OCR을 이용한 영상 속 문자 추출과 텍스트 분류 웹 서비스

권혁진, 박세준, 손용락, 주성진, 이상준
숭실대학교 소프트웨어학부
{tpwns6693, eoanswkgksk}@gmail.com, ssknock@daum.net, castlereal@naver.com

Web service for text extraction and classification using OCR from video

Hyeokjin Kwon, Sejun Park, Yonglak Son, Seongjin Joo, Sangjun Lee
School of Software, Soongsil University

요 약

디지털 시대로 인해 동영상 컨텐츠의 소비가 활성화 되었다. 동영상 컨텐츠의 공급이 많아짐에 따라 소비자들은 더 짧고 유익한 동영상을 선호하게 되었다. 이러한 소비자들의 요구에 맞춰 동영상을 시청하지 않고 동영상 내에 존재하는 정보를 추출하는 서비스들이 존재하지만, 기존 방식은 음성 데이터를 사용하여 잡음이 섞여 신뢰성이 낮아지거나, 텍스트 데이터를 사용하더라도 사용자가 동영상 시청과 유사한 시간을 소모해야 한다는 문제가 존재한다. 본 논문에서는 이런 한계점을 극복하기 위해 OCR을 이용한 영상 속 텍스트 추출과 텍스트 분류 웹 서비스를 제안한다. 동영상을 프레임 단위로 분석하고 OCR을 통해 화면 속 텍스트를 추출하는 방식으로, 여기에 추출된 스크립트에 대한 카테고리 분류와 번역 기능을 제공하여 접근성을 높인다. 해당 방식을 통해 사용자는 기존 방식 대비 동영상 내에 존재하는 핵심 정보를 빠르고, 정확하게 얻을 수 있다.

ABSTRACT

As the digital industry develops, the consumption of video has been activated. Consumers prefer the shorter and more useful videos as the supply of the video contents increases. According to their needs, there are existing services which extract the information from the video without watching it. However, the existing methods have problems that they use voice data which can reduce the reliability, or even if they use text data, the user has to spend the entire video's execution time. In this paper, we propose a novel web service for text extraction and classification using OCR to extract the information from video. It extracts the text from the video's frame using OCR and provides additional functions such as category classification and translation for extracted scripts to increase accessibility. Thorough this method, user can obtain the key information in the video quickly, compared to the existing method.

I. 서 론

4 차 산업혁명과 함께 찾아온 디지털 시대로 인해 검색엔진, SNS, 동영상 스트리밍 서비스 등 다양한 정보를 얻을 수 있는 정보의 출처가 증가했다. 그 중에서도 동영상 스트리밍 서비스의 대표 주자인 유튜브는 100 개가 넘는 국가에서 매월 2 억명 이상의 사용자가 이용하고 있다.[1] 유튜브를 중심으로 문화예술 콘텐츠 플랫폼의 대안으로 활용[2]하는 연구가 진행될만큼 동영상 컨텐츠를 소비하는 사용자가 늘어나고 있는 추세이다. 동영상 컨텐츠의 소비가 활성화됨에 따라 사용자들은 수 많은 동영상들 중에서 좋은 퀄리티의 동영상을 손쉽게 접하기를 원한다. “유튜브 ‘인기급상승’ 장기 노출을 위한 콘텐츠 전략에 관한 연구”[3]와 또 다른 동영상 스트리밍 서비스인 Tiktok 에 관한 “Tiktok 서비스 이용자의 몰입과 중독에 미치는 영향요인 연구”[4]에 따르면 사용자들은 10 분이 넘어가는 긴 동영상을 선호하지 않으며 특히 서비스의 사용시간이 짧은 사용자들은 짧은 동영상을 중독적으로 즐긴다는 사실을 알 수 있다.

사용자들의 이러한 니즈를 충족하기 위해 동영상을 시청하지 않고 동영상 내에 존재하는 정보를 추출할 수 있는 서비스들이 개발되고 있다. 대표적으로 Adobe Premiere Pro[5]에 동영상 파일의 음성 데이터를 분석하여 텍스트를 생성하는 기능이 추가되거나 화면에 존재하는 텍스트를 실시간으로 추출하는 Retro Arch 서비스[6]가 있다. 동영상에서 정보를 추출하기 위해 음성 데이터를 사용하는 경우 관련없는 외부 소리로 인해 음성 데이터의 신뢰성이 낮아지는 문제가 존재하고 텍스트 데이터를 사용하는 경우 동영상을 재생해서 화면을 인식해야하기 때문에 사용자가 동영상 시청과 유사한 시간을 소모해야 한다는 문제가 존재한다.

본 논문에서는 OCR 을 이용한 영상 속 텍스트 추출과 텍스트 분류 웹 서비스를 제안한다. 제안 서비스는 동영상을 재생하지 않고 동영상 내에 존재하는 정보를 추출하는 서비스이다. 동영상을 프레임 단위로 분석하여 OCR 을 통해 동영상 화면 속에 존재하는 텍스트를 인식한다. 인식한 텍스트를 기반으로 스크립트를 생성하고 전체 스크립트와 해당 내용을 분류한 키워드를 사용자에게 제공한다. 또한 해외 영상의 경우 번역 기능을 제공하여 해외 영상에 대한 접근성을 높일 수 있다는 장점을 가지고 있다. 제안 서비스를 통해 사용자는 동영상을 시청하지 않고 동영상 내에 존재하는 정보를 빠르고 정확하게 얻을 수 있다.

본 논문의 구성은 다음과 같다. 2 절에서는 제안 서비스를 구현하기 위해 사용한 관련 연구에 대해 살펴보고 3 절에서는 제안 서비스의 설계 및 구현에 대해 설명한다. 4 절에서는 동영상 내에 존재하는 정보를 추출하는 기존 서비스와 제안 서비스를 비교 및 분석한다. 마지막으로 5 절에서는 본 논문의 결론을 맺는다.

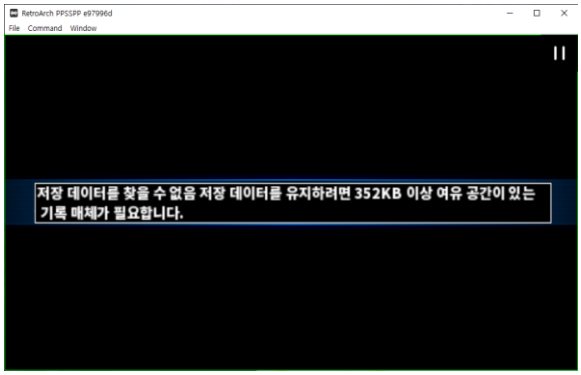
II. 관련 연구

2.1 RetroArch

다양한 비디오 게임들을 플레이 할 수 있도록 에뮬레이터를 제공하는 오픈소스 프로그램인 Retro Arch [6] 는 2019 년 8 월 출시된 1.7.8 버전부터 화면의 언어를 인식해서 자국 언어에 맞게 번역하여 화면에 띄우거나 이를 음성으로 제공하는 AI Service 를 지원하고 있다 [7].

해당 기능은 버튼을 누르면 게임 화면이 일시정지 된 후 게임 화면을 OCR 하여 문자를 검출해 기계번역 하여 사용자에게 음성 혹은 문자로 제공한다. 사용자는 Ztranslate[8]나 RetroArch-AI-with-IoTEdge[9] 서비스를 이용하여 해당 기능을 손쉽게 사용할 수 있다. [그림 1]은 Ztranslate 서비스를 이용한 AI Service 의 실제 동작 모습이다. [그림 2]과 같이 사용자는 AI Service 를 사용하여 익숙하지 않은 언어로 만들어진 게임일지라도 큰 어려움 없이 플레이 할 수 있다.

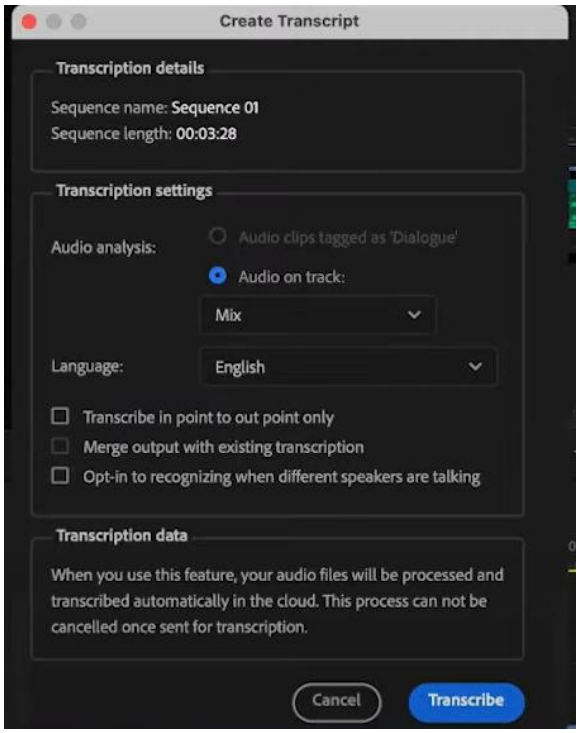
[그림 1] AI Service 를 사용하지 않은 모습



[그림 2] AI Service 를 사용한 결과

2.2 Adobe Premiere Pro

Adobe 의 동영상 편집 프로그램인 Adobe Premiere Pro[5]는 15.4 버전 업데이트 이후로 음성을 텍스트로 변환하는 기능이 추가되었다. 해당 기능은 입력으로 주어진 동영상 파일의 음성 파일을 분석하여 캡션(자막)을 생성하는 기능이다.



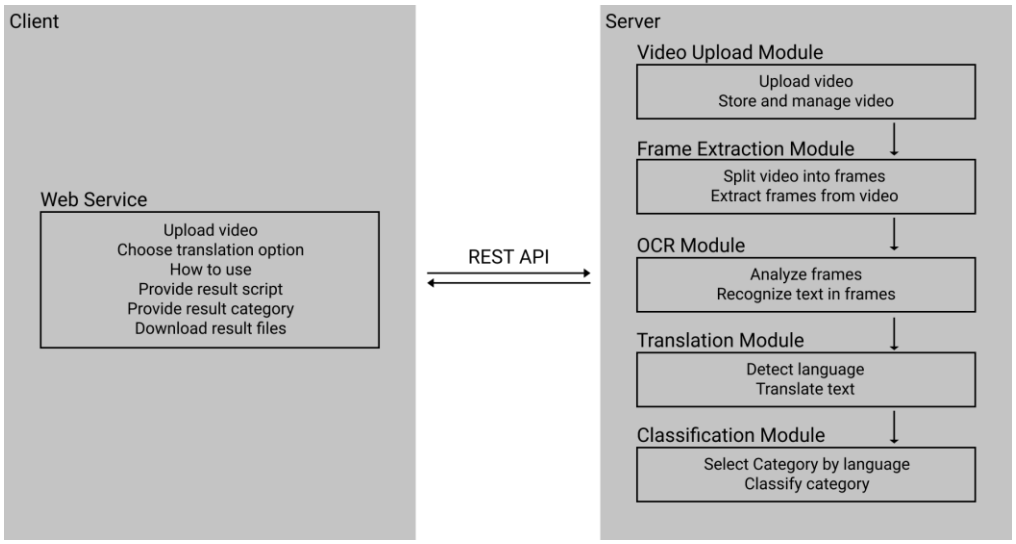
[그림 3] 음성을 자막으로 변환하기 위한 생성 창

[그림 4] 타임라인에 자막이 생겨난 모습

해당 기능이 추가되기 이전에는 동영상을 하나하나 확인하며 동영상 내에 자막을 달거나, 편집 프로그램 외에 별도의 음성 인식 프로그램을 사용하였으나 이제는 편집 프로그램에서 기능을 자체적으로 제공하게 된 것이다.

이 기능은 주어진 동영상 파일을 인공지능 기술을 사용하여 스크립트를 생성하고 Adobe Sensei 머신 러닝을 활용하여 타임라인에 자막을 배치하여 음성과 자막의 속도를 일치시킨다. 또, 인터넷 연결 없이도 기능을 용할 수 있게 하는 언어 팩을 지원하고 있다.

[그림 3]은 해당 기능을 사용하기 위해 사전 설정 창이다. 번역하려는 언어와 음성에 대해 설정해주면 된다.



[그림 5] 제안된 프로그램 구조

[그림 4]는 해당 기능을 사용하여 타임라인에 맞게 자막이 생성된 모습이다.

III. 제안 프로그램

본 절에서는 OCR을 이용한 영상 속 텍스트 추출과 텍스트 분류 웹 서비스의 구조와 구현 방법에 대해서 설명한다. 제안 프로그램은 OCR을 이용한 영상 속 텍스트 추출과 텍스트 분류 웹 서비스로 동영상을 재생하지 않고 동영상 내에 존재하는 정보를 추출하는 서비스이다. 서버는 Java 플랫폼을 위한 오픈 소스 애플리케이션 프레임워크인 Spring 프레임워크를 사용했고 클라이언트 실행 환경으로는 Chrome 브라우저를 사용했다.

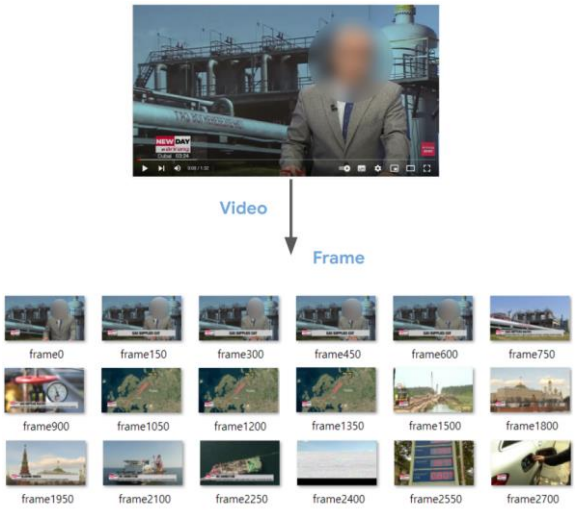
3.1 제안 프로그램 구조

제안된 프로그램은 [그림 5]와 같이 크게 웹 클라이언트와 서버로 구성되어 있다. 웹 클라이언트는 제안된 프로그램을 사용하는 사용자가 정보를 추출하고 싶은 동영상을 업로드하고 동영상에서 추출한 정보를 얻을 수 있는 UI/UX를 제공한다. 서버는 웹 클라이언트로부터 전달받은 동영상을 가공하여 동영상 내에 존재하는 정보를 추출하고 결과를 웹 클라이언트에게 전달한다.

또한 [그림 5]는 제안된 프로그램의 전체 구조를 보여주고 있다. 사용자가 웹 클라이언트를 통해 동영상을 업로드하면 웹 클라이언트는 HTTP 기반의 REST API를 사용하여 서버와 통신한다. 웹 클라이언트가 요청한 API에 따라 서버는 각각 다른 모듈을 실행한다. 모듈은 총 5개로 구성되어 있으며 동영상 업로드 모듈, 프레임 추출 모듈, 텍스트 추출 모듈, 번역 모듈, 텍스트 분류 모듈이 존재한다. 모듈에 대한 상세한 내용은 3.1.1 ~ 3.1.5를 통해 설명한다.

3.1.1 동영상 업로드 모듈

동영상 업로드 모듈은 사용자가 웹 클라이언트를 통해 정보를 추출하고 싶은 동영상을 업로드할 때 실행되는 모듈이며 사용자가 업로드한 동영상에 대한 모든 처리가 끝날 때까지 동영상을 관리하는 모듈이다. 사용자가 업로드한 동영상을 서버의 로컬 디스크에 임시로 저장한다. 이때 동영상별로 고유한 ID 값을 부여해서 사용자가 업로드한 동영상 파일이 덮어써지는 일이 없도록 한다. 이후에 서버에서 발생하는 모든 작업에서



[그림 6] 프레임 추출 모듈 결과 예시[19]

동영상을 사용할 때 동영상별로 부여된 고유한 ID 값을 기준으로 동영상을 구분해서 사용한다. 동영상에서 정보를 추출한 후 결과를 사용자에게 전송하고 나면 해당 동영상은 더 이상 서버에서 사용하지 않으므로 로컬 디스크에서 제거한다.

3.1.2 프레임 추출 모듈

프레임 추출 모듈은 사용자가 업로드한 동영상 파일을 프레임 단위로 분리하여 추출하는 모듈이다. [그림 6]은 해외 뉴스 동영상에서 프레임을 추출한 예시이다. 150 프레임 단위로 프레임을 추출했다. 총 2778 프레임의 영상이고 18 개의 프레임이 추출된 것을 확인할 수 있다. 프레임 추출 모듈은 사용자가 업로드한 동영상 파일의 크기가 클수록 모듈의 실행 시간이 오래걸리므로 8 개의 스레드를 사용하는 멀티 스레드 환경을 구축했다. 사용자가 업로드한 동영상의 전체 프레임 수를 계산하고 8 개의 스레드에게 균등하게 분배하여 모듈 실행 시간을 최대한 단축했다. 추출된 프레임은 서버의 로컬 디스크에 임시로 저장되며 동영상 업로드 모듈과 마찬가지로 더 이상 서버에서 사용하지 않는 시점에 로컬 디스크에서 제거된다.

	동영상 길이(초)	프레임 수	실행 시간(초)
1	92	2778	2.534
2	117	3516	3.506
3	132	3958	3.954
4	132	3967	4.157
5	266	7982	5.111

[표 1] 프레임 추출 실험 결과

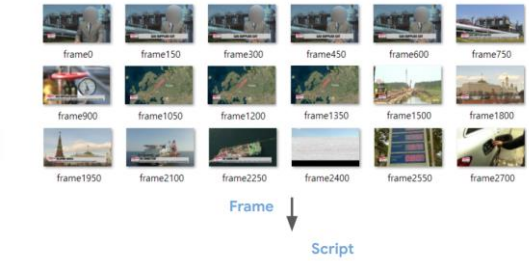
추출된 프레임은 동영상별로 가진 고유한 ID 값을 동일하게 사용해서 어떤 영상의 프레임인지 구분할 수 있다. [표 1]은 프레임 추출 모듈을 사용해서 프레임을 추출할 때 소요되는 모듈 실행 시간을 시각화한 자료이다. 동영상 길이가 길어질 수록 모듈 실행 시간이 증가하는 것을 확인할 수 있다.

3.1.3 텍스트 추출 모듈

텍스트 추출 모듈은 프레임 추출 모듈의 결과로 추출된 프레임을 분석하여 프레임 내에 존재하는 텍스트를 인식하고 스크립트 형태로 제공하는 모듈이다. [그림 7]은 해외 뉴스 동영상의 프레임에서 텍스트를 추출한 결과 예시이다. 프레임 추출 모듈 결과 예시와 동일한 동영상을 예시로 사용했다. 텍스트 추출 모듈의 결과로 영어 스크립트를 확인할 수 있다. 프레임 추출 모듈이 추출한 프레임의 순서대로 OCR 모델을 사용하여 프레임 내에 존재하는 텍스트를 인식하여 추출한다. 추출한 텍스트를 프레임 순서에 맞게 스크립트 형태로 저장한다. OCR 모델은 Clova ai 와 Kakao brain 의 PORORO 를 활용하여 구현한 OCR 모델을 사용한다. OCR 모델에 대한 자세한 설명은 3.2.1 에서 설명한다.

3.1.4 번역 모듈

번역 모듈은 텍스트 추출 모듈의 결과 스크립트를 번역하는 모듈이다. 사용자가



"gas supplies cut
gas supplies halted russia halts flow of gas supplies to europe sending prices surging
blaming russia western experts accuse russia of slashing gas supplies
no connection kremlin spokesperson insists supply halt is purely commercial
situation
autogasl lpg"

[그림 7] 텍스트 추출 모듈 결과 예시[19]

동영상을 업로드할 때 번역 여부를 선택할 수 있다. 사용자의 선택에 따라 사용자가 번역을 원할 경우 번역된 스크립트를 제공한다. 번역은 외부 API 인 파파고 API[10]를 사용한다. 텍스트 추출 모듈의 결과 스크립트를 파파고 API 의 언어 감지 API 를 사용하여 어떤 언어인지 감지한다. 서버는 감지된 언어를 번역할 수 있는지 확인한 후 번역할 수 있다면 파파고 API 의 번역 API 를 사용하여 텍스트 추출 모듈의 결과 스크립트를 한국어로 번역하고 번역된 스크립트를 사용자에게 제공한다.

3.1.5 텍스트 분류 모듈

텍스트 분류 모듈은 텍스트 추출 모듈의 결과 스크립트를 카테고리로 분류하여 키워드를 제공하는 모듈이다. 카테고리는 한글과 영어별로 다르게 관리되며 20 개 이상의 카테고리가 존재한다. 외부 API 인 파파고 API 의 언어 감지 API 를 사용해서 한글 스크립트일 경우 한글 카테고리 분류 모델을 사용하고 영어 스크립트일 경우 영어 카테고리 분류 모델을 사용한다. 현재는 한글과 영어 2 가지 언어만 지원한다. 텍스트 분류 모듈은 KoBERT 를 통한 분류와 BERT 를 통한 분류를 사용하며 자세한 설명은 3.2.2 에서 설명한다.

"gas supplies cut
gas supplies halted russia halts flow of gas supplies to europe sending prices surging
blaming russia western experts accuse russia of slashing gas supplies
no connection kremlin spokesperson insists supply halt is purely commercial
situation
autogasl lpg"

Script
Keywords

"World News"

[그림 8] 텍스트 분류 모듈 결과 예시

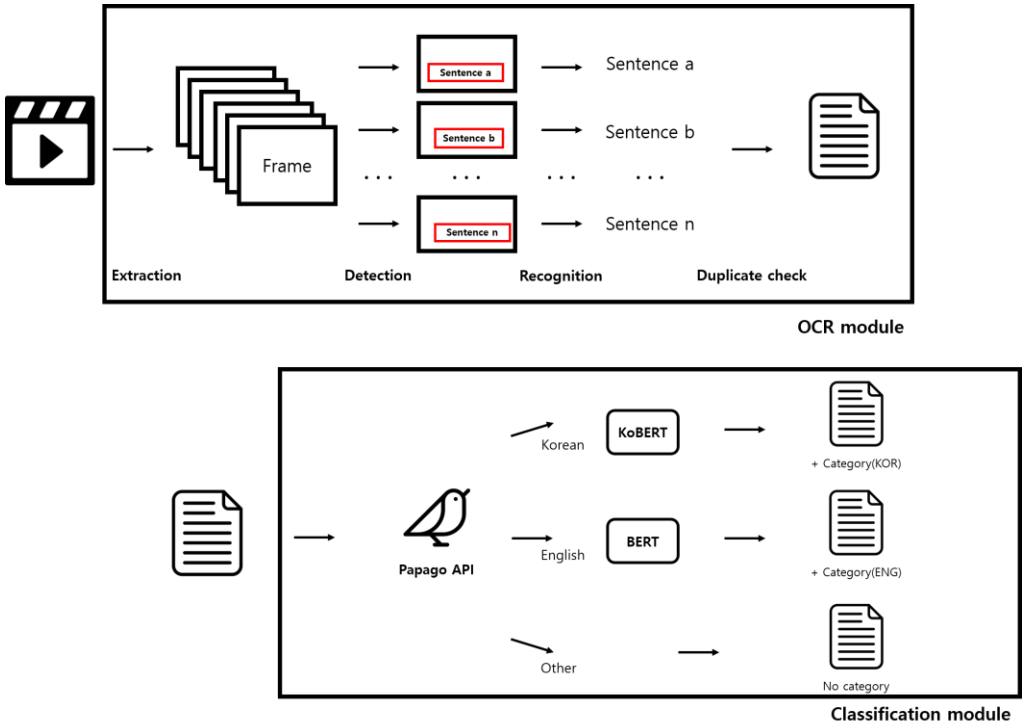
[그림 8]은 해외 뉴스 동영상에서 텍스트를 추출한 스크립트를 기반으로 텍스트 분류 모듈을 실행한 결과 예시이다. 텍스트 추출 모듈 결과 예시와 동일한 동영상을 예시로 사용했다. 텍스트 분류의 결과로 "World News" 카테고리가 선택된 것을 확인할 수 있다.

3.2 동영상 속 정보 추출

제안된 프로그램은 OCR 을 통해 영상에서 텍스트를 추출하고, 중복을 제거하여 영상을 하나의 스크립트로 만든 뒤 카테고리 분류 작업을 통해 추출된 스크립트의 키워드를 제공하는 방식으로 영상에서 정보를 추출한다. 텍스트를 통해 정보를 제공하기에 자막이 핵심 내용을 전달하는 뉴스 영상에 초점을 두어 해당 모듈을 개발했다. 3.2 에서는 제안된 프로그램이 동영상 속에서 정보를 추출하는 방식에 대한 자세한 설명과, 구현 방식에 대해서 다룰 것이다. 정보를 추출하는 부분의 전체적인 구조 및 흐름은 [그림 9]에서 확인할 수 있다. 제안된 프로그램은 영어와 한글에 대해 텍스트 추출 및 분류 기능을 제공한다.

3.2.1 OCR 을 통한 영상 속 텍스트 추출

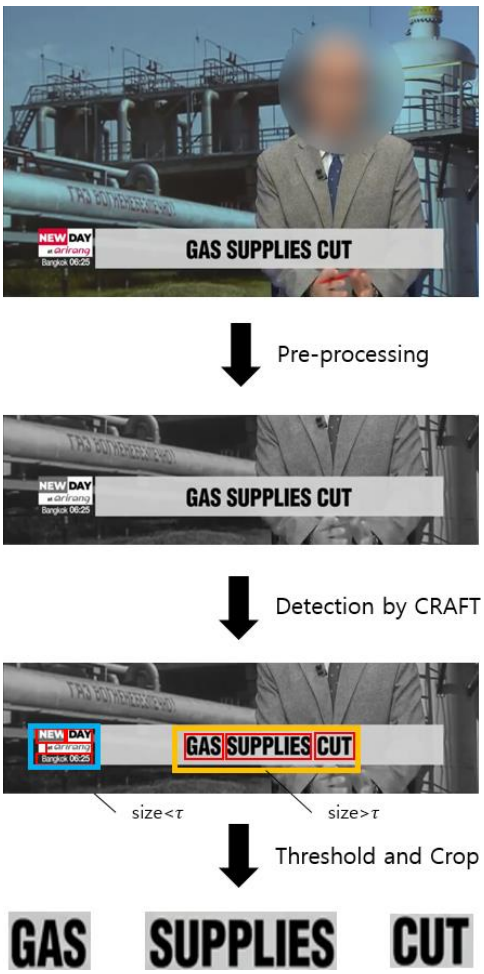
OCR 을 통해 영상 속에서 텍스트를 추출하는 것은 크게 detection, recognition, 중복 제거 총 3 단계로 나뉜다. 각 부분에 대한 구현 방식을 살펴보자면 다음과 같다.



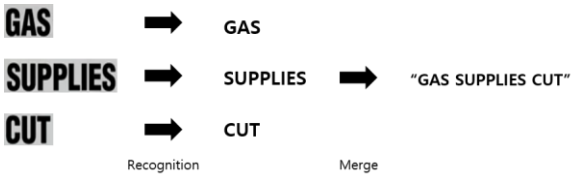
[그림 9] 동영상 속 정보 추출의 전체적인 구조 및 흐름

3.2.1.1 Detection

영상 속에서 텍스트를 추출하기 위해선, 먼저 추출된 프레임에서 글자를 감지하는 작업이 선행되어야 한다. 뉴스 영상에서 주요 헤드라인 자막은 영상의 하단에 나오는 점에서 착안하여 입력된 프레임을 절반으로 잘라 하단 부분을 사용했고, 원활한 인식을 위해 컬러 영상을 회색 조로 바꾼 뒤 detection 을 진행했다. Detection 은 Clova ai 의 CRAFT[11]를 사용했으며 detection 결과는 단어 단위의 bounding box 가 좌표 형태로 나오게 된다. 너무 작은 문자까지 감지한다면 불필요한 정보가 다수 포함될 수 있기에 특정 임계값을 두어 해당 값보다 크기가 작은 문자는 감지 결과에서 제외하는 방식을 사용했다. Detection 단계를 거치면 프레임은 감지된 단어의 bounding box 좌표를 통해 단어 단위로 잘라져 저장되고, 이렇게 저장된 단어 단위의 사진은 인식 단계로 넘어가 실제 글자로 변환된다. [그림 10]은 영어 뉴스 영상을 예시로 하여



[그림 10] Detection 과정 예시[19]



[그림 11] Recognition 과정 예시

CRAFT 를 사용한 detection 의 전반적인 과정을 보인다.

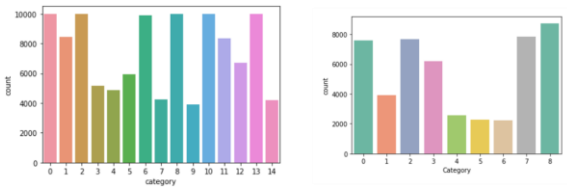
3.2.1.2 Recognition

프레임에서 성공적으로 단어가 감지되었으면, 해당 단어 사진이 실제로 어떤 단어를 나타내는지 인식하는 과정이 필요하다. 개발 초기 단계에서는 Clova ai 의 deep text recognition benchmark[12]에서 제시한 TRBA (TPS-ResNet-BiLSTM-Attn)모델을 사용했지만, 제공되는 모델은 영어 글자로만 학습되어 한글 인식이 불가능했고, 한글 인식을 위해선 한글 데이터를 사용하여 모델을 다시 학습할 필요가 있었다. 하지만 새롭게 학습하는 과정에서 영문 글자의 조합 대비 복잡한 한글 글자의 조합과, 데이터의 부족으로 인해 원하는 정확도 달성에 실패했다. 따라서 한글과 영어를 모두 지원하는 통합 자연어 처리 프레임워크인 Kakao brain 의 PORORO[13]의 OCR 기능을 사용하여 recognition 부분을 구현하였다.

Recognition 단계에서는 입력으로 프레임에서 감지된 단어 사진들이 들어오면, 해당 단어 사진에 대해 순서대로 인식을 진행하고 이를 한 문장으로 만든다. Detection 과 recognition 단계를 거치면, 영상 속 프레임은 문장으로 변환된다. [그림 11]은 예시를 사용하여 recognition 의 전반적인 과정을 보인다.

3.2.1.3 중복 제거

한 영상은 많은 프레임들로 구성되어 있기에 같은 장면으로 보여도 다수의 프레임으로



[그림 12] 학습 데이터 분포도. 왼쪽 그림은 영어 뉴스 카테고리 학습 데이터[14]이고 오른쪽 그림은 한국어 뉴스 카테고리 학습 데이터[15]이다.

구성되고, 이로 인해 텍스트를 추출하는 과정에서 중복되는 문장이 다수 발생할 수 있다. 프레임에서 추출된 문장들을 모아 스크립트로 만드는 과정에서 이런 중복되는 문장들은 제거 되어야 하고, 이를 위해 N-gram 분석을 사용한다. 프레임에서 추출된 문장이 새롭게 입력되면 N-gram 단위로 문장을 토큰화하고 이를 직전에 입력되었던 문장을 마찬가지로 N-gram 으로 토큰화한 결과와 비교한다. 비교 결과 80% 이상 중복된다고 판단하면 해당 문장은 중복된 문장으로 판단하여 버리고, 그렇지 않으면 스크립트에 새롭게 검출된 문장을 추가한다. 이런 중복 제거 과정을 거치면 영상은 최종적으로 하나의 스크립트로 압축된다.

3.2.2 BERT 를 통한 카테고리 분류

OCR 을 통해 성공적으로 영상을 하나의 스크립트로 압축하는데 성공하면 카테고리 분류를 통해 사용자에게 해당 스크립트의 키워드를 제공한다. 앞에서도 언급했듯이, 제안된 프로그램이 뉴스 영상에 초점을 두고 구현되었기에 뉴스 카테고리 분류를 진행하였으며 한글과 영어 문장을 지원한다.

카테고리 분류는 Google 의 pre-trained 된 BERT 를 fine-tuning 하여 구현했다. BERT 는 많은 자연어 처리 분야에서 좋은 성능을 내고, 다른 task 에서 추가 데이터를 이용한 training 을

[그림 13] 카테고리 분류 예시

진행하여 hyper-parameter 를 재 조정하는 fine-tuning 에 좋은 성능을 보인다[16]. 다만 한국어 스크립트의 카테고리 분류는 BERT 가 지닌 한국어 성능의 한계 때문에 SKT brain 의 KoBERT 를 fine-tuning 하여 구현했다. KoBERT 는 수백만 개의 한국어 말뭉치를 학습하여 한국어 관련 task 에 좋은 성능을 보인다[17]. 학습 시 사용하는 데이터의 부족과 하드웨어 자원의 한계로 높은 정확도를 얻는 것은 힘들다고 판단했고, 카테고리 분류 기능 구현이 가능함을 확인하기 위해 0.8 이상의 정확도를 얻으면 유효하다고 판단했다.

[그림 9]의 분류 모델 부분에서 확인할 수 있듯이 스크립트가 입력되면 Papago API[10]의 언어 감지 기능을 통해 해당 스크립트가 어느 언어로 구성되어 있는지 판단하고, 영어인 경우에는 BERT 를 fine-tuning 한 모델을 통해 분류를 진행하고, 한글인 경우에는 KoBERT 를 fine-tuning 한 모델을 통해 분류를 진행하고, 스크립트와 키워드를 함께 사용자에게 보인다. 영어와 한국어가 모두 아닌 다른 언어라면 카테고리 분류를 진행하지 않고 스크립트만 사용자에게 보인다.

3.2.2.1 BERT fine-tuning

BERT-uncased-base 를 뉴스 카테고리 분류 task 에 적합하도록 fine tuning 을 진행하여 분류 모델을 구현했다. [그림 12]의 왼쪽 부분은 영어 뉴스 카테고리 학습 데이터의 분포를 나타낸다. 데이터 원본 자체가 불균형하고 너무 많은 카테고리를 포함하기에 비슷한 카테고리를 통합하고 레이블 당 최대 데이터 개수를 10000 개로 강제하는 데이터 전처리 과정을 먼저 수행한 뒤 학습을 진행했다. 데이터의 개수는 약 13 만개이며 15 개의 레이블이 존재한다. 훈련 데이터와 검증 데이터와 시험 데이터의 비율은 7:2:1 로 두었다. 학습은 10 epoch 를 수행했고, 학습 결과 0.81 의 정확도를 얻을 수 있었다. 너무 많은 레이블과 데이터의 불균형으로 인해 높은 정확도를 얻지는 못했지만, 이는 차후 더 나은 데이터를 구할 수 있으면 해결될 것으로 판단된다. 학습된 모델을 통해 OCR 결과에 대해 분류를 진행하면 [그림 13]의 위 부분과 같다.

3.2.2.2 KoBERT fine-tuning

한글의 경우 KoBERT 를 fine tuning 하여 분류 모델을 구현했다. [그림 12]의 오른쪽 부분은 한국어 뉴스 카테고리 학습 데이터 분포를 나타낸다. [그림 12][에서 4, 5, 6 카테고리는 데이터의 개수가 굉장히 작아 보이지만, 각각 해외 축구, 해외 야구, 국내 야구를 나타내기에 학습 시에는 스포츠라는 하나의 레이블로 통합되어 사용되어 다른 레이블과 데이터의 개수가 비슷하다. 네이버 뉴스를 크롤링 하여 데이터를 수집했으며, 데이터의 개수는 약 5 만 6 천개이며 7 개의 레이블이 존재한다. 훈련 데이터와 검증 데이터와 시험 데이터의 비율은 7:2:1 이고, 10 epoch 의 학습을 수행했으며 0.85 의 정확도를 얻을 수 있었다. 데이터의 불균형이 해결되었기에 영어 카테고리 분류에 비해 정확도는 상승했지만, 여전히 하드웨어 자원의 한계로 크롤링한 데이터의 기간이 길지 않아 데이터가 특정 기간에 몰려 편향 되어 있고 개수 자체도 많지 않아, 높은 정확도를 얻지는 못했다. 학습된 모델을 통해 실제로

OCR 결과에 대해 분류를 진행하면 [그림 13]의 아래 부분과 같다.

3.3 클라이언트와 서버의 통신

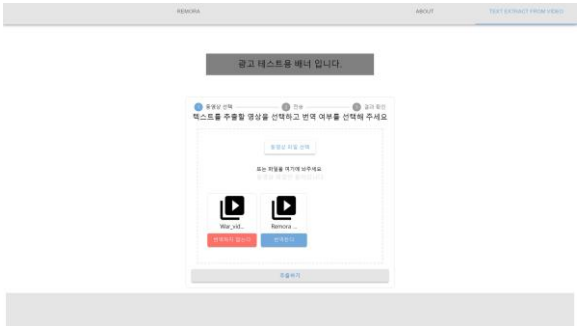
제안된 프로그램에서 서버는 REST API 형태로 이루어져 있다. REST(Representational State Transfer)란 웹과 같은 하이퍼미디어 프로토콜에 대한 표준으로, 몇 가지의 제약을 준수하여 네트워크 자원을 쉽게 주고받기 위한 표준이다. REST 에는 많은 제약 조건이 있지만 반드시 지켜야 할 제약 조건은 다음과 같은 것들이 있다. [18]

- Client-Server : 사용자 인터페이스와 데이터 저장을 분리하는 제약으로써 클라이언트의 이식성과 서버의 규모 확장성을 개선하여 둘 다 독립적으로 진화할 수 있어야 함을 의미한다.

- Stateless : 클라이언트와 서버 간의 상태가 존재하지 않아야 한다는 제약으로, 클라이언트와 서버의 독립적인 구조를 위한 제약이기도 하다. 모든 요청에 대해 그 요청을 처리하기 위해 필요한 데이터를 모두 담아야 한다는 제약인데, 이를 통해 Task 실패에 대한 복원이 쉽고, 상태를 저장하기 위한 소모가 없으므로 서버나 클라이언트의 규모 확장성이 개선된다.

- Cache : 중복되는 데이터를 반복적으로 보내는 것은 네트워크 성능 문제를 야기할 수 있으므로, 임시적으로 중복되는 데이터를 저장할 수 있어야 한다는 것이다. 요청에 대한 응답은 캐시가 가능한 지 혹은 불가능한 지를 명시해야 한다.

제안된 프로그램의 서버는 위와 같은 REST 를 준수하는 REST API 형태로 클라이언트와 통신하고 있으며, 클라이언트에



[그림 14]번역 여부 선택 및 업로드 화면

[그림 15] 결과 화면

[그림 16] 결과 상세 화면

데이터를 제공하기 위해 다음과 같은 API 들이 존재한다.

- Post /upload : 요청 데이터로 스크립트 추출에 사용할 동영상과 번역 여부를 하나의 쌍으로 받아서 서버 내에 저장한다. 서버는 클라이언트로부터 동영상을 전달받으면 요청한 동영상을 식별하기 위한 Code 값을 생성하고, 그 값을 결과 데이터로 넣어 클라이언트에 반환한다.

- Post /frames : 요청 데이터로 upload API 에서 반환받은 Code 를 넣으면, 서버에서는 해당 값과 일치한 동영상을 찾아 프레임 추출 작업을 수행한다. 추출된 프레임은 서버에

일시적으로 저장하고, String 배열 형태의 Set 으로 클라이언트에게 반환한다.

- Post /text : 요청 데이터로 frames API 에서 반환받은 Frame set 을 넣으면, 서버에서는 해당 Frame Set 을 토대로 저장한 Frame 이미지의 텍스트를 OCR 모듈을 사용하여 추출한다. 그 결과로 추출한 텍스트를 클라이언트에게 반환한다.

- Post /text/translated : 요청 데이터로 text API 에서 반환받은 String 을 넣으면, 서버에서는 해당 텍스트를 Papago API 를 사용하여 번역하고, 그 결과를 클라이언트에 반환한다.

- Post /keyword : 요청 데이터로 text API 에서 반환받은 String 을 넣으면, 서버에서는 해당 텍스트를 서버 내의 모듈을 통하여 키워드 분류 작업을 수행하여 그 결과를 List 형태의 String 으로 반환한다.

현재 서버의 API 들을 보면, 결과물 스크립트를 얻기 위해 upload API 를 가장 처음 호출해야 한다. upload API 를 호출한 다음에는 반환받은 데이터를 요청 데이터로 하여 frames API 를 호출하고, 또 그 반환 데이터를 요청 데이터로 하여 text API 를 호출한다. 이와 같은 Chaining 형태로 keyword API 를 수행하여 클라이언트는 최종적으로 동영상에 대한 스크립트와 분류된 키워드 데이터를 얻을 수 있게 된다.

3.4 UI/UX

[그림 14] 은 동영상의 번역 여부를 선택하는 업로드 페이지의 화면이다. 해당 화면을 통해 서버로 데이터를 전송하고 서버에서 OCR 및 번역을 진행하여 [그림 15]과 같은 결과 화면을 얻는다. 각 블록들을 클릭하면[그림 16]와 같이 상세한 사항을 제공받게 되고 클립보드로의 복사, 파일로의 다운로드 등의 기능을 사용할 수 있다.

구분	Remora	RetroArch	Adobe Premiere Pro
사용하는 정보	동영상	게임 속 이미지	음성
추출에 걸리는 시간	동영상의 길이보다 짧거나 같음	사용자가 게임을 중간 중간 일시정지 시킨 후 추출을 진행해야 함	동영상의 길이보다 길거나 같음
추출 방법	동영상 속 자막을 검출해 프레임 셋을 만들고 OCR을 진행	사용자가 OCR 및 번역을 진행 할 게임 프레임에 게임을 일시정지 시키고 OCR을 진행	동영상 속 음성을 인식해 스크립팅을 진행
모바일 지원 여부	O	O	X
결과물을 파일로 저장 가능 여부	O	X	O
클라이언트 형태	Web	Application	Application

[표 2] 제안 시스템과 기존 시스템의 비교 및 분석

IV. 기존 시스템과의 비교 및 분석

우리가 제안하는 서비스 Remora 와 기존에 서비스 중인 RetroArch 의 AI Service, Premiere Pro 의 Speech to Text 의 주요 흐름을 살피고 몇 가지 기준에 따라 비교했다. [표 2]는 각 서비스들의 공통점 및 차이점을 비교한 결과다. 각 서비스의 목적 및 기능의 주요 흐름은 다음과 같다.

Remora 는 자막이 있는 영상 속 자막 검출 및 번역을 목적하는 서비스로 사용자가 자막이 존재하는 동영상과 번역 여부를 서버로 전송하면 해당 동영상에서 프레임을 추출해 각 프레임으로부터 문자를 추출하고 필요 여부에 따라 번역한 후 추출된 스크립트의 범주를 만들어 스크립트와 번주를 사용자에게 함께 제공한다.

RetroArch 는 게임 화면 속 글자를 인식, 번역을 목적하는 서비스로 게임을 실행하는 도중에 번역이 필요한 부분에서 사용자가 멈추면 화면을 입력으로 서버에 전송하여 번역을 진행한 후에 번역된 결과를 다시 게임 화면에 오버레이 하는 방식이다.

Adobe Premiere Pro 는 동영상 속 음성을 인식해 자막을 자동으로 달아주는 것을 목적하는 서비스로 음성이 있는 동영상을 넣으면 해당 음성을 서버에 전송하고, 서버에서는 머신러닝을 통해 자막으로 사용할 수 있는 텍스트를 반환하여 타임라인에 맞게 알아서 배치해 준다.

3 개의 서비스는 모두 동영상으로부터 데이터를 추출하는 서비스라는 공통점이 존재하지만, 우리가 제안하는 Remora 와 동일한 목적을 가지고 방법을 취하지는 않는다. 우리가 제안하는 Remora 서비스와 목적 및 방법론이 정확히 일치하는 서비스가 존재하지 않았기 때문에 비슷한 범주에 속하는 동영상으로부터 데이터를 추출하는 두 서비스와 비교를 진행하였다. 동영상으로부터 데이터를 추출하는 방법에 있어서 Remora 가 가지는 차별점은 다음과 같다. Remora 는 동영상을 입력 받아 글자가 있는 프레임을 얻고 글자를 검출 및 번역하는 프로세스를 가져 사용자가 그때 그때 게임을 정지 시켜야 하는 retroArch 의 AI Service 나 동영상보다 긴 시간이 필요한 Premiere Pro 의 Speech to Text 서비스보다 적은 수행 시간을 소요한다. 또한 다른 서비스와 달리 가벼운 web 페이지의 형태로 서비스를 제공하기 때문에 소프트웨어를 설치할 필요가 없으며 retroArch 의 AI Sevice 와는 달리 결과물을 파일의 형태로 저장할 수 있다.

V. 결 론

본 논문에서는 OCR 을 이용한 영상 속 텍스트 추출과 텍스트 분류 웹 서비스를 설계하고 구현했다. 제안된 서비스를 이용하는 사용자는 웹 페이지에 접속해서 정보를 추출하고 싶은 동영상을 업로드하는 것만으로 동영상 내에 존재하는 정보를 손쉽게 얻을 수 있다. 제안된 서비스는 동영상에서 정보를 추출하기 위한 수단으로 동영상 내에 존재하는 텍스트를 사용했다. 사용자가 전달한 동영상을 프레임 단위로 분할하고 프레임마다 OCR 을 사용해서 동영상 내에 존재하는 텍스트를 추출했다. 추가로 추출한 텍스트를 기반으로 카테고리 분류를 진행해서 사용자에게 전체 정보에 대한 키워드도 제공했다. 제안된 서비스는 동영상에서 정보를 추출하는 기존 서비스와 비교했을 때 훨씬 적은 수행 시간을 요구한다는 점과 소프트웨어 설치 없이 웹

페이지를 통해 서비스를 사용할 수 있다는 점이 장점이다. 또한 동영상에서 추출한 정보에 대해 키워드를 제공한다는 차별점이 존재한다. 본 논문에서 구현한 서비스는 동영상에서 정보를 추출하는 서비스로써 실용적으로 사용될 수 있으며 동영상에서 정보를 추출하는 서비스의 확장에 긍정적인 역할을 할 것으로 기대된다.

참고 문헌

[1]YouTube Official Blog, https://blog.youtube/press/

[2]김수정. “문화예술 콘텐츠 플랫폼의 대안적 활용 방안 : 유튜브를 중심으로”. 부산대학교, 2021.

[3]이민영, 변국도, 최상현. “유튜브 ‘인기급상승’ 장기 노출을 위한 콘텐츠 전략에 관한 연구”. 한국융합학회논문지, 04/30/2022, Vol. 13, Issue 4, p. 359-372.

[4]주이모, 이상호. “Tiktok 서비스 이용자의 몰입과 중독에 미치는 영향요인 연구”. 한국융합학회논문지, 12(3), pp.125-132 Mar, 2021.

[5]Adobe Premiere Pro - speech to text, https://helpx.adobe.com/kr/premiere-pro/using/speech-to-text.html

[6]RetroArch 공식 사이트, https://www.retroarch.com/

[7] RetroArch AI Service ,libretro Docs, https://docs.libretro.com/guides/ai-service/

[8]Ztranslate 공식 사이트, https://ztranslate.net/

[9]″RetroArch-AI-with-IoTEdge”,github, https://github.com/toolboc/RetroArch-AI-with-IoTEdge

[10]Naver Papago API developers document, https://developers.naver.com/docs/papago/README.md

[11]Baek, Youngmin, et al. "Character region awareness for text detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[12]Baek, Jeonghun, et al. "What is wrong with scene text recognition model comparisons? dataset and model analysis." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

[13] Kakao brain PORORO: Platform Of neuRal mOdels for natuRal language prOcessing github, https://github.com/kakaobrain/pororo

[14] News Category Dataset, https://www.kaggle.com/datasets/rmisra/news-category-dataset

[15]네이버 뉴스, https://news.naver.com

[16]Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL.

[17]SKT Brain KoBERT github, https://github.com/SKTBrain/KoBERT

[18] Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.

[19] Arirang News, “Russia halts flow of gas supplies to Europe, sending prices surging”, https://youtu.be/rSzb2FMOvsI