조건과 잘 맞아 선택했다. 이 클래스를 사용하여 구현했을 때 가장 힘들었던 부분은 AsyncTask에 전달되는 인자의 값이 무엇이며 어디에 쓰이는지 정확히 파악이 안 돼 사용할 수 없었던 부분과 각 메소드 간 호출되는 순서가 혼란스러웠던 부분이다. 구글링과 사용 예제를 통해 각 인자가 나타내는 값이 무엇인지 알게 되었고 동작 과정도 정확하게 파악할 수 있었다.

```
new AsyncTask<Void,Void,Void>(){
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        strUrl = "http://210.115.46.65:10002/hello-tornado";
    }
    @Override
    protected void doInBackground(Void... voids) {
        try{
            Url = new URL(strUrl);
            String boundary = "+";
            HttpURLConnection conn = (HttpURLConnection) Url.openConnection();
            conn.setRequestMethod("GET");
            conn.setDoOutput(true);
            conn.setDoInput(true);
            conn.setUseCaches(false);
            conn.setDefaultUseCaches(false);
            conn.setRequestProperty("content-type", "multipart/form-data;boundary="+boundary);
            conn.setRequestProperty("Connection", "Keep-Alive");
            InputStream is = conn.getInputStream();
            StringBuilder builder = new StringBuilder();
            BufferedReader reader = new BufferedReader(new InputStreamReader(is, charsetName: "UTF-8"));
            String line;
            while ((line = reader.readLine()) != null) {
                builder.append(line);
            }
            result = builder.toString();
        }catch(MalformedURLException | ProtocolException exception) {
            exception.printStackTrace();
        }catch(IOException io){
            io.printStackTrace();
        }
        return null;
    }
    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);
        System.out.println(result);
    }
}.execute();
```

그림 3. AsyncTask 객체 구현 코드

음성추출 및 텍스트 변환 : STT(Speech To Text) 구현 부분

① STT(Speech to Text) 담당자 프로필

 중혁	이름	: 안중혁 (강원대학교 4학년)
	담당분야	: 음성추출 및 텍스트변환 부분 개발
	개발력	: ★★★★★☆
	리더십	: ★★★★★
	해결력	: ★★★★★
	기능	: 영상에서 오디오를 추출하여 Text 파일로 변환, 번역 서버로 전송하는 기능 구현

② 수행내용

▶ 영상에서 음성추출 부분

MP4 영상으로부터 오디오를 추출하는데 오픈소스인 ffmpeg 라이브러리를 사용했다. 그런데 ffmpeg은 C언어로 구현된 코덱이다. 자바 기반의 안드로이드 개발환경에서 어떻게 활용할 수 있을지 고민 속에 기지를 발휘하여 해결할 수 있었다. 이 코덱을 안드로이드 개발환경에서 포팅하기 위해서는 Android NDK가 필요하다. Android NDK는 C언어와 JAVA를 연결해줄 수 있는 중요한 매개체 역할을 하는 컴파일러이다.

```
try {
    String line;
    String mp4File = "./resources/input.mp4";
    String mp3File = "./resources/output.wav";

    //using cmd
    String cmd = "./ffmpeg/bin/ffmpeg -i " + mp4File + " -ac 1 " + mp3File;
    Process p = Runtime.getRuntime().exec(cmd);
    BufferedReader in = new BufferedReader(
        new InputStreamReader(p.getErrorStream()));
    while ((line = in.readLine()) != null) {
        System.out.println(line);
    }
    p.waitFor();
    System.out.println("Video converted successfully!");
    in.close();
} catch (IOException | InterruptedException e) {
    LOG.log(Level.SEVERE, null, e);
}
```

그림 4. 자바에서 mp4 to mp3 변환 코드

▶ STT(Speech To Text)

STT(Speech To Text)는 자막 생성을 위해 영상에서 변환된 음성을 인식하여 텍스트로 변환하는 작업이다. 실제로 높은 수준의 고난도 작업이기 때문에 구글 클라우드 플랫폼 API를 사용했다. 구글 STT API를 호출하기 위하여 오디오 파일의 샘플레이즈 및 인코딩 방식의 지정이 필요하다. 개발 초기에 다양한 샘플레이즈와 인코딩의 존재를 간과하였고, 조금만 달라도 인식이 되지 않았다. 다양한 인코딩의 존재를 배제하고 영상 형식을 MP4로 제한한 후, ffmpeg을 이용하여 모노 타입의 MP3와 WAV 형식으로 변환했다. 그 후 샘플레이즈를 구하기 위하여 각 형식의 헤더에서 샘플레이즈를 계산하고, LINEAR16 형식을 지정하여 API를 호출하여 해결하였다.

▶ APP과 AI 번역 모델과의 통신

APP과 모델 사이의 통신은 JSON 형식을 사용했는데 STT API 사용 결과를 JSON 형식으로 타임스탬프와 함께 생성했다. JSON은 JavaScript Object Notation의 약자로 경량의 DATA 교환 형식이다. 이 형식은 사람이 읽고 쓰기에 쉬우며, 기계가 분석하고 생성함에도 용이하다.

```
temp = {}
words = []
for result in response.results:
    alt = result.alternatives[0]
    print(u'Transcript: {}'.format(alt.transcript))
    print('Confidence: {}'.format(alt.confidence))
    temp['transcript'] = alt.transcript
    for word in alt.words:
        words.append({'word': word.word,
                      'start_sec': word.start_time.seconds, 'start_ms': int(word.start_time.nanos / 1000000),
                      'end_sec': word.end_time.seconds, 'end_ms': int(word.start_time.nanos / 1000000)})

temp['words'] = words
jsonObj = json.dumps(temp, indent=4)
```

그림 5. Text로 변환된 단어와 영상의 타임스탬프정보를 JSON 형태로 코딩한 부분

딥러닝 알고리즘 적용을 통한 번역모델링 개발 부문

딥러닝 알고리즘 모델링 및 개발

① 모델링 및 개발 담당자 프로필

 신혁	이름	: 오신혁 (강원대학교 3학년)
	담당분야	: 딥러닝 알고리즘을 적용한 번역 모델 설계 및 개발
	개발력	: ★★★★★☆
	리더십	: ★★★★★★
	해결력	: ★★★★★★
기능	: 딥러닝 알고리즘 모델링 및 번역서버 구현	

② 수행내용

▶ 딥러닝 알고리즘 모델 (Seq2seq with attention)

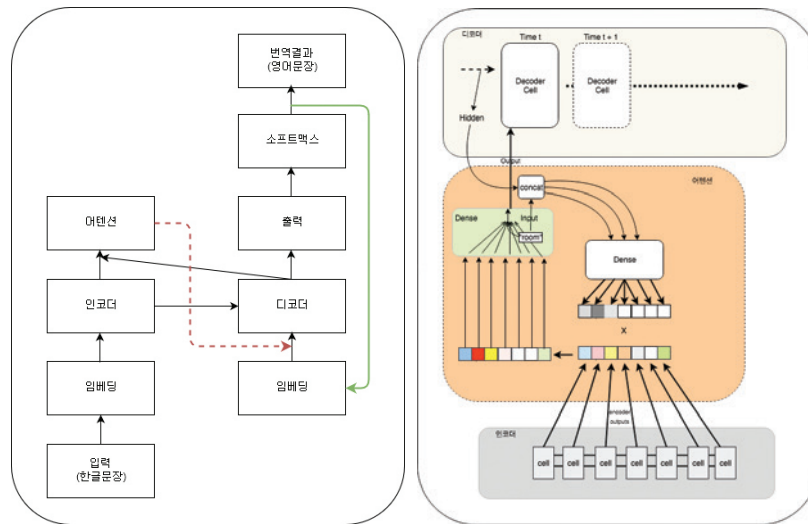


그림 6. Attention mechanism을 이용한 encoder-decoder 아키텍처 모델

Sequence-To-Sequence 모델은 딥러닝 기법의 하나로 네트워크에서는 연속적인 입력(Sequential Input)으로부터 연속적인 출력(Sequential Output)을 생성할 수 있다. 예를 들어, 한글로 된 문장을 영어 문장으로 번역하는 것 또는 인간과 로봇의 대화 시스템(예: 챗봇)을 위한 문장을 생성할 수 있다. 시퀀스-투-시퀀스 모델은 이를 목적으로 디자인 되었다. 그리고 번역하는 서버의 GPU가 충분한 메모리를 가졌다면 기본적인 시퀀스-투-시퀀스 모델의 RNN을 더 깊게 할 수 있기 때문에 더 좋은 성능의 기계 번역을 구현할 수 있다.

이 모델에 적용된 어텐션(Attention)은 기본적인 시퀀스-투-시퀀스 모델에서 인코더 네트워크는 한글 문장을 하나의 벡터 h 로 압축하고 디코더 네트워크는 h 를 이용해 영어 문장으로 출력한다. 그러나 이 기본적인 시퀀스-투-시퀀스 모델에는 두 가지 문제가 있다. 하나는 인코더의 출력 h 가 입력된 한글 문장을 표현하기에는 너무 작다는 것이며 다른 하나는 디코더 네트워크가 각 Step의 출력과 인코더 출력을 어떻게 연관시키는지 알지 못한다는 것이다. 그래서 이 부분을 어텐션 모델을 부가하여 보완한다.

어텐션의 기본적인 아이디어는 인코더의 내부 상태 $h_1, h_2, \dots, h_t$ 의 가중 평균을 디코더에게 전달해주는 것이다. 이 가중(가중치를 더한) 구조를 어텐션이라고 부르는데 디코더 네트워크에 더 많은 정보를 전해주는 이 구조는 어텐션 가중치(Attention Weighting)를 이용해 입력에 관련된 많은 정보를 얻을 수 있다.

► Word Embedding

[그림 7]은 실제로 딥러닝 모델을 적용하여 학습시키기 전에 Word Embedding을 적용하는 모습인데 JSON 형태로 넘겨받은 Text 데이터들은 Embedding train 단계에서 형태소 단위로 쪼개지고 벡터값을 갖게 된다. Word Embedding은 단어 간의 의미 연관성을 거리 단위(벡터값)로 표현한다. 예를 들면 사과와 배는 거리가 가깝고, 사과와 고양이는 거리가 상대적으로 멀다. 아래는 뉴스영상에서 오디오를 추출 후 JSON 형태로 기사를 전달받았고 이를 번역하기 위해 Word Embedding을 생성하는 과정이다.

[illegible]

그림 7. Embedding train 단계에서 형태소 단위 임베딩을 생성하는 과정

▶ 딥러닝 모델 학습과정

[그림 8]은 딥러닝 기법을 사용한 실제 번역 모델 학습과정인데 자세히 보면 Train Loss와 Test Loss 두 가지를 확인할 수 있다. 일단 loss의 정확한 정의를 말하면 학습을 통해 직접 줄이고자 하는 값인데, 이를 활용하여 모델의 학습이 잘 되고 있는지 판단하는 척도로 사용할 수 있다. 모델의 실제 번역 성능에 대한 정확도는 다른 방법을 사용하거나 번역된 결과로 판단해야겠지만 학습이 잘 되고 있는지에 대해서는 loss의 줄어드는 정도로 확인할 수 있다.

또, Train Loss란 학습 데이터로 한정해서 계산한 loss값이고 Test Loss란 학습데이터를 배제하고 학습데이터에 없는 데이터로 한정해서 계산한 loss값이라고 말할 수 있다. 따라서 Test Loss가 줄어드는 방향으로 학습하는 것이 이 모델의 성능에 좋은 영향을 끼친다

이 모델의 성능에 좋은 영향을 끼친다고 할 수 있다. 만약 Train Loss만 줄어 들고 Test Loss가 줄어들지 않는다면 overfitting(훈련 데이터에 대해 과적합)된 상태라고 볼 수 있다. 참고로 그림에 보이는 이상한 문자는 실제 학습하는 과정을 실시간으로 보여주는 진행률을 표시하는 문자이다.



그림 8. 딥러닝 모델을 적용하여 학습하는 과정

▶ 학습 후 번역 결과 구현

학습시킨 모델을 이용한 번역 결과로, 학습 데이터가 많지 않아 일반적인 데이터에 대해 성능이 잘 나오지는 않았지만 정치, 경제 분야 데이터로 학습시켰기 때문에 그림과 같은 문장에 대해서는 비교적 괜찮은 성능을 보이고 있다. 우리가 적용한 모델에서는 약 10만 문장을 활용하여 학습시켰는데, 범용적으로 번역에 쓰이기 좋은 성능을 내려면 억 단위 이상의 문장이 필요하다.

```
kor > 북한은 핵무기를 가지고 있다 .
kor > 북한은 핵무기를 가지고 있다 .
eng > north korea has a nuclear weapon .
kor > 오바마는 미국의 대통령이다 .
kor > 오바마는 미국의 대통령이다 .
eng > obama is the presid of the presid elect .
```

그림 9. 학습 후 번역 결과

프로젝트 완료 후기

멘토 : 어떻게 인공지능 프로젝트 중에서도 이 프로젝트를 선택하게 됐지?

신혁 : 평소에도 AI 수업을 들으면서 흥미를 갖게 되었고 또 과거에 자막 편집용 SW를 개발하는 아르바이트를 한 경험이 있어서 접근하기 어렵지 않았습니다.

종혁 : 학교에서 수업에서 배운 내용을 토대로 실무적인 개발을 해보고 싶어서 알아보던 중 한이음 프로젝트를 알게 되었습니다. 저와 같은 생각을 갖고 있었던 후배 두 명과 함께 알아보게 되어 지원하게 되었습니다.

현수 : 지금의 멤버들과 같이 인공지능 수업을 같이 들으면서 뜻이 맞아 해보게 되었습니다.



그림 10. 2018년 6월 오프라인 모임 모습, 좌로부터 이현수 오신혁 강자원 멘토 안종혁

멘토 : 프로젝트를 수행하면서 무엇을 느꼈어?

신혁 : 관련 분야의 지식을 얻고 경험을 쌓게 된 것이 좋았습니다. 다만 번역 결과가 기대한 성능에 미치지 못했다는 것이 아쉬웠습니다.

종혁 : 프레임워크나 개발환경 등 사용할 기술에 대한 사전조사와 설계가 많이 미흡하여 중간 수정이 많았습니다. 제대로 된 설계와 일정관리가 되었다면 프로토타입이 빨리 나와서 다양한 기능 추가와 테스트가 가능했지 않을까 아쉬움이 있습니다.

현수 : 인공지능 관련 수업에서 간단히 훑고 넘어가는 수준을 넘어 좀 더 깊이 있는 학습을 할 수 있어서 해당 분야 지식을 쌓는데 많은 도움이 되었습니다.

멘토 : 앞으로 계획은?

신혁 : 이번 프로젝트 경험을 토대로 자연어 처리 분야의 연구에 더 깊게 파고 들어볼 생각입니다. 현재도 산학연계프로젝트 일환으로 기업과 협력하여 인공지능 관련 개발을 하고 있습니다. 관련하여 더 깊은 지식을 쌓고 실무경험을 해볼 수 있을 것 같습니다.

종혁 : 이번 프로젝트를 통해 API와 통신을 담당하며 백엔드 개발에 관심이 생겼습니다. 관련 기술에 대하여 관심을 가지고 공부하는 중인데, 플랫폼 서비스에 관심을 많이 가지고 있습니다. 백엔드 개발자로서 견고하고 효율적인 API를 제공하는 플랫폼 개발자로 목표를 잡고 있습니다.

현수 : 비록 프로젝트는 끝났지만, 아직 부족한 점이 많아 보완해야 할 부분들이 많다고 생각합니다. 관련 분야에 대해 좀 더 심도 있게 공부해서 자기 계발과 더불어 성능 개선을 해나가고 싶습니다.

멘토 : 매달 오프라인 모임을 진행하면서 눈부시게 성장해나가는 너희 모습을 보며 표현은 하지 않았지만 얼마나 놀랐는지 몰라. 매달 춘천에서 서울까지 기차를 타고 오는 너희들의 시간을 허투루 쓰지 않아야지 하며 나 스스로도 더 열심히 준비했던 것 같아. 어떻게든 프로젝트를 완료해내는 너희들을 보면서 참 대견하고 기뻐했어. 특히 팀의 리더로서 종혁이의 눈부신 리더십과 책임감 있는 모습은 늘 멘토를 든든하게 했고 PM으로서 귀찮은 일들까지 모두 맡으면서 또 이 프로젝트의 핵심인 딥러닝 모델을 구현한 신혁이의 역할은 엄지척이었어. 우리의 아이돌급 페이스를 가진 현수는 이번 프로젝트 진행하면서 다치기도 하고 힘든 일도 겪었는데 책임감 있게 끝까지 해내고 또 더 완성도를 높이고 싶다는 의지에 박수를 보내고 싶어. 부족한 멘토를 믿고 끝까지 따라와 줘서 고맙고 언제나 눈부신 결과를 가져온 너희 모두에게 나 또한 감사의 말을 전하고 싶어. 덕분에 나도 많이 배웠고 성장했어. 고맙고! 언제나 응원할게~