(서식 6)

**졸 업 논 문**

제 목 : 머신러닝을 이용한 개인 방송 하이라이트 자동 추출 시스템

지 도 교 수 : 선 지 웅 교수님

2021년 10월 25일

한 국 외 국 어 대 학 교 공과 대 학 산업경영공학 과

학 번 : 201602843 성 명 : \_\_임유비

학 번 : 201603742 성 명 : \_\_허준호

학 번 : 201603149 성 명 : \_\_정재건

학 번 : 201800969 성 명 : \_\_김유정

**머신러닝을 이용한 개인 방송 하이라이트 자동 추출 시스템**

**Automated Extraction System of Personal Broadcast Highlights Using Machine Learning**

The point of personal broadcast highlight editing is to reduce the total time and cost spent to summarize and edit 5 to 10 minutes of streaming broadcasts for 5 to 6 hours. As the popularity of single-person broadcast media services increases, the demand for editing programs is also increasing. However, on the contrary, finding that the editor's work environment has not improved, we felt that it was worth studying the highlight editing prediction model. For the data preprocessing process for machine learning, our team extracts chat traffic and voice energy analysis-based highlight sections, analyzes highlight section chat characteristics, and proposes a final highlight prediction model through LSTM based on highlight section chat characteristics.

**Keywords:** Shipping Management System, Delivery distribution, Destination allocation algorithm, Destination optimal route algorithm, genetic algorithm

**목 차**

**1. 서론 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5**

**2. 데이터 분석 및 검증 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6**

2.1 시스템 프로세스 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6

2.2 데이터 수집 및 전처리 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

2.3 알고리즘 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

**3. 특성 분석 및 머신 러닝 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11**

3.1 하이라이트 채팅 특성 분석 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11

3.2 머신 러닝 및 검증 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17

**4. 시스템 구현 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20**

4.1 시스템 구조 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20

4.2 웹사이트 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_21

**5. 결론 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23**

5.1 기대효과 및 결론 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_23

**6. 참고자료 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_24**

1. **서론**

본 프로젝트의 주제는 채팅 특성 머신러닝을 이용하여 개인 방송 하이라이트 자동 추출 시스템으로, 비인기부터 인기 스트리머의 긴 방송 시간의 하이라이트 구간을 채팅 특성 분석을 통해 추출하여 시청자들에게 편집 영상이 유튜브에 업로드 되기 전에 미리 영상의 하이라이트를 볼 수 있는 방식이다. 개인 게임 방송이 유명한 트위치(Twitch) 안에서 유튜브(YouTube) 편집자와 게임 방송 시청자를 대상으로 스트리밍 서비스가 이루어지고 있는 여러 영상과 해당 영상의 하이라이트 편집본인 유튜브 영상을 대상으로 연구하고 있다.

본 프로젝트는 비인기 ~ 인기 스트리머의 개인 게임 및 일반 스트리밍 방송을 채팅 특성 머신러닝을 이용하여 하이라이트 자동 추출하는 것을 목적으로 한다.

기존 하이라이트 추출 관련 연구의 경우 주로 승패가 발생하는 sports 또는 e-sports 경기를 대상으로 연구하였다. 이는 승률, 승점 등의 객관적인 지표가 하이라이트에 큰 영향을 끼치고, 관중의 소리에 따른 하이라이트 판별도 가능하다. 하지만 본 프로젝트의 대상인 다양한 개인 방송은 승패 결과가 존재하는 게임 방송의 하이라이트 일지라도 승패 결과에 거의 영향을 받지 않는다. 게임 결과와 상관없이 흥미를 유발하는 부분이 하이라이트로 선정되었다. 이때 유일하게 비교할 수 있는 기존 유튜브 하이라이트 편집본과 그에 해당하는 스트리밍 영상 비교를 기반으로 데이터를 검증하였다.

채팅 데이터를 이용한 기존의 연구들은 대개 채팅의 수를 나타낸 트래픽 위주로 머신러닝 기법을 적용하였다. 하지만 본 연구가 대상으로 하는 개인 방송의 개인 방송의 경우에는 시청자의 흥미가 하이라이트 구간 결정에 중요한 지표가 되기 때문에 채팅의 수뿐만 아니라 내용도 중요한 영향을 미친다. 그렇기 때문에 채팅의 트래픽은 하이라이트 구간을 선정하는 데에 이용하였고, 채팅의 내용을 분석하여 얻어낸 채팅 데이터의 특성이 머신러닝의 기반이 된다.

음성 데이터를 활용하여 하이라이트 구간을 탐색한 기존 연구의 경우, 축구나 농구 같은 스포츠 경기에서의 휘슬 소리나 관중들의 함성소리, 해설자의 목소리 등 객관적으로 하이라이트에 영향을 미치고 머신러닝에 이용할 수 있는 요소들이 있다. 그러나 개인방송의 경우, 목소리의 차이, 음악소리 등으로 방송과 방송인마다 음성데이터가 큰 차이를 보였고, 모든 영상에서 공통적으로 발생하는 특징이 없다. 그러므로 머신러닝을 통해 음성을 분석하지 않고, 음성의 구간별 에너지를 비교하여 하이라이트 구간 선정에 이용하였으며, 이를 채팅 트래픽을 기반으로 선정한 하이라이트 구간과 병합하여 사용하였다.

기존 유튜브 하이라이트 편집본의 경우, 전체 스트리밍 영상을 편집자가 직접 시청하며 주관적인 기준으로 하이라이트를 선정하여 편집했기 때문에 다른 하이라이트 포인트를 놓칠 수도 있다. 데이터에 근거하는 하이라이트 자동 탐색 시스템을 이용한다면 조금 더 합리적인 하이라이트 구간을 제공하여 기존 유튜브 편집본의 단점을 개선할 수 있다. 뿐더러, 편집자 또는 새로 유입된 스트리밍 시청자들에게 시간 및 비용적인 측면에서 감소 효과를 얻을 수 있다.

프로젝트의 머신러닝 측면에서는 하이라이트 구간 채팅 특성 분석을 머신러닝을 위한 데이터 전처리 과정으로 보고, 하이라이트 구간 채팅 특성으로 기반으로 LSTM 모델을 연구하고 있다. 시스템 개발 측면에서는 Web과 연동하여URL을 입력하면 해당 영상과 그 영상에 따른 하이라이트 구간이 추출된다.

1. **데이터 분석 및 검증**

2.1 시스템 프로세스

본 프로젝트의 프로세스는 머신러닝을 위한 데이터 전처리 과정과 최종 하이라이트 예측 모델 구축으로 나눌 수 있다. 각각의 프로세스는 다음과 같다.

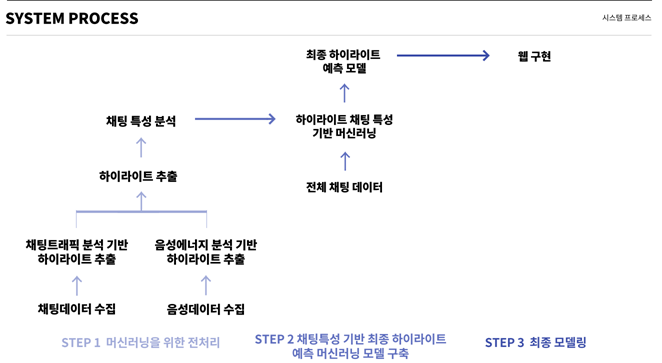


Figure 1. 시스템 프로세스

먼저 개인방송의 채팅 데이터 수집 및 음성 데이터를 수집한다. 그 후, 각 채팅과 음성에서 이동평균법을 이용하여 하이라이트 자동추출 알고리즘을 통해 각각의 하이라이트 포인트들을 선정한다. 채팅 트래픽 분석 기반으로 선정된 하이라이트 포인트들과 음성 에너지 분석 기반으로 선정된 하이라이트 포인트들을 합친 후 그 하이라이트 포인트 구간들에 해당하는 채팅들을 추출한다. 그 후, 하이라이트 구간 채팅들의 특성을 분석하여 머신러닝을 위한 준비 단계를 마친다.

그 이후, 분석된 결과를 토대로 머신러닝 과정을 진행한다. 전체 채팅들을 각각의 채팅마다 하이라이트 채팅 / 비하이라이트 채팅이라 구분한 후 머신러닝 Tensorflow 환경에서 LSTM을 통해 학습시킨다. 이후 새로운 채팅이 들어왔을 때 그것을 보고 그 채팅이 나오는 부분이 하이라이트인지 비하이라이트인지 구분해주는 최종 하이라이트 예측 모델을 만든다. 그 모델을 웹이랑 연동하여 마지막 단계에서는 어떠한 영상을 입력받으면 해당 영상의 채팅을 보고 하이라이트 포인트들을 추출해준다.

2.2 데이터 수집 및 전처리

테이블이(가) 표시된 사진

자동 생성된 설명

Figure 2. 데이터 수집

트위치 개발자 API를 이용하여 원하는 비디오의 채팅 데이터를 다운로드하여 사용하였다. 트위치는 발전을 위해 개발자 API를 이용하여 필요한 데이터를 다운받을 수 있다. Python으로 코드를 작성하여 해당 비디오의 채팅을 다운받았다.

영상 데이터 다운로드 후, 음성 데이터를 분리하여 wav, Flac으로 변환하여 Python를 이용하였다. 그 다음, 음성 raw 데이터를 liborsa.onset\_onset\_strength 모듈을 이용하여 데이터 전처리 과정을 마무리하였다. Onset\_strength은 one-dimensional array로 각 프레임에서 증가하는 에너지 양을 시각화로 나타냈다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Figure 3. 음성 데이터

음성 데이터에서 하이라이트 포인트를 추출하기 위해 초당 증가하는 스펙트럼 에너지를 1차원 array로 구성하여 전체 상위 1.5%의 음성 에너지를 가지는 하이라이트 포인트를 추출하였다. 음성의 경우 상위 1.5%까지의 음성에너지 데이터 양이 전체 영상 길이의 약 3%에 해당하는데 이는, 채팅 하이라이트 수와 비슷한 하이라이트 수로 뽑아 채팅과 음성 하이라이트의 수가 대등하도록 만들었다.

2.3 알고리즘

본 장에서의 알고리즘이란 하이라이트를 선정하기 위한 알고리즘으로 여러가지 값들을 다양하게 이용하여 영상에서의 하이라이트를 구분해주는 알고리즘을 말한다.

먼저, 기존 연구에서 Local Mac peak Detection Algorithm을 이용하였다. 이 알고리즘은 초당 변화하는 채팅 특성과 수를 고려하지 않았기 때문에 하이라이트 선정 시, 문제 발생 가능성이 있다고 판단하였다.

2.3.1 **Local Max Peak Detection Algorithm**

처음에는 시간 t 부분에서 채팅 트래픽이 가장 많은 부분을 찾아주는 알고리즘을 사용하려 하였다. 시간 t 부분에서 가장 큰 값을 갖는 후보군 T를 찾아 형성한다.

T= {t: C(t) is a local max}

지역 최대점은 이전, 이후에 비해 트래픽이 큰 지점을 의미한다. 즉, C(t) > C(t-1), C(t) > C(t+1)이면 t는 지역 최대점이다. 후보군 T가 하이라이트라고 선별되면 채팅 딜레이 등을 고려하여 앞뒤 구간을 하이라이트 구간이라 채택한다.

2.3.2 **Peak Detection Algorithm**

Peak Detection Algorithm은 이동평균법을 이용하여 편차가 일계 값보다 크면 detect하여 채팅이 많이 발생하는 지점을 Peak로 선정하였다. 주요 기능은 lag(구간), Threshold(임계값), Influence(영향)이 있다. 이를 활용하여 이동평균법을 쓰는 이유는 생방송 시청자 수가 시간에 달라지기 때문이다.

Local Max만을 이용하여 하이라이트를 추출해주는 알고리즘은 단순히 앞뒤 구간만을 고려하여 채팅이 가장 많은 하이라이트 구간만을 추출해주기 때문에 단순하다고 판단이 되어 다른 알고리즘까지 찾아보았다. 또한, 시청자수가 많은 영상들은 채팅의 수가 다양하게 나타나서 판별하기 쉬웠지만, 채팅자수가 적거나 영상의 길이가 긴 영상들은 Local Max를 구하기가 힘들어 참여자수 대비 상위 트래픽을 가진 채팅을 나타나지 못한다는 문제점이 발생하였다. 그래서 새롭게 고려한 알고리즘은 이동평균법을 이용한 Peak Detection Algorithm이었다. 이동평균법을 이용하여 편차가 임계 값보다 크면 detect하여 그 지점을 Peak로 선정하였다. 이동평균법을 쓰는 이유는 각 방송바다 참여하는 생방송 시청자 수가 시간에 달라지기 때문이다. 단순히 t지점에서의 앞 뒤 부분만을 고려한 것이 아니라 영상마다 적절한 값의 lag를 설정을 하여 그 구간 내에서의 이동평균, 표준편차를 활용하여 하이라이트를 판별해준다. 이동평균법을 이용한 알고리즘을 개발하여 적용하여 이러한 특징을 반영하고 더 좋은 결과를 뽑아낼 수 있었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**Figure 4. Highlight Detection 알고리즘**

2.3.3 **검증**

제안한 하이라이트 추출 알고리즘의 성능을 평가하기 위해 대조군과 비교하였다. 하이라이트 편집의 대조군은 기존 유튜브 하이라이트 편집본이다. 기존 유튜브 하이라이트 편집본을 정답 하이라이트라고 판단하였다. 한 유튜브 하이라이트 편집본은 여러 개의 하이라이트 구간을 가진다. 또한, 한 하이라이트 구간은 여러 개의 하이라이트 포인트들로 구성되어 있다. 본 프로젝트에서 제안하는 알고리즘에 대한 성능을 평가할 때, 하나의 하이라이트 구간에서 하나 이상의 하이라이트 포인트가 판별되면 하이라이트 구간이 탐지하였다고 보고 검증을 진행하였다. 채팅 트래픽 기반 하이라이트 추출 알고리즘으로 뽑은 하이라이트 포인트들은 기존 유튜브 하이라이트 편집본 구간의 하이라이트 포인트들을 평균 53% 정도 포함하였다. 음성 에너지 기반 하이라이트 추출 알고리즘으로 뽑은 하이라이트 포인트들은 기존 유튜브 하이라이트 편집본 구간의 하이라이트 포인트들을 평균 39% 포함하였다. 각각의 알고리즘의 정확도는 다소 낮았지만 각각의 알고리즘에서 뽑은 하이라이트 포인트들을 합쳐 계산을 하면 정확도가 약 70% 정도가 나왔다. 여기서 단순히 실험군의 수가 증가하여 정확도가 올라간 것은 아닌지 확인하기 위해 각 포인트가 대조군과 일치하는 포인트를 확인해보았고 그 결과 실험군1인 채팅 포인트와 대조군이 일치하는 포인트와 실험군2인 음성 포인트와 대조군이 일치하는 포인트가 대조군의 서로 다른 포인트와 일치하는 경우가 서로 같은 포인트와 일치하는 경우보다 더 많이 나타났다. 이는 채팅 포인트와 음성 포인트가 서로 다른 하이라이트 특징을 반영한다고 볼 수 있으며 서로 상호보완 되기 때문에 각각 따로 추출한 하이라이트 포인트 보다 좋은 하이라이트 포인트를 추출한다고 볼 수 있다.

1. **특성 분석 및 머신러닝**

3.1 하이라이트 채팅 특성 분석

3.1.1 개인 방송 실시간 채팅 통계

개인 방송 시청자의 실시간 채팅의 언어적 특징을 파악하기 위해 한 채팅의 평균 음절과 어절, 한글 낱글자의 채팅 내 평균 개수를 분석했다. 어절은 문장 성분의 최소 단위로서 띄어쓰기의 단위이며, 음절은 음의 한 마디로서, 단어 또는 단어의 일부를 이루는 하나의 단위로, 한 글자이다. 낱글자란 하나하나의 글자를 글자의 단위로 이르는 말로, ‘ㅋ’, ’ㅠ’와 같이 다른 자소와 조합되지 않고 발생한 글자이다.

수집한 전체 방송의 채팅 통계는 다음과 같다.



**Figure 5. 전체 방송의 채팅 통계**

온라인상에서 발생한 채팅이 띄어쓰기가 잘 이루어지지 않는 등의 이유로 분석이 용이하지 않을 수 있기 때문에 세종말뭉치 통계와 비교하여 이를 확인해 보았다. 그 결과 표 1에서 음절 수 대비 평균 어절 수가 3.8으로 말뭉치의 4.35와 큰 차이를 보이지 않았다. 이는 실시간 채팅에서도 띄어쓰기가 습관화되어 잘 나타나는 것을 의미한다. 그렇기 때문에 채팅을 분석하여 유효한 특성들을 추출해 낼 수 있을 것이라 생각했다.

3.1.1 하이라이트 & 비하이라이트 구간 채팅의 특징

하이라이트로 태깅된 채팅과 태깅 되지 않은 비하이라이트 구간의 채팅으로부터 각각의 언어적 요소들을 비교하여 분석해 보았다.



**Figure 6. 하이라이트 구간의 채팅 통계**



**Figure 7. 비하이라이트 구간의 채팅 통계**



**Figure 8. 하이라이트 & 비하이라이트 채팅 통계 비교**

먼저 초당 평균 채팅, 어절, 음절, 낱글자의 개수를 살펴보면, 초당 발생하는 평균 채팅의 수는 하이라이트 구간에서 3.595로 비하이라이트 구간의 1.498보다 평균 2.5배가량 많이 발생한다. 영상에 따라서는 최소 1.7배부터 최대 8.4배까지 하이라이트 구간의 초당 채팅 수가 높게 나타났다. 스트리머의 평균 시청자 수가 많을수록 하이라이트 구간과 비하이라이트 구간의 초당 평균 채팅 수의 차이가 적게 나타나는 경향이 있고, 반대로 평균 시청자 수가 적은 스트리머일수록 하이라이트 구간과 비하이라이트 구간의 차이가 극명한 경향이 있다.

하이라이트 구간과 비하이라이트 구간의 한 채팅당 어절, 음절, 낱글자 수를 비교한 결과, 하이라이트 구간에서는 한 채팅당 어절이 1.949개로 비하이라이트 구간의 2.271개 보다 작게 나타났으며, 음절 또한 하이라이트 구간에서 7.471개로 비하이라이트 구간의 7.904개 보다 작게 나타났다. 이는 비하이라이트 구간에서의 채팅이 하이라이트 구간의 채팅이 길게 나타남을 의미한다. 채팅당 낱글자 수의 경우 하이라이트 구간의 채팅에서는 3.162개, 비하이라이트 구간에서는 2.144개로 하이라이트 구간에서 훨씬 크게 나타났다. 이는 하이라이트 구간에서는 낱글자를 포함한 짧은 채팅이 많이 발생하는 특징이 있음을 알 수 있다.

3.1.3 초를 윈도우로 나타내어 분석

하이라이트 구간 채팅의 특성을 파악하기 앞서, 분석 단위가 한 채팅이라면, 한 채팅마다 특징을 추출하기에 채팅 안에 들어있는 특징이 부족하기 때문에 정확한 분석이 어렵다. 그래서 분석 단위를 초단위로 하려 했으나, 데이터의 부족으로 인해 분석에 어려움이 있었다. 그래서 3초 단위의 윈도우를 생성하였다. 예를 들어, 0초에서 2초까지 발생한 채팅을 윈도우0의 정보로 저장하고, 1초에서 3초까지 발생한 채팅을 윈도우 1의 정보로 저장하는 방식으로 겹쳐서 이동하며 채팅의 특징을 얻는다. 이 때, 시작 시간을 기준으로 설정한 이유는 채팅은 영상을 시청 후에 나타나는 반응이기 때문이다. 윈도우 내의 채팅 특성을 수치로 변환하여 머신러닝에 사용한다.

3.1.4 머신러닝 특성 구성

앞에서 분석한 결과를 바탕으로 아래 목록과 같은 머신러닝 특성을 구성하였다.

1. 채팅 빈도
2. 윈도우 내 채팅 수
3. 윈도우 내 채팅 참여자 수 비율
4. 윈도우 시간 정규화
5. 언어적 빈도
   1. 윈도우 내 낱글자 수
   2. 윈도우 내 총 음절 수
   3. 윈도우 내 총 어절 수
   4. 윈도우 내 한 채팅당 평균 낱글자 수
   5. 윈도우 내 한 채팅당 평균 음절 수
   6. 윈도우 내 한 채팅당 평균 음절 수
6. 내용 정보
7. 윈도우 내 형태소의 하이라이트 점수 합
8. 윈도우 내 형태소의 비하이라이트 점수 합

① - (ㄱ)은 윈도우에 발생한 채팅의 단순 발생 수로, 하이라이트 구간에서 비하이라이트 구간보다 채팅이 많이 발생하는 특성을 반영한다 ① - (ㄴ)은 전체 채팅 참여자 수를 현재 채팅 참여자 수로 나눈 값으로, 하이라이트 구간에 다양한 채팅 참여자가 나타나는 특성을 반영하였다. ① - (ㄷ)은 현재 윈도우의 시간적 위치 비율을 나타내는 값으로, 로 계산하여 나타냈다.

② - (ㄱ)은 윈도우 내에 발생하는 낱글자의 수이다. 낱글자 수의 경우 하이라이트 구간에서 비하이라이트 구간보다 평균 3.5배가량 많게 나타나는 것을 볼 수 있다. 그렇기 때문에 하이라이트 구간의 특성을 잘 반영한다. ② - (ㄴ)와 ② - (ㄷ)은 윈도우 내에 발생하는 음절, 어절의 개수로, 하이라이트 구간에서 더 높게 나타난다. ② - (ㄹ)은 윈도우 내에서 한 채팅에 발생하는 낱글자 수이다. 이는 하이라이트 구간에서 더 많이 나타난다. ② - (ㅁ)과 ② - (ㅂ)은 오히려 비하이라이트 구간에서 더 많이 나타나는데, 이는 비하이라이트구간의 채팅이 하이라이트구간의 채팅보다 더 긴 특성을 반영한다.

③은 한 형태소가 하이라이트 구간에 많이 나타나는지 비하이라이트 구간에서 많이 나타나는지를 반영하기 위한 정보로, 하이라이트 점수는 윈도우 내 채팅들이 하이라이트 구간에 많이 나오는 형태소를 얼마나 포함하는지를 반영하고, 비하이라이트 점수는 이와 반대로 윈도우 내 채팅들이 비하이라이트 구간에 많이 나오는 형태소를 얼마나 포함하는지를 반영한다. 이때Konlpy의 Okt형태소 분석기를 사용하여 윈도우 내 채팅을 형태소 단위로 분리하였고, 영상에서 발생한 모든 형태소 별로 하이라이트 구간과 비하이라트 구간에서 나타나는 수를 세어 수식에 적용하였다.

이때 낱글자 형태소의 경우, ‘ㅋㅋㅋㅋㅋㅋㅋㅋ’ 나 ‘ㅂㄷㅂㄷㅂㄷ’과 같은 자소열의 다양한 반복 패턴을 동일 두 낱글자의 4번 이상의 반복은4개로 줄여(ex. ‘ㅋㅋㅋㅋ’, ’ㅂㄷㅂㄷ’) 한글 채팅의 특성을 반영하고 자료 부족 문제를 감소시켰다.

임의의 한 형태소 w가 가지는 하이라이트 점수 Hscore(w), 비하이라이트 점수Nscore(w)는 다음과 같다.

이때,

는 형태소 w가 하이라이트에 존재하는 비율이다. 는 형태소 w가 비하이라이트에 존재하는 비율이다. 보다 가 작을 경우, 형태소 w는 비하이라이트 구간에 더 많이 발생한다는 의미이다. 따라서 는 0이 된다. 보다 가 클 경우에는 반대로 형태소 w가 하이라이트구간에서 더 많이 발생한다는 의미이다. 이때 는 위 식에서와 같은 방식으로 0~1사이의 점수로 구해진다. 는 의 반대로, 비하이라이트 구간에 더 많이 나타날 경우에 0~1 사이의 점수로 구해진다. 이러한 점수를 머신러닝 특성 중 하나로 반영한다. 아래 예시와 같이 영상에서 발생한 모든 형태소에 대해 Hscore, Nscore가 구해지고 윈도우 내 채팅에서 발생한 각 형태소의 Hscore, Nscore 합을 머신러닝 특성으로 가진다.

테이블이(가) 표시된 사진

자동 생성된 설명

**Figure 9. 수집한 영상의 채팅 데이터에서 발생한 형태소와 각 형태소의 점수 계산 예시**

수집한 채팅 데이터의 특성에 기반하여 머신러닝 모델을 구현할 때, 전처리 과정에서 하이라이트 여부가 라벨링된 하이라이트 여부를 이용해 수집한 채팅 데이터에서 발생한 형태소의 Hscore, Nscore를 계산할 수 있었다. 그러나 머신러닝 모델에 적용될 임의의 새로운 채팅 데이터의 경우, 머신러닝의 결과로 하이라이트가 라벨링 된다. 즉 머신러닝 모델 분석에 적용되기 이전에는 하이라이트 여부가 존재하지 않는다. 따라서 윈도우 내 채팅의 형태소의 Hscore, Nscore합의 계산이 불가능 해지는데 이를 해결하기 위해 수집한 모든 영상의 채팅 데이터에서 발생한 모든 형태소의 Hscore, Nscore를 가지는 형태소Scoreboard를 생성하였다 형태소Scoreboard에 있는 중복을 제외한 형태소의 수는 137,129개로 489,814개의 채팅데이터에서 발생한 모든 형태소의 집합이다. 이를 이용해 머신러닝 모델에 적용해 하이라이트를 예측할 채팅 데이터에서 발생한 형태소의 Hscore, Nscore를 계산한다. 이때 Scoreboard에 없는 형태소가 채팅에서 발생했을 경우, 일반적인 형태소라고 보기 어렵다고 판단해 그 형태소의 Hscore, Nscore를 모두 0으로 하여 윈도우 내 채팅의 형태소의 Hscore, Nscore합에 미치는 영향이 없도록 하였다.

3.2 머신러닝

3.2.1 모델 학습 및 비교

데이터에서 추출한 채팅 특성들을 위와 같이 수치화하여 파이썬의 tensorflow keras 환경에서 여러 머신러닝 모델을 학습시킨 후 비교하였다. 학습시킨 모델들은 Simple RNN, LSTM, Bi-LSTM, GRU, KNN, SVM 이며, 약 50만개의 채팅의 특성을 학습시킨 후 정확도를 비교하였다.

먼저 Simple – RNN의 경우 짧은 순차열의 학습을 진행하며 단기적인 기억능력이 좋은 모델로, 학습 결과 0.8790의 정확도를 얻었다. KNN (K-Nearest Neighbor Algorithm) 알고리즘의 경우, 분류에 적합한 모델로서 데이터가 클수록 좋은 성능을 내며, 노이즈에 큰 영향을 받지 않는 모델이고, 학습 결과 0.8511의 정확도를 얻었다. SVM (support vector machine)은 고차원 데이터를 학습시킬 때에 효율적이고, 과적합이 적으며 분류/예측에 모두 이용 가능한 모델로, 0.7862로 다소 낮은 정확도를 보였다. LSTM 모델은 장단기 기억 보존에 유용하며, 시계열 데이터를 다루는데에 효과적인 모델로 0.9389의 가장 높은 정확도를 보였다. Bi-LSTM 모델은 양방향 LSTM(Long Short Term Memory) 모델로, 기존의 LSTM 계층에 역방향으로 처리하는 LSTM계층을 추가한 모델로, 0.9225의 정확도를 얻었고, GRU(Gated Recurrent Unit) 모델은 LSTM의 구조를 간단하게 개선한 모델로, 빠른 계산 속도를 가졌다. GRU 모델은 0.9278의 정확도를 보였다. 6개 모델의 정확도를 나타낸 표는 다음과 같다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 모델 | Simple-RNN | LSTM | Bi-LSTM | GRU | KNN | SVM |
| 정확도 | 0.8790 | 0.9389 | 0.9225 | 0.9278 | 0.8511 | 0.7862 |

**Figure 10. 머신러닝 정확도 비교**

위의 모델들 중 LSTM, Bi-LSTM, GRU 세 모델은 0.92~0.93으로 비슷한 정확도를 보였기에, 세 모델 중 최종 사용할 모델을 선정하기 위하여 각각의 모델에 특성은 분석하였으나 학습 시에는 사용되지 않은 영상들의 채팅 특성을 적용하여 하이라이트 포인트를 추출해 비교하였다. 그 결과 GRU 모델이 기존 라벨링된 하이라이트 포인트 수와 가장 비슷한 수의 포인트를 추출하였으나 실제 하이라이트 포인트와 일치하는 포인트의 비율은 다소 낮았으며, 아래와 같이 LSTM 모델의 정확도가 71~85%로 세 모델 중 가장 높은 정확도를 보였다. 또한, LSTM 모델은 시계열 데이터인 채팅 데이터를 다루는 데에도 유용할 것이라 판단하였고, 기존 관련 연구 에서도 LSTM모델을 사용한 사례가 많았으므로 최종 하이라이트 예측에 LSTM 모델을 사용하였다.

3.2.2 하이라이트 포인트 구간화

LSTM 모델을 통해 하이라이트 포인트를 추출하면 1초 단위마다 하이라이트 여부가 0,1로 라벨링 되기 때문에, 매끄러운 하이라이트 구간 재생이 어렵다. 그러므로 하이라이트 여부가 1로 라벨링된 시간의 +-5초, 총 10초를 하이라이트 구간으로 나타냈다. 이 때, 아래와 같이 5초 이하의 간격으로 하이라이트 여부가 1로 라벨링 되어 구간이 겹칠 경우, 해당 구간을 이어서 나타냈다. 테이블이(가) 표시된 사진

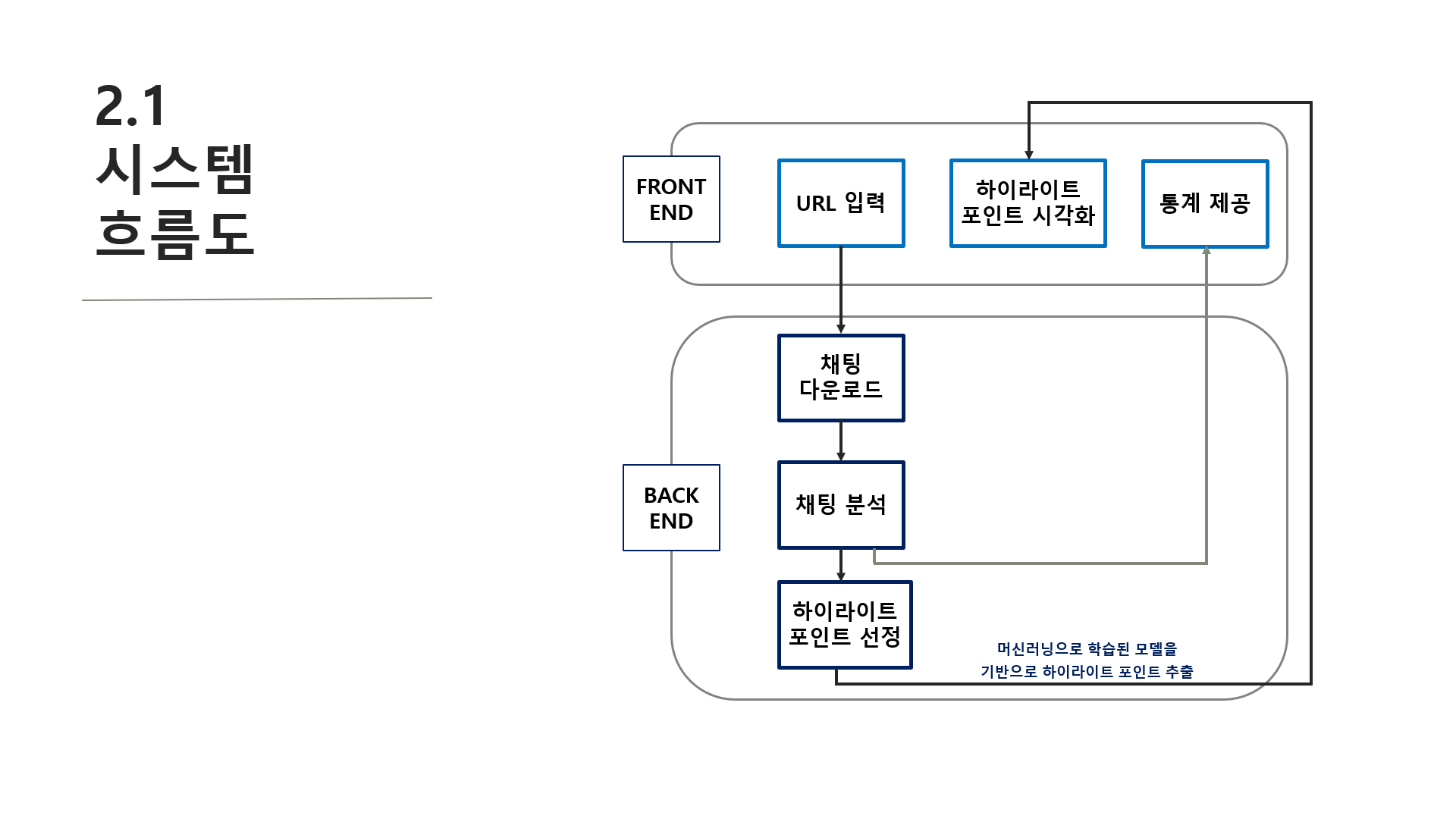
자동 생성된 설명

**Figure 11. 추출된 하이라이트 구간**

1. **시스템 구현**

4.1 시스템 구조

전체 시스템 구조는 아래와 같다.



**Figure 12. 전체 시스템 구조**

검색하는 칸에URL을 입력하면 해당 영상의 채팅을 다운로드 받는다. 채팅이 다운되면 채팅을 만들었던 머신러닝 기반 하이라이트 자동추출 모델을 통해 하이라이트 포인트가 선정된다. 원하는 하이라이트 포인트를 클릭하면 그 구간으로 이동한다. 또한, 해당 영상들의 각종 통계들도 표시되어 하이라이트 포인트와 같이 웹사이트에 나타난다.

Data 형식 및 전달 방식 변경은 다음과 같다.

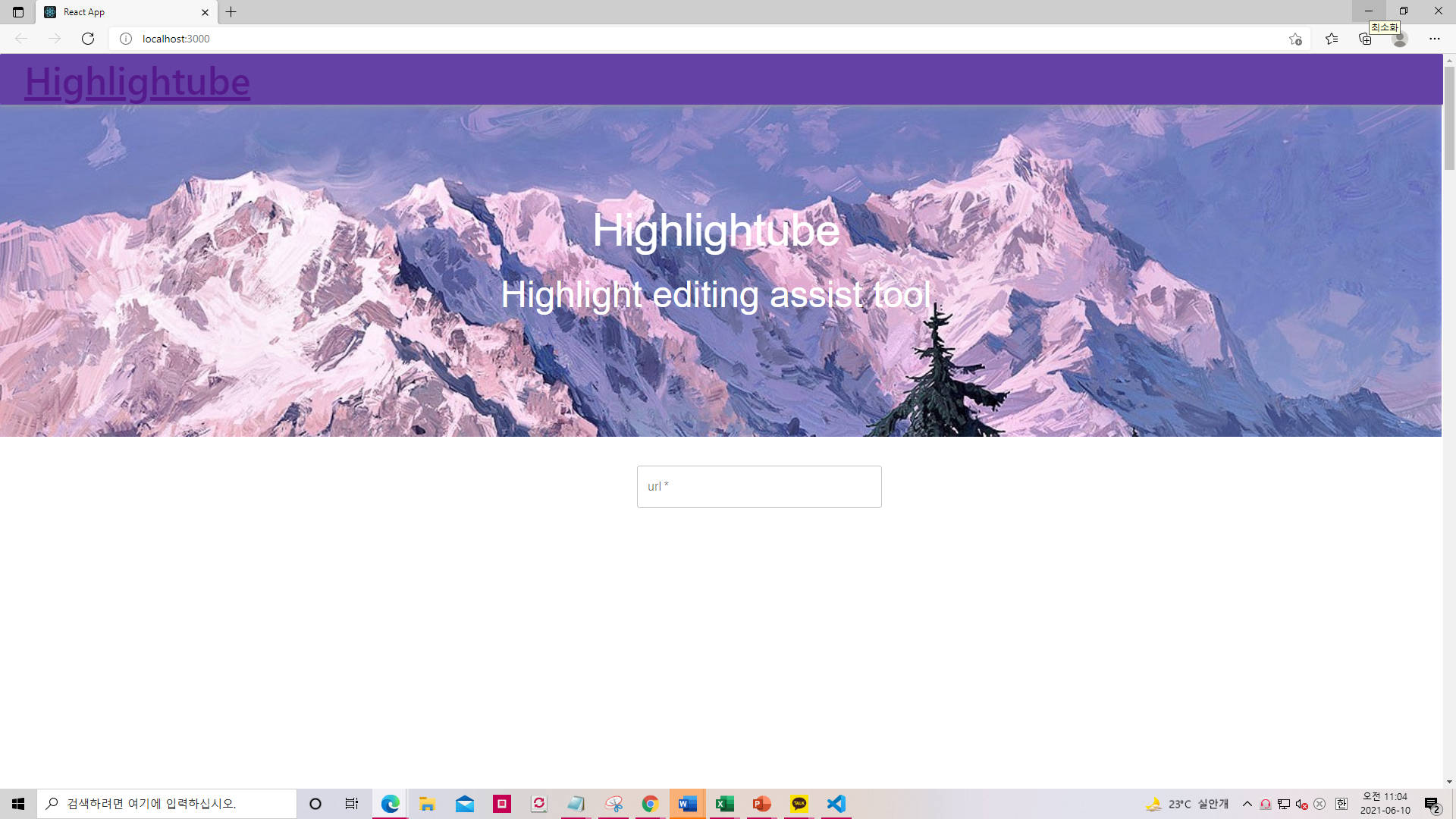
텍스트이(가) 표시된 사진

자동 생성된 설명

자바스크립트 라이브러리의 하나로서 사용자 인터페이스를 만드는데 이용하는 React를 활용했다. React는 프레임워크와 라이브러리가 많아 유연성이 높다. 때문에 Python과 json 언어를 연결할 때 child process tool을 이용하여 데이터 저장 및 이동이 쉽게 가능했다.

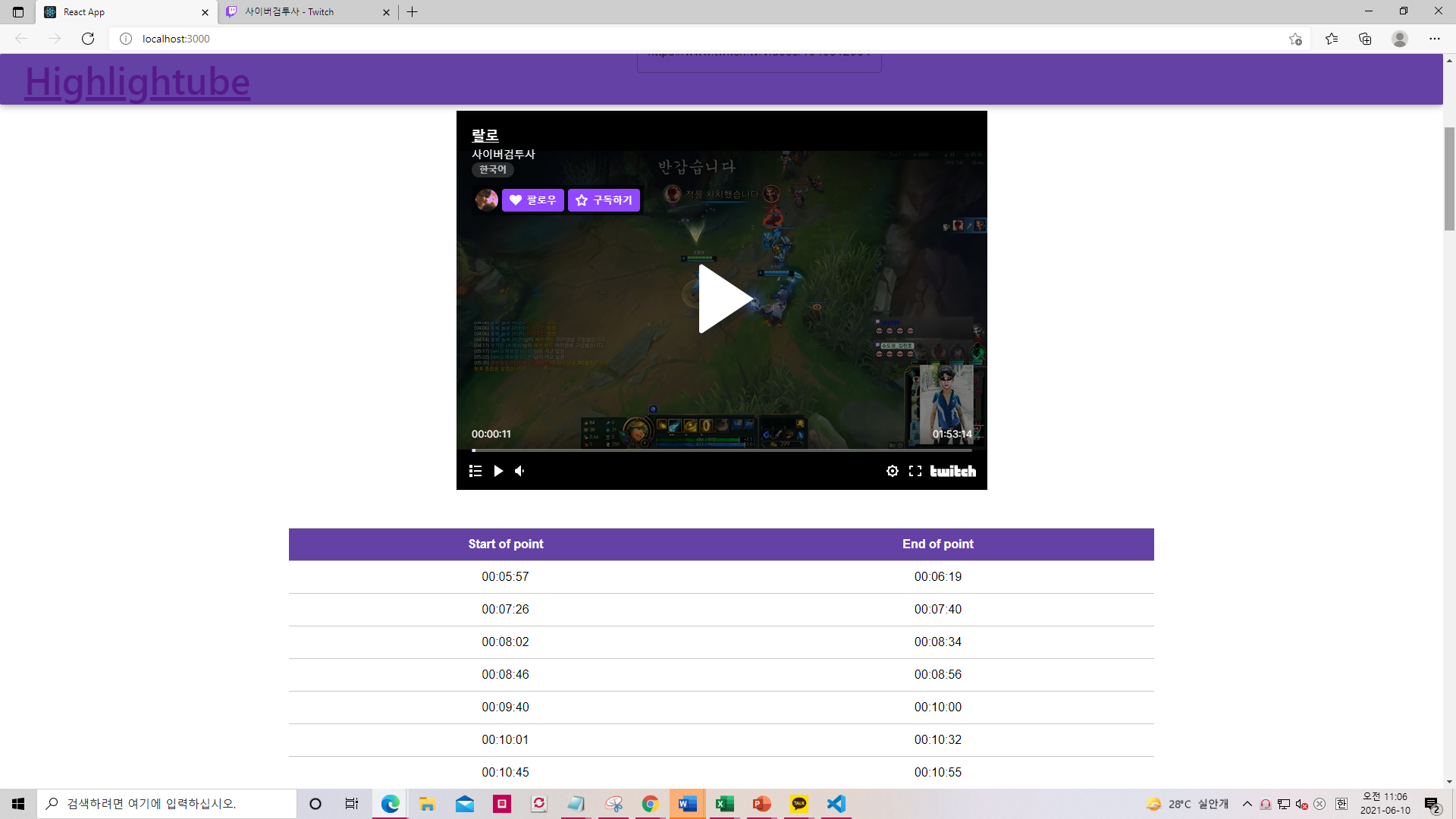
4.2 웹사이트

웹 사이트의 디자인은 아래와 같다.



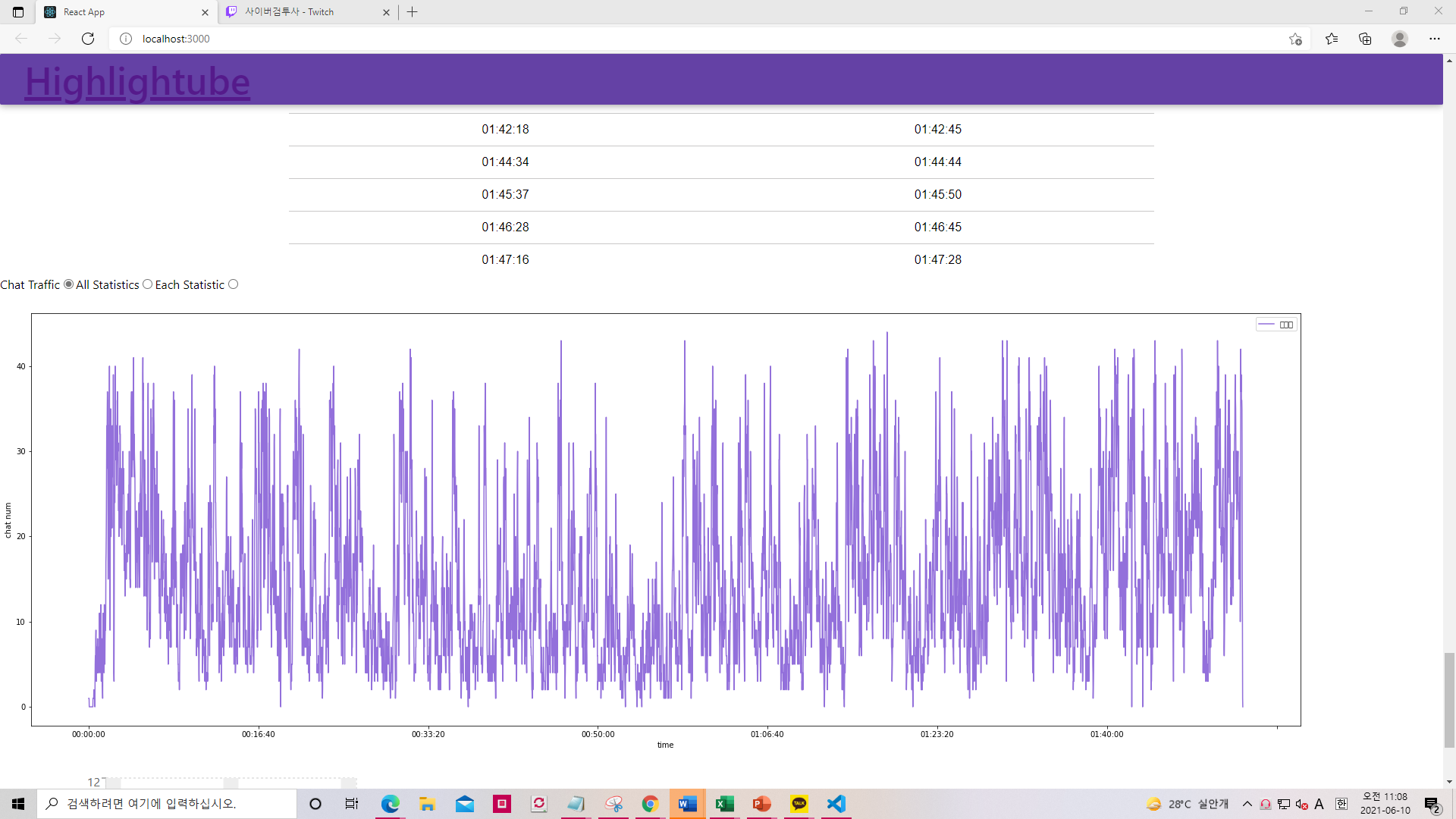
**Figure 13. 웹 사이트의 첫 화면**

URL 검색 창에 트위치 URL을 복사하여 입력하면,



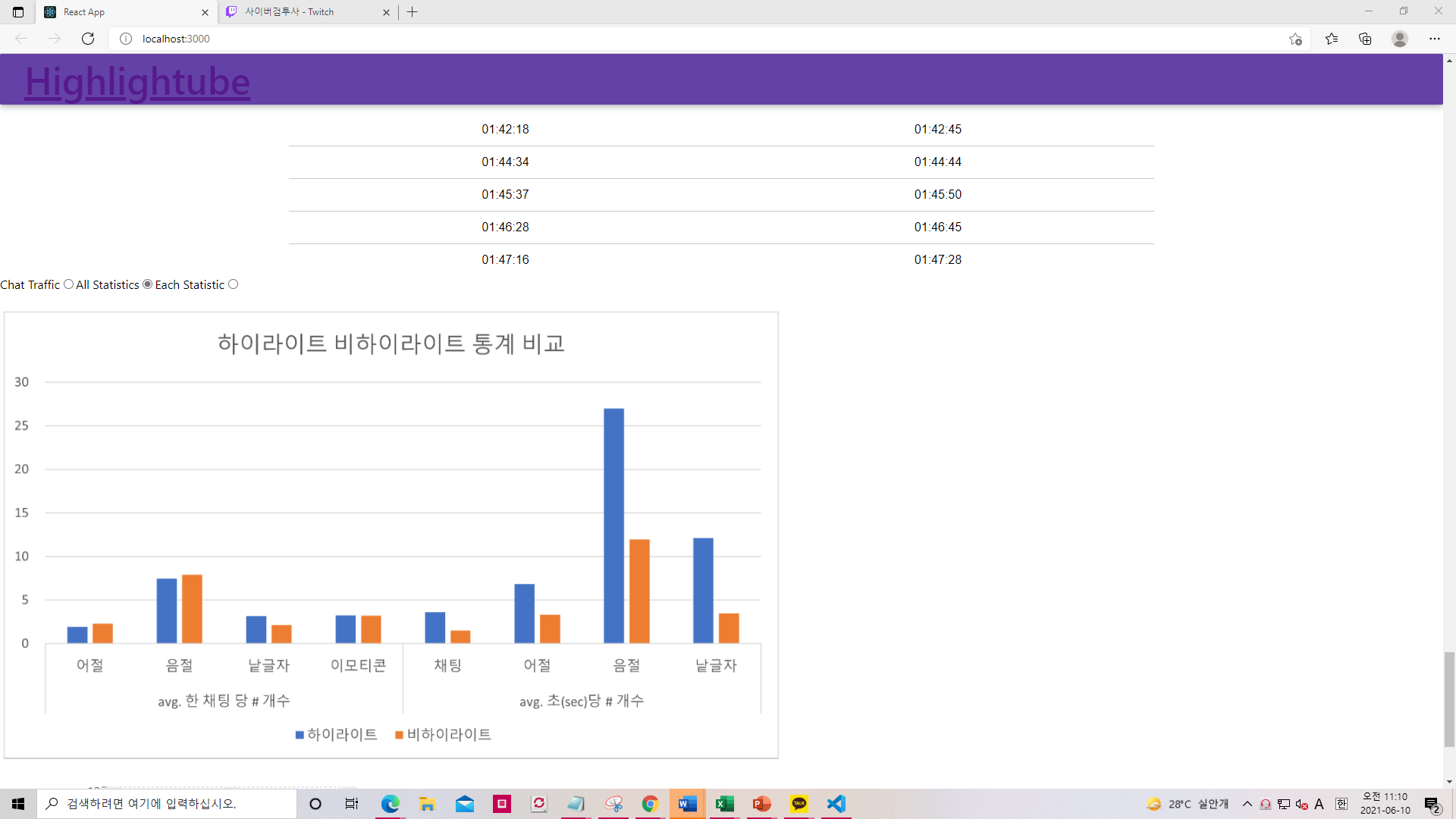
**Figure 14. 하이라이트 포인트 추출**

트위치 영상과 하이라이트의 시작점과 끝점의 구간들이 추출된다. 원하는 구간들을 클릭하여 영상이 그 구간으로 넘어갈 수 있도록 설정하였다.



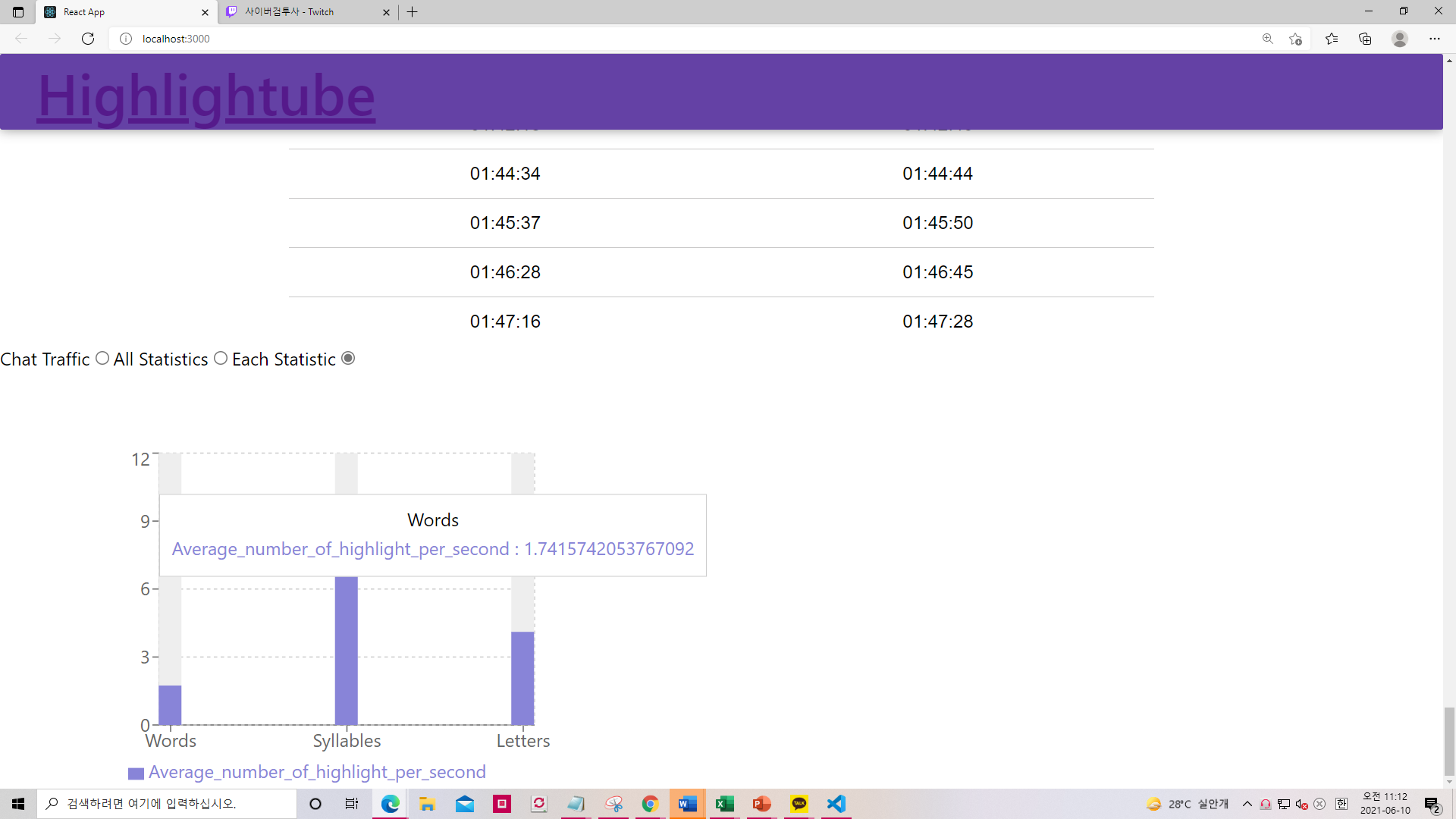
**Figure 15. 통계 출력**

그 아래로, 총 세 가지의 통계를 보여준다. 첫 번째는 해당 영상의 Chat Traffic으로 총 시간에 따른 채팅 수의 변화를 한 눈에 볼 수 있도록 설정하였다.



**Figure 16. 하이라이트와 비하이라이트 통계 비교**

두 번째 그래프는 우리가 이용했던 전체 데이터의 하이라이트와 비하이라이트 통계 비교로,



**Figure 17. 어절 수, 음절 수, 낱글자 수의 평균 통계**

해당 영상의 하이라이트 초 당 평균 수와 비교할 수 있다. 이 그래프는 어절 수, 음절 수, 낱글자 수의 평균을 나타냈다.

**5. 결론**

5.1 기대효과 및 결론

프로젝트의 목적은 개인 방송 하이라이트 편집의 요점은 5~6시간 스트리밍 방송을 5~10분으로 요약 및 편집을 하기 위해 소비되는 총 시간 및 비용을 줄이는 것이다. 1인 방송 미디어 서비스의 인기가 증가함에 비례하여 편집 프로그램의 수요도 증가하고 있다. 하지만 이와 반대로 편집자의 업무환경이 개선되지 못함을 발견하면서, 우리는 하이라이트 편집 예측 모델에 연구할 가치가 있다고 느꼈다. 우리의 연구 과정은 머신러닝을 위한 데이터 전처리 과정을 위해 채팅 트래픽 및 음성 에너지 분석 기반 하이라이트 구간을 추출하여 하이라이트 구간 채팅 특성을 분석한 다음, 하이라이트 구간 채팅 특성 기반 LSTM을 통해 최종 하이라이트 예측 모델을 제안한다.

우리 프로젝트의 결과는 정상적인 기준의 개인방송 하이라이트를 채팅데이터에 기반으로 하여 머신 러닝 기법으로 예측하는 객관적인 시스템을 제공한다. 또한, 이 시스템은 노동집약적이고 비효율적인 업무구조를 해결했으며, 고도화된 AI 기술을 더해 편집자의 손을 거치지 않고 바로 활용 가능한 하이라이트 추출 시스템으로 발전이 가능하다. 객관적인 지표들이 존재하고 대상이 한정적인 E-스포츠 대상의 기존 연구와 달리, 더 까다롭고 정성적인 기준의 개인방송 하이라이트를 추출하는 시스템으로서 최근 급부상하는 1인 미디어 컨텐츠 제작자를 대상으로 하기 때문에 시장성, 확장성이 좋고 잠재적 가치가 높은 시스템이 될 수 있다.

**6. 참고자료**

1. Eun-Yul Kim, Gyemin Lee, Highlights Detection based on Chat Traffic in Personal Broadcasting,
2. Seonghun Yoon, SeungJin Lee, Kyung-Joong Kim, League of Legends Game Video Highlight Extraction using Autoencoder”

<http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07613722>

1. Video Highlight Prediction Using Audience Chat Reactions(2017)
2. Z. Xiong, R. Radhakrishnan, A. Divakaran, and TS. Huang, “Highlights extraction from sports video based on an audio-visual marker detection framework”, IEEE International Conference on Multimedia and Expo, Amsterdam, Netherlands, pp. 29-32, 2005, doi:10.1109/ICME.2005.1521352.
3. Hansol Lee and Gyemin Lee, Video Highlight Prediction Using GAN and Multiple Time-Interval Information of Audio and Image
4. Muhammad Rafiqul Islam, Manoranjan Paul, Michael Antolovich, Ashad Kabir, Sports highlights generation using decomposed audio information (2019)
5. 이선희 저, 게임 관련 온라인 개인 방송 시장 동향과 트위치(Twitch) 사업자 전략 (2019)