



boostcamp

Tving Hackathon

주제 : 장면 탐색을 위한 Video to Text / Text to Video 모델

CV-15 (티빙)

팀명: 핑핑이들

팀원: 정현우, 박지완, 최재훈, 임찬혁, 민창기, 이단유

목차

1. Intro
2. 데이터 분석 및 전처리
3. Video to Text
4. Text to Video
5. Evaluation
6. Result
7. 자체 평가 의견
8. 별첨



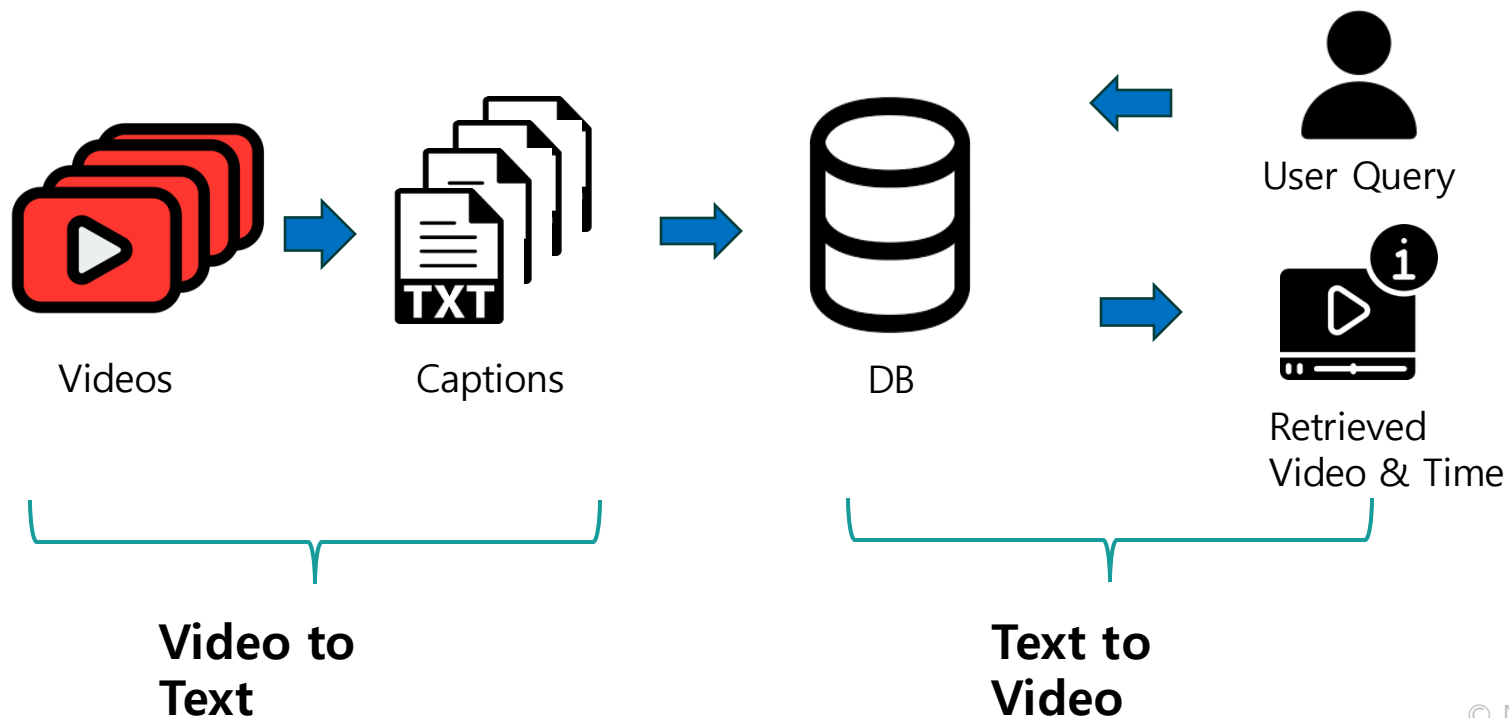
Intro

Intro

해커톤 소개

Video to Text: 비디오를 텍스트로 변환하여 저장하면 메모리를 효율적으로 관리할 수 있으며, 검색이 쉬워지고 비디오의 주요 특징을 한눈에 파악할 수 있다.

Text to Video (Frame): 텍스트를 입력하여 적절한 비디오(Frame)를 검색할 수 있도록 함으로써 사용자 편의성을 높이고 데이터 관리의 유연성을 증대시킬 수 있다.



Intro

팀원 역할

팀원	역할
정현우	전체 파이프라인 실험 및 설계, Video-to-Text 코드 구현, 코드 최적화
박지완	Text-to-Video : embedding, retrieval 코드 구현, web demo 구현
최재훈	모델 탐색 및 전략 수립, 검색 성능 평가 실험, 최종 파이프라인 구축 및 정리
임찬혁	데이터 전처리, captioning 및 query DB 구축, embedding 모델 fine-tuning 실험
민창기	모델 탐색, 모델 훈련 및 평가, 최종 파이프라인 구축, 오디오 캡셔닝 및 자막 모델 실험
이단유	Video-to-Text 캡셔닝 모델 및 프롬프트 실험, 입력 query 생성

Intro

프로젝트 타임라인

내용	1월														2월																				
	1주차					2주차					3주차					4주차					5주차														
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	설연휴	31	1	2	3	4	5	6	7	8	9	10	11	12			
EDA & Preprocessing																																			
Method Search																																			
Modeling & Pipeline																																			
Evaluation																																			
Database Update																																			
Final Pipeline																																			
Prepare Presentation																																			



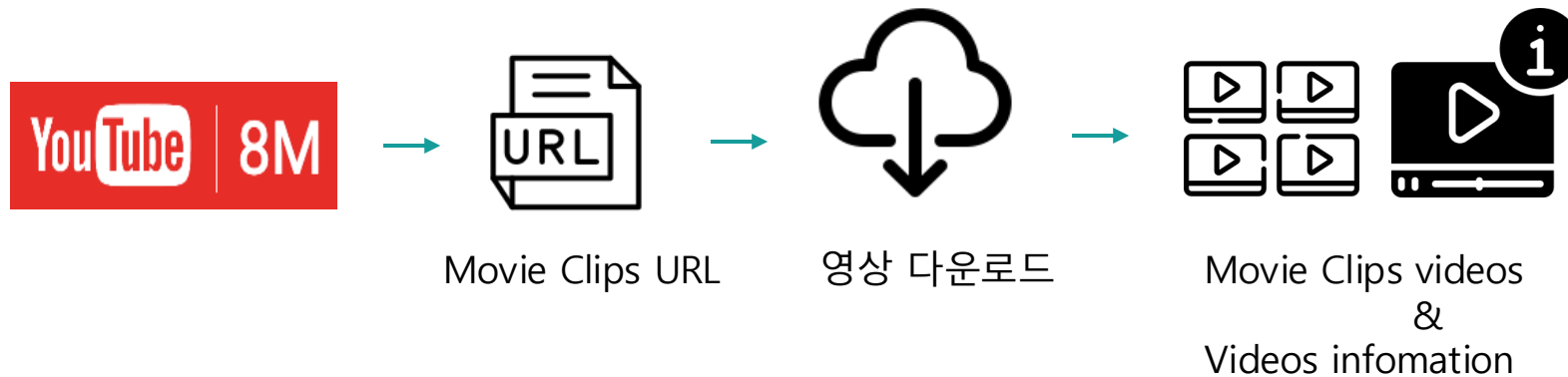
데이터 수집 및 분석

데이터 수집

YouTube-8M Movie Clips Dataset

비디오 데이터 다운로드

- GitHub에서 YouTube-8M 데이터의 **Movie Clips** 분야 URL 정보를 수집했다. ([github](#))
- 수집한 URL을 통해 유튜브 동영상을 직접 다운로드 했다.



데이터 분석

YouTube-8M Movie Clips Dataset

	Duration (초)	FPS	Width (픽셀)	Height (픽셀)
Mean	166.62	24.11	638.94	358.54
Min	97.62	18.00	320.00	240.00
Max	487.94	30.00	640.00	360.00

결과적으로, 접근 불가능한 영상들을 제외하고 총 1,218개의 영상을 다운로드했다.

다운로드한 영상에 대해서 총계 값을 분석해 보았고, 영상들에 대한 평균 길이는 166.62초로, 평균적으로 3분 미만의 영상이고, 해상도는 638.94×358.54 로 낮은 해상도를 가진다.

영상 평균 길이는 데이터 처리 시간 예측에 사용하였고, 평균 FPS(24.11)는 일관된 프레임 유지, 평균 해상도는 모델 입력 크기 조정 및 메모리 최적화에 활용되었다.

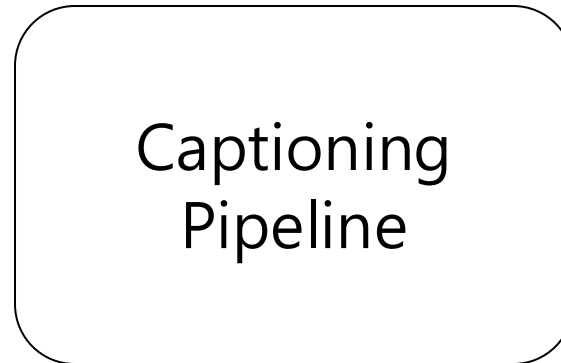


Video to Text

Video to Text

단순화 된 파이프라인

사용자는 비디오 정보를 [(video_id_1, timestamp_start, timestamp_end), (video_id_2, timestamp_start, timestamp_end), ...]와 같은 형태로 입력한다. 이후 파이프라인은 이 정보를 바탕으로 비디오를 작은 클립으로 분할하고, **각 클립에 대해 캡셔닝 파이프라인을 통해 자동으로 캡션을 생성한다.**



비디오에 대한 캡션:

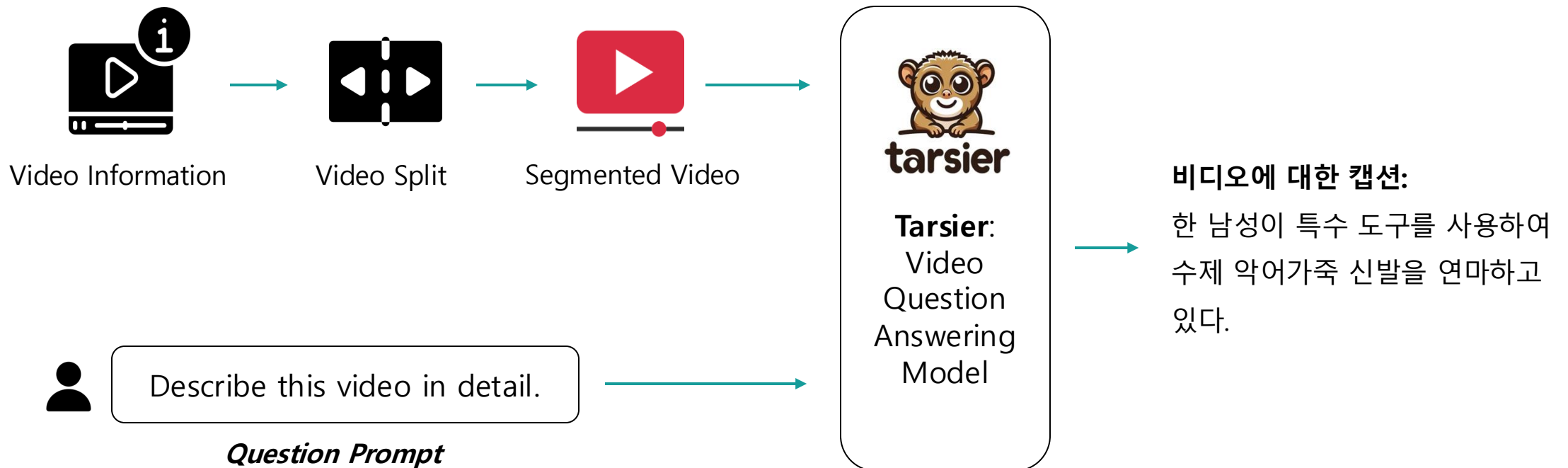
한 남성이 특수 도구를 사용하여
수제 악어가죽 신발을 연마하고
있다.

Video to Text

파이프라인 구체화

비디오를 작은 클립으로 나누면 장면별 캡션을 생성할 수 있다. **Tarsier**는 비디오의 영상, 텍스트를 분석해 질문에 답하는 모델이다.

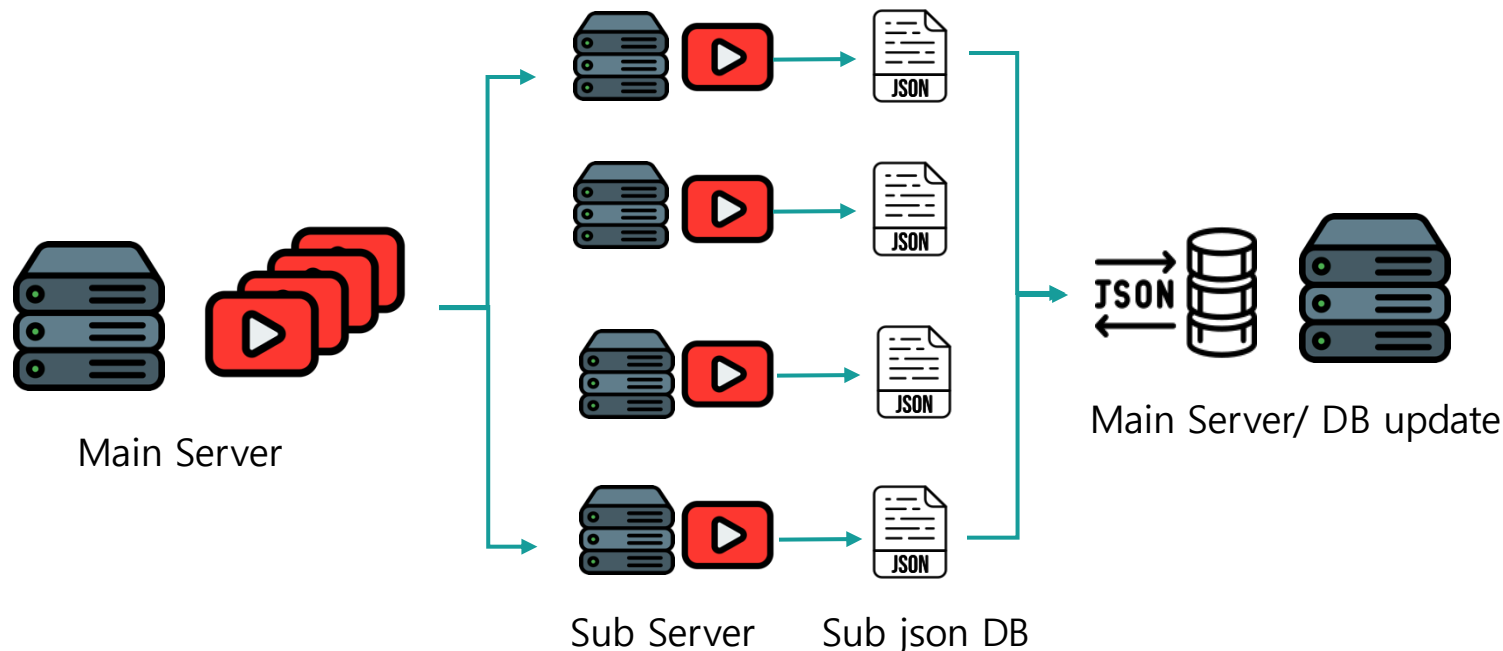
입력 명령어로 "Describe this video in detail." 같은 질문을 할 수 있으며, 모델은 장면별로 캡션을 생성해 자연어로 설명한다.



Video to Text

분산 병렬 처리

효율적인 처리를 위해 메인 서버의 비디오가 업로드되면, 이를 **4개의 서버로 분할하여 처리**한다. 각 서버는 할당된 비디오 부분을 분석하고 캡션을 생성한 후, JSON 형식으로 저장한다. 생성된 JSON 데이터는 다시 메인 서버로 전송되며, **메인 서버는 이를 병합하여 최종 결과를 생성**한다. 이후, 이 데이터를 바탕으로 데이터베이스를 업데이트하여 검색 및 분석을 가능하게 한다. 이 방식은 **분산 처리**를 통해 빠른 캡셔닝이 가능하며, **JSON 포맷**을 활용해 유연하게 데이터를 관리할 수 있다.



Video to Text

Captioning 모델 성능 평가 기준

정성 평가 : 영상을 샘플링해서 5명의 평가자가 정확성, 포괄성, 간결성 평가한다.

정량 평가 : 영상 하나의 caption을 생성하는 데 걸리는 시간 측정한다.

정성 평가 요소	설명
정확성 (5점)	Caption이 영상을 얼마나 정확하게 설명하는지 평가
포괄성 (5점)	[장소], [배경], [인물 외형], [인물 동작], [인물 간 상호 동작] 등의 정보 포함 여부
간결성 (5점)	Caption 길이에 대한 필요한 정보의 비율, 얼마나 간결하게 영상을 설명하는지 평가

정량 평가 요소	설명
소요시간	Caption을 생성하는 데 걸리는 시간

Video to Text

Captioning 정성 평가

mPlug-Owl3

영상에서 한 남성이 방에 서서 **총을 들고 카메라를 향해 겨누고 있는 모습**이 보입니다. 그런 다음 방아쇠를 당기면 총이 발사됩니다. 남성은 비명을 지르며 **팔을 마구 흔드는 모습**이 목격되어 공포나 공포에 휩싸인 것으로 보입니다. 이 남성의 의도가 무엇인지는 불분명하지만 **위험한 상황에 처해 있다는 것**은 분명합니다. 영상에는 남성이나 장면이 촬영된 위치에 대한 추가적인 맥락이나 정보는 제공되지 않습니다.

Tarsier-7B

한 사람이 **녹색 벽과 식물이 있는 방**에 있습니다. 그 사람은 **막대기 같은 물체를 들고 있으며** 말을 하거나 제스처를 취하고 있는 것처럼 보입니다. 그런 다음 그 사람은 물체를 **머리 위로 들어 올려 사용을 준비**하는 것처럼 보입니다.

InternLM-XComposer

비디오에는 **곱슬머리와 수염이 있는 남자가 녹색 벽과 식물이 있는 방**에 있는 모습이 보입니다. 그는 **흰 셔츠 위에 검은색 코트를 입고** 있으며, **무기처럼 보이는 검은색 물체**를 들고 있습니다. **처음에는 그 물체를 들고 무언가를 말하거나 외치는 듯한** 모습입니다. 시간이 지남에 따라 그의 표정은 점점 더 강렬 해지며, **두 팔을 들어 그 물체를 꼭 쥐고 있습니다**. 그의 크게 뜬 눈과 벌어진 입은 **극도의 감정**이나 **놀람 상태**를 나타냅니다. 장면은 배경이나 남자의 옷차림에 큰 변화 없이 일관성을 유지합니다.



파랑 : 정확하게 묘사함

빨강 : 틀리게 묘사함

초록 : 디테일한 정보를 묘사 못함

Video to Text

Captioning 모델 성능 평가

5명의 평가자가 모델에 대한 정성평가를 진행한 결과 Tarsier-7B 모델의 정성 평가 성능이 가장 높다.

mPlug-Owl3의 경우 소요 시간이 가장 짧지만, 정성평가 점수가 낮아 선택하지 않았다.

최종적을 Tarsier-7B 모델이 정성 평가 성능이 가장 좋았으며, 소요 시간도 적게 걸렸다.

모델	정성 평가				정량 평가
	정확성 ↑	포괄성 ↑	간결성 ↑	평균 ↑	소요시간(초) ↓
mPLUG-Owl3	1.4	1.0	2.0	1.3	<u>5.43</u>
InternLM-XComposer	<u>3.4</u>	4.8	<u>4.0</u>	<u>4.13</u>	29.37
Tarsier-7B	4.2	<u>4.2</u>	4.8	4.3	6.76



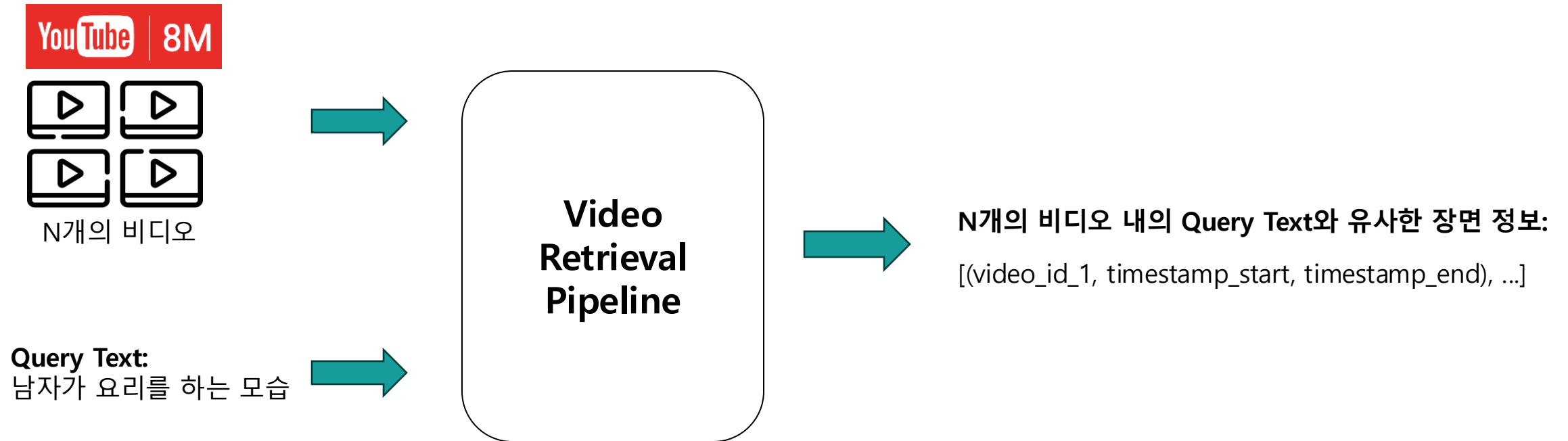
Text to Video

Text to Video

단순화 된 파이프라인

YouTube 8M와 같은 대규모 데이터셋과 새로운 N개의 비디오 데이터를 검색 DB로 사용한다.

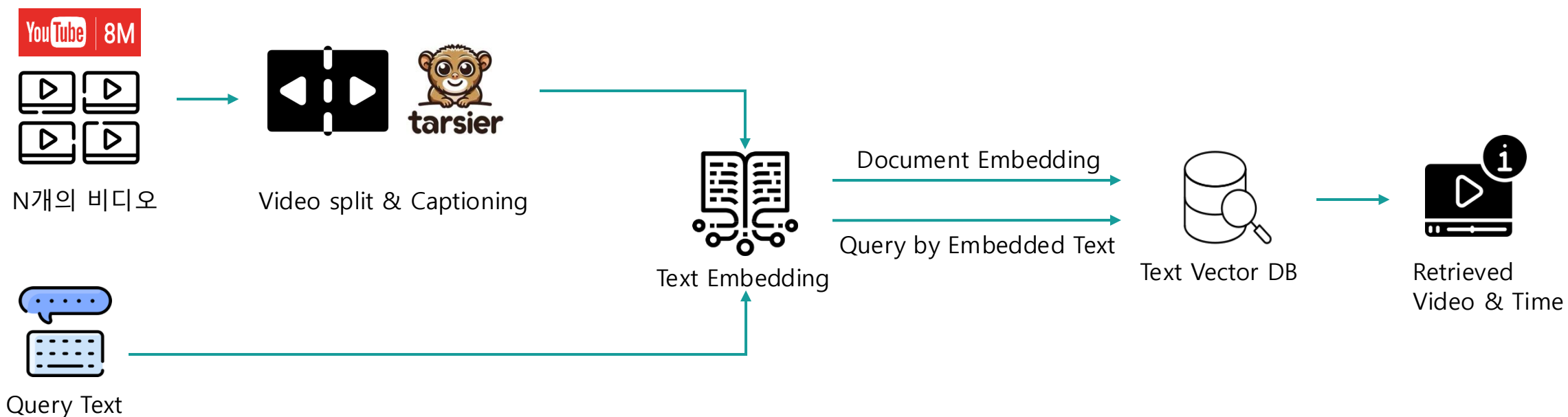
검색된 텍스트 임베딩을 바탕으로 가장 유사한 비디오와 시간 정보를 반환하여 사용자가 특정 문장을 검색했을 때 관련된 비디오 클립을 바로 찾을 수 있다.



Text to Video

파이프라인 구체화

비디오는 작은 클립으로 분할된 후, 각 클립에 대한 설명(캡션)이 생성된다. 이 과정에서 Ffmpeg와 같은 도구들이 활용된다. 생성된 캡션은 **Transformer 기반 문장 임베딩 모델**을 사용해 벡터로 변환된다. 사용자의 검색 문장도 같은 방식으로 임베딩되어 **텍스트 벡터 데이터베이스**에 저장된다. 이를 통해 사용자가 입력한 쿼리는 **FAISS** 같은 벡터 검색 엔진을 사용해 쿼리와 **가장 유사한 캡션과 비디오 정보**를 검색할 수 있다.



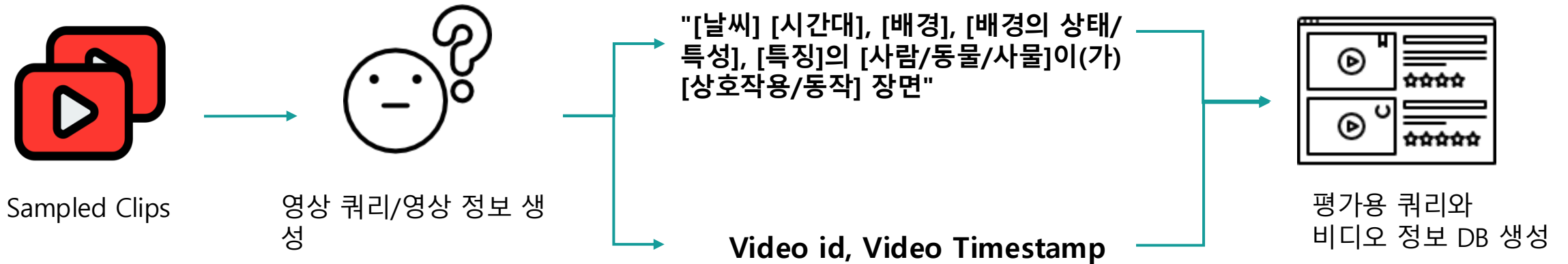


Evaluation

Evaluation

자체 평가용 쿼리 생성 파이프라인

평가자는 전체 영상에서 일부 영상을 샘플링하여 특정 템플릿에 맞춘 쿼리 생성했고, 해당 쿼리와 관련된 비디오 ID 및 timestamp 정보를 이용해 평가용 데이터를 생성했다. 이를 통해, 쿼리와 비디오의 관련성을 평가하고, 각 비디오 클립의 정확한 위치를 추적할 수 있는 데이터를 구축했다.



Evaluation

자체 평가용 쿼리 생성 기준

모델이 학습하기 쉬운 일관된 표현을 유지하기 위해 템플릿을 다음과 같이 구성했다.

"[날씨] [시간대], [배경], [배경의 상태/특성], [특징]의 [사람/동물/사물]이(가) [상호작용/동작] 장면" 이를 활용하여 구조를 통일했다.

환경적 요소: 날씨 및 시간대 (예: "비 오는 낮")

배경 및 상태: 배경 및 바닥 상태 (예: "시멘트 바닥과 나무")

주요 객체 및 특징: 등장인물 및 의상 (예: "청자켓을 입은 남자")

행동 및 상호작용: 인물 간의 관계 및 동작 (예: "한 우산을 같이 쓰고 있는 장면")

=> "비 오는 낮, 시멘트 바닥, 배경에 나무, 청자켓을 입은 남자와 여름 교복을 입은 여학생이 한 우산을 같이 쓰고 있는 장면"



Evaluation

쿼리 임베딩 모델 성능 평가 기준

- 1) 전체 쿼리에 대하여 정답에 해당하는 영상구간이 유사도 Top N (상위 N개)에 들어온 비율

$$Recall@N = \frac{\text{정답을 포함한 쿼리의 개수 (상위 } N \text{ 개 결과 내)}}{\text{전체 쿼리 개수}}$$

- 2) 전체 쿼리에 대하여 Top 1으로 검색된 캡션의 유사도 평균

$$\text{평균 유사도} = \frac{\sum(\text{Top} - 1 \text{ 검색 결과와 입력 쿼리의 유사도})}{\text{전체 쿼리 개수}}$$

- 3) 전체 쿼리에 대하여 정답구간 캡션이 검색된 순위의 평균

$$\text{평균 순위} = \frac{\sum(\text{입력 쿼리에 대한 정답의 순위})}{\text{전체 쿼리 개수}}$$

Evaluation

쿼리 임베딩 모델 성능 평가

모델	Recall@20 ↑	평균 유사도↑	평균 순위 ↓
stella-en-1.5B-v5	24.06%	0.6397	4.72
all-MiniLM-L6-v2	48.66%	0.5983	5.64
all-mpnet-base-v2	50.80%	0.6284	4.71

* 187개의 GT 기준

- Tarsier-7b로 생성된 캡션을 기준으로, 쿼리 생성 규칙에 맞춰 187개의 캡션-쿼리 쌍을 직접 작성했다. 이후, 해당 쿼리에 대한 캡션이 유사도 Top 20 내에서 검색되는지를 실험했다.
- 임베딩 모델 평가를 위해 세 가지 지표를 고려했으며, **Stella-em-1.5B-v5**는 평균 유사도가 높았으나 다른 지표에서 낮은 성능을 보여 사용하지 않았다.
- **all-mpnet-base-v2**는 평균 유사도가 다소 낮았지만, **Recall@20**과 **평균 순위**에서 높은 성능을 보여 최종적으로 선택했다.

Evaluation

쿼리 임베딩 모델 훈련 성능 비교

모델(all-mpnet-base-v2)	Recall@3 ↑	Recall@1 ↑	평균 유사도 ↑
Base	27.81%	19.25%	0.6612
Trained	34.22%	20.86%	0.7467

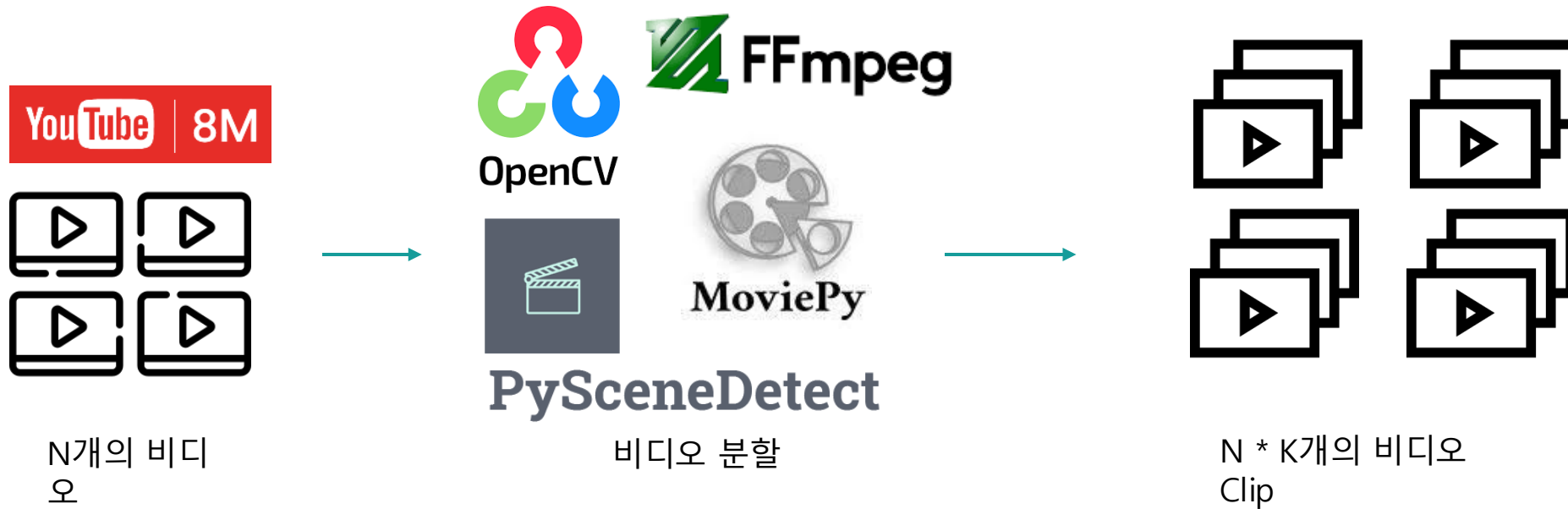
* 187개의 GT 기준

- 임베딩 모델을 fine-tuning 하기 위해, Tarsier-7b로 생성된 캡션을 기반으로 쿼리 생성 규칙에 맞춰 429개의 캡션-쿼리 쌍을 생성했다. 이 과정에서, 생성된 캡션과 유사한 쿼리를 사람이 직접 작성하여 학습 데이터로 사용했다.
- 이는 우리 데이터셋에 맞게 임베딩 모델의 벡터 표현을 학습시키기 위해서다. 이후, 429개 쌍을 Positive Sample로 활용하여 모델을 훈련했고, 평가에는 기존의 187개 캡션-쿼리 쌍을 사용했다.
- 그 결과, Recall@3이 약 **7%**, Recall@1이 약 **1%** 상승했으며, 평균 유사도는 약 **15%** 증가했다.

Evaluation

비디오 분할 파이프라인

아래 파이프라인을 통해 권장 데이터셋을 이용해 데이터베이스(DB) 생성했다.
비디오를 클립 단위로 분할하고 이를 저장했다.



Evaluation

비디오 분할 방법 설명

1) PySceneDetect (Scene Detector 활용)

- **ContentDetector**: HSV 색 공간에서 픽셀 변화의 가중 평균을 이용해 장면 전환을 감지하는 방법
- **AdaptiveDetector**: HSV 색 공간에서 차이의 이동 평균을 사용해 빠른 움직임이 있는 장면에서도 안정적으로 감지하는 방법

2) OpenCV (Shot Boundary 감지)

- RGB -> Gray scale
- 프레임 간 픽셀 차이 계산
- Threshold 보다 크면 새로운 장면으로 감지

3) FFmpeg (단순 초 단위 분할)

- 3초 간격으로 분할
- 5초 간격으로 분할
- 7초 간격으로 분할

Evaluation

비디오 분할 방법 성능 평가

권장 데이터 셋에서 영상 108개를 샘플링 해서 각각 분할 방법으로 DB 만든 후 자체 평가용 GT 쿼리로 실험을 진행했다.
지표를 고려했을 때 단순 분할 방식(FFmpeg)이 가장 우수하다.

Split Method		Recall@1 ↑	Recall@5 ↑	Recall@10 ↑	Recall@15 ↑	Recall@20 ↑
PysceneDetect	Content	22.99	48.66	56.68	62.57	64.17
	Adaptive	25.13	47.59	58.29	63.64	66.31
OpenCV		25.67	48.13	57.22	63.10	65.24
FFmpeg	3s	29.95	54.55	64.71	70.05	74.33
	5s	26.74	52.94	65.24	73.80	75.40
	7s	28.34	52.41	64.17	67.38	71.12

Evaluation

최종 DB 성능 평가

모든 실험 결과를 참고하여 최종 DB를 구축했다.

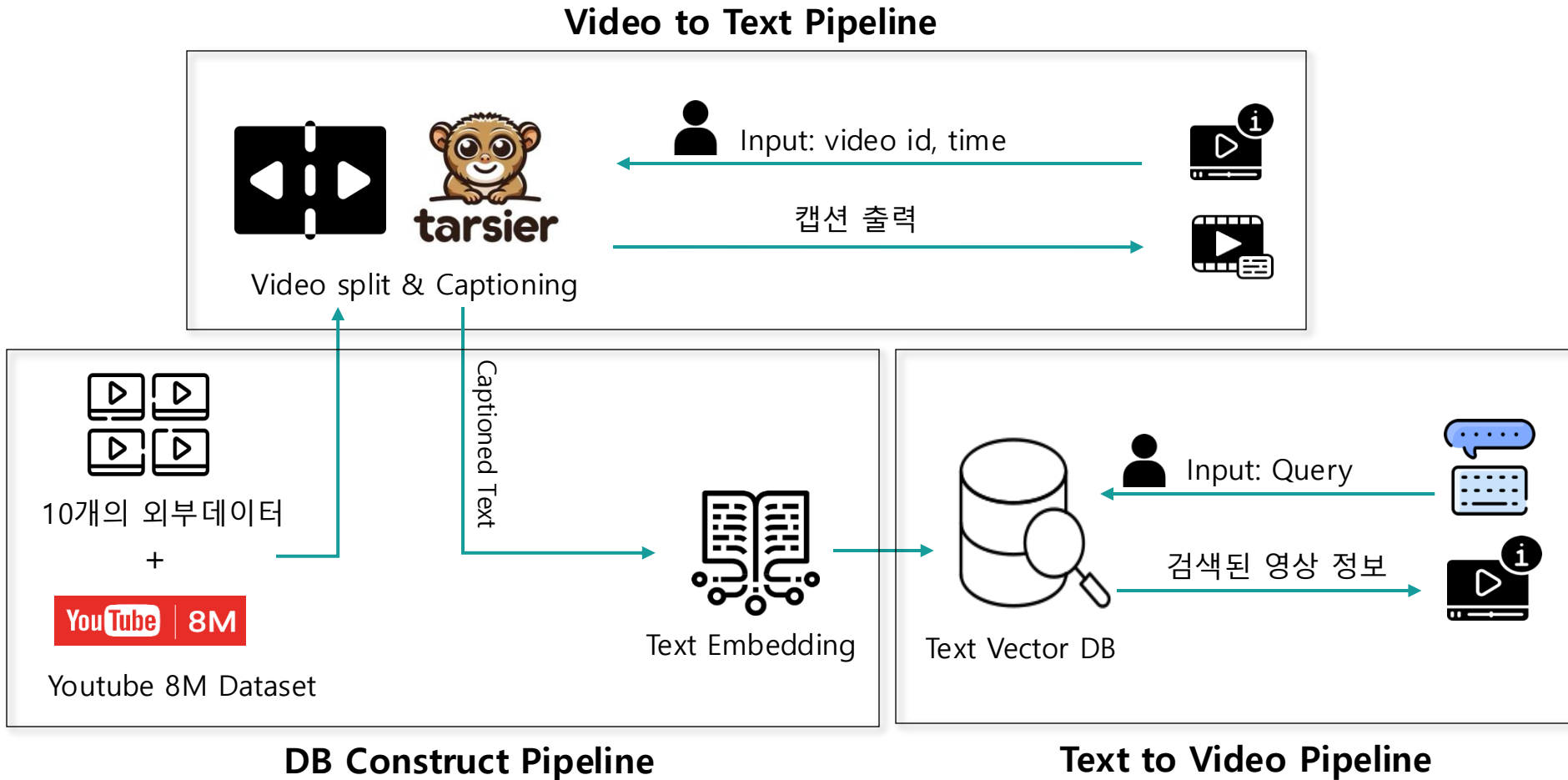
DB 비디오 종류	Embedding Model	Split Time	Recall@1 ↑	평균 유사도
108	all-MiniLM-L6-v2	3s + 5s	33.69	0.6509
		3s + 7s	32.09	0.6548
		5s + 7s	34.22	0.6320
		3s + 5s + 7s	35.29	0.6577
1218 (All)	all-MiniLM-L6-v2	5s	12.30	0.6255
		3s + 5s	14.44	0.6578
	all-mpnet-base-v2	5s	16.58	0.6273
		3s + 5s	19.25	0.6612
	Trained model	3s + 5s	20.86	0.7683
		3s + 5s + 7s	21.39	0.7656



Result

Result

최종 파이프라인



Result

Video to Text Web 시연 영상

비디오 검색

비디오 → 텍스트

Video Captioning 활성화 ☐

video_516

35

40

video_200

100

105

video_

시작 시간 입력

종료 시간 입력

+ 추가

삭제

전체 삭제

검색

검색 결과 1 / 전체 3 클립

비디오 제목: The Rules of Attraction (10/10) Movie CLIP - People Like Us (2002) HD

유사도: 0.7852

비디오: 00011.mp4

구간: 50.00초 ~ 55.00초

캡션: 두 사람이 눈 덮인 거리에서 나란히 걷고 있습니다. 배경은 큰 조명으로 장식된 출입구와 계단이있는 건물을 보여줍니다. 눈이 꾸준히 떨어지고 있으며 두 사람은 어두운 옷을 입고 있습니다. 남자는 때때로 손을 입에 가져다가 여자는 그 옆에 걸어 간다.

검색 결과 2 / 전체 3 클립

비디오 제목: Cold Mountain (9/12) Movie CLIP - Reunited (2003) HD

유사도: 0.7602

비디오: 00018.mp4

구간: 85.00초 ~ 90.00초

캡션: 두 사람이 배경에 바위 절벽이있는 눈 덮인 야외 환경에서 걷고 있습니다. 한 사람이 검은 코트와 넓은 모자를 쓰고 가방을 등에 들고 있습니다. 다른 사람은 또한 어두운 옷을 입고 있습니다. 카메라는 걷고 상호 작용할 때 두 개인에게 초점을 맞추는 것 사이를 번갈아 가며 교대합니다.

검색 결과 3 / 전체 3 클립

비디오 제목: The Rules of Attraction (10/10) Movie CLIP - People Like Us (2002) HD

유사도: 0.7328

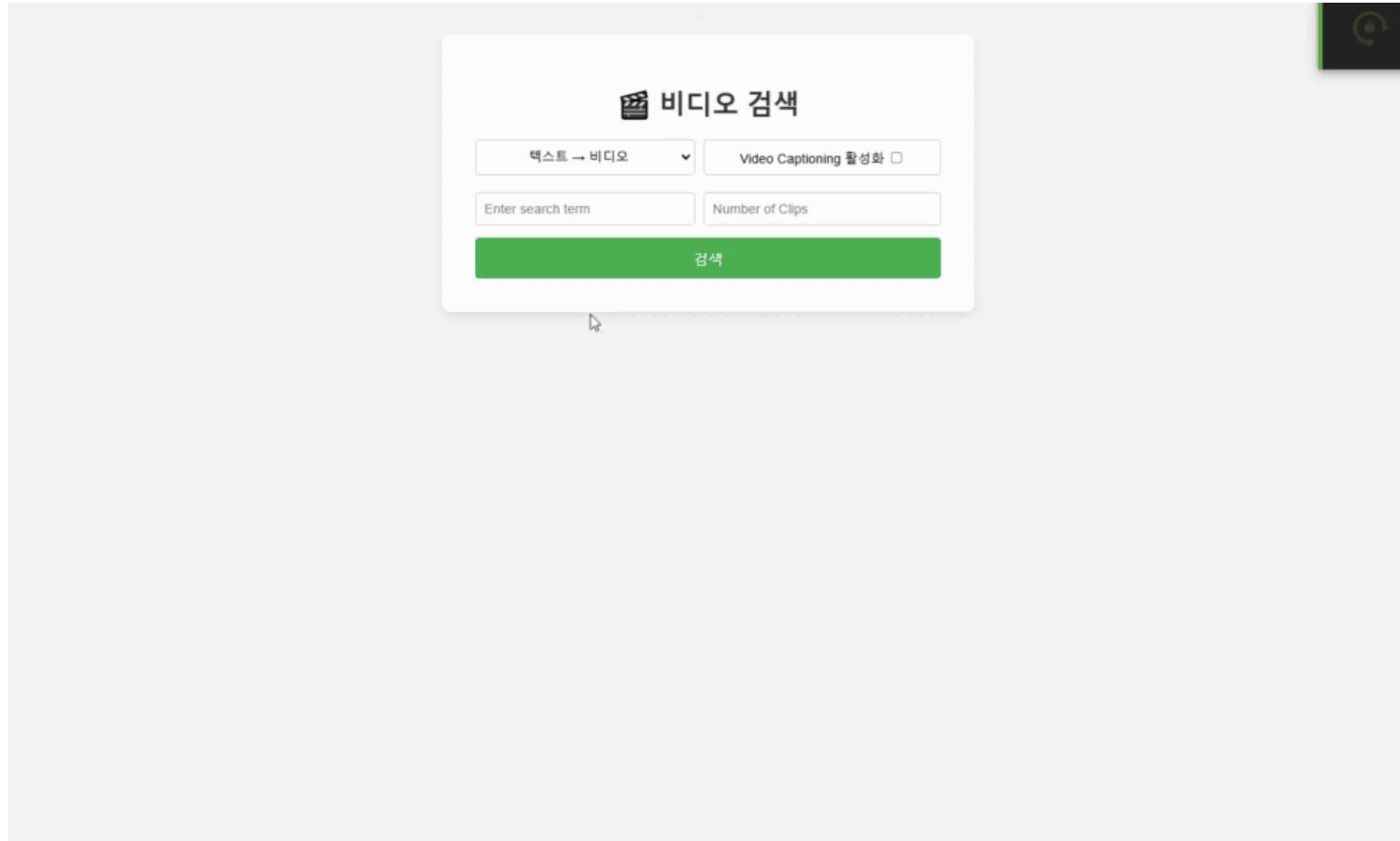
비디오: 00013.mp4

구간: 60.00초 ~ 65.00초

캡션: 두 사람이 밤에 눈 덮인 거리에서 나란히 걷고 있습니다. 그들은 어두운 옷을 입고 있습니다. 배경에는 조명된 창문과 장식 조명이있는 건물이 있습니다. 땅은 눈으로 덮여 있으며 눈은 계속해서 현장 전체에 떨어집니다.

Result

Text to Video Web 시연 영상



The screenshot shows a web interface for video search. At the top, there is a title "비디오 검색" (Video Search) with a clapperboard icon. Below the title, there are two input fields: "텍스트 → 비디오" (Text → Video) with a dropdown arrow, and "Video Captioning 활성화" (Video Captioning Activation) with a checkbox. Below these, there are two more input fields: "Enter search term" and "Number of Clips". At the bottom, there is a large green button labeled "검색" (Search). The interface is set against a light gray background.



자체 평가 의견

자체 평가 의견

잘했던 점

- 초반에 설계한 파이프라인을 모두 완성하였기 때문에 달성도 측면에서는 훌륭하게 완수했다고 생각된다.
- 또한, 파이프라인을 완성한 후에 개선사항을 계속해서 생각하여 완성도를 높였다.
- 예를 들어, 단순한 5초 캡셔닝 DB에 3초, 7초 DB를 추가하여 검색 성능을 높인다거나, 임베딩 모델을 fine-tuning 하여 검색 성능 높였다.
- GPU 분산처리를 이용하여 추론 속도를 획기적으로 상승시켰다.
- 직접 데이터를 생성하는 작업을 통해 데이터의 이해도를 높이고, 라벨의 정확도를 높였다.
- Web Demo를 구현하여, 파이프라인의 시연을 보기 쉽게 하였다.

자체 평가 의견

아쉬운 점

1) 오디오 캡셔닝, 자막 추가

현재 평가 기준은 "시각 정보" 만에 대한 캡셔닝 및 검색 성능이다.

그러므로 오디오 정보가 주어진다면, 오히려 노이즈가 될 것으로 판단하여 오디오 정보를 제거했다.

하지만, 실제 서비스 상황에서는 오디오 정보(캡셔닝, 자막)를 추가한다면 더 나은 사용자 경험을 제공할 수 있을 것이다.

2) 등장인물 검색

실제 서비스에서 사용자가 검색하는 경우, 영상 내 등장인물의 이름을 통해 검색하는 경우가 존재한다.

이 경우 연예인 이미지를 학습시킨 Multi Label Classification 모델 사용하면 비디오의 등장인물을 파악하는 것이 가능하다.

3) fine-tuning

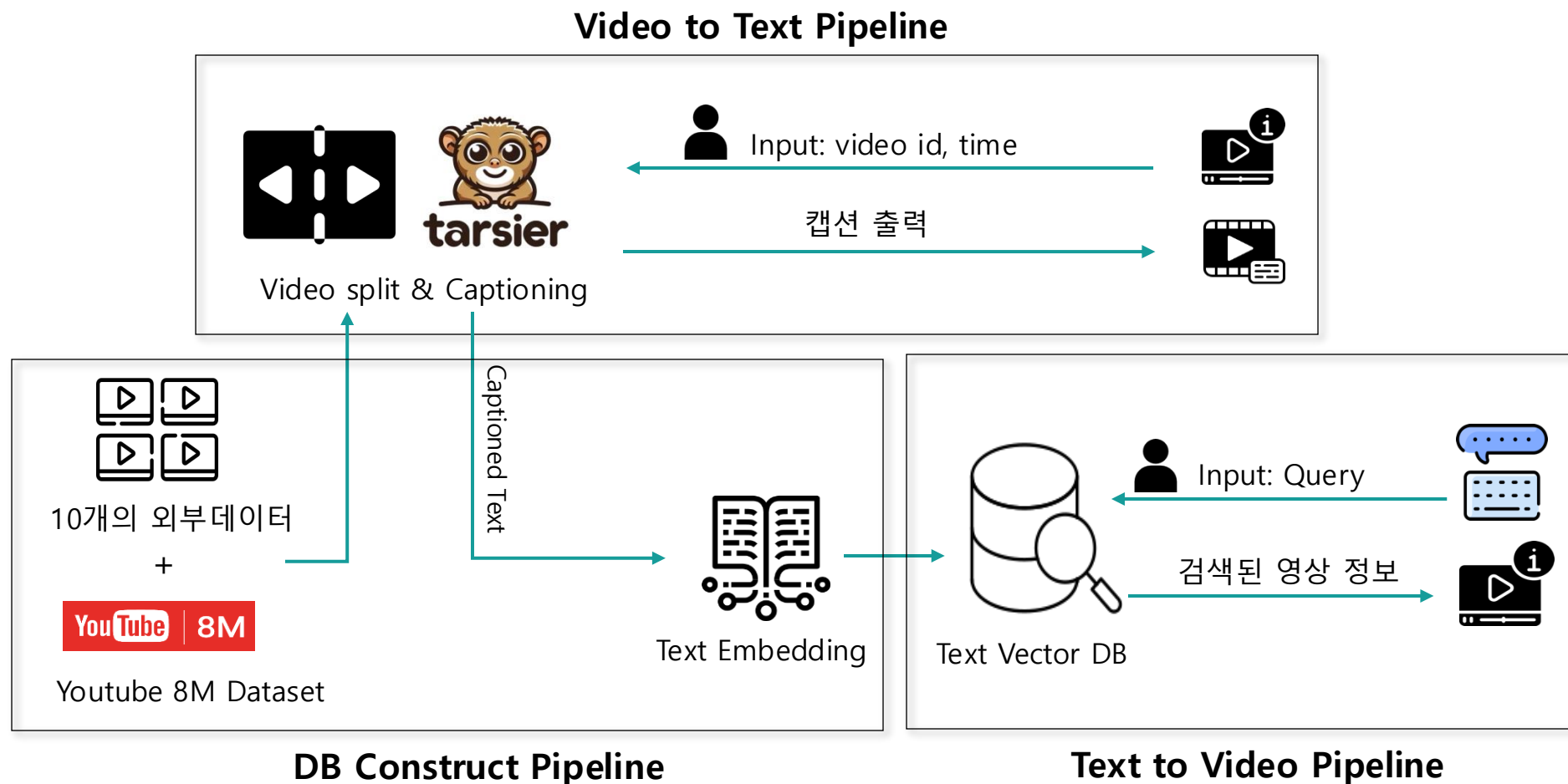
모델을 학습시키기 위해 대용량 데이터를 labeling 하는 작업은 큰 비용을 요구한다. 정확한 모델을 통해 대용량 데이터를 labeling 하거나 self-supervised learning 기법을 사용하여 모델을 fine-tuning 한다면 큰 labeling 비용 없이 모델의 성능을 개선할 수 있다.



별첨

별첨

최종 파이프라인 ([github](#))





감사합니다.

Q&A

CV-15

