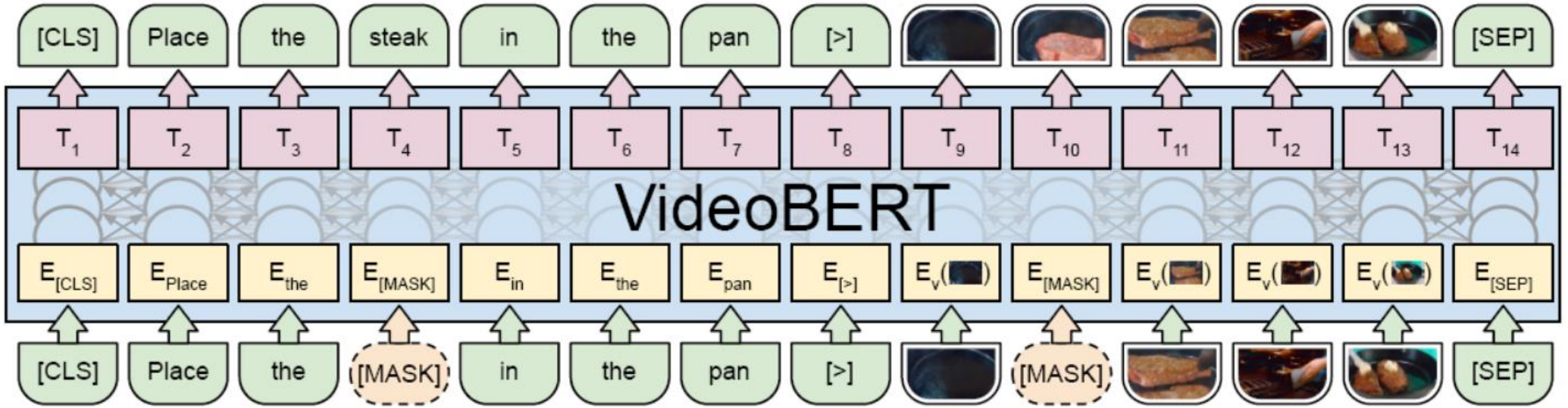
COOT:
Cooperative Hierarchical Transformer for Video-Text Representation Learning

집현전 최신반 8조
이나연(발표자), 민지웅, 임정환

CONTENTS

논문 선택 이유



<VideoBERT의 구조>

- 기존의 연구에서는 Text embedding과 Video-embedding을 단순히 결합하는 방법(예시- VideoBERT)으로 Video-text joint embedding을 다루었음
- Text, video의 다양한 수준의 정보를 더욱 효과적으로 반영할 수 있는 모델이 있지 않을까?

➢ **계층적 트랜스포머를 도입해 Joint embedding space를 효과적으로 학습**하는 방법 도입!

01 Introduction

- Long-Range Temporal Dependency 문제에 초점

⇒ joint cross-modal embedding을 학습할 때, 비디오와 텍스트 모두에서 long-range temporal context를 활용할 수 있는 계층적 모델 제안

Contribution

1) 새로운 attention-aware feature aggregation layer, contextual attention module을 가진 계층적 트랜스포머 구조 제안
2) Joint embedding space에서 비전과 텍스트의 의미적 정렬을 향상시키는 cross-modal cycle-consistency loss 제안
3) Video-text 검색에서 SOTA 달성

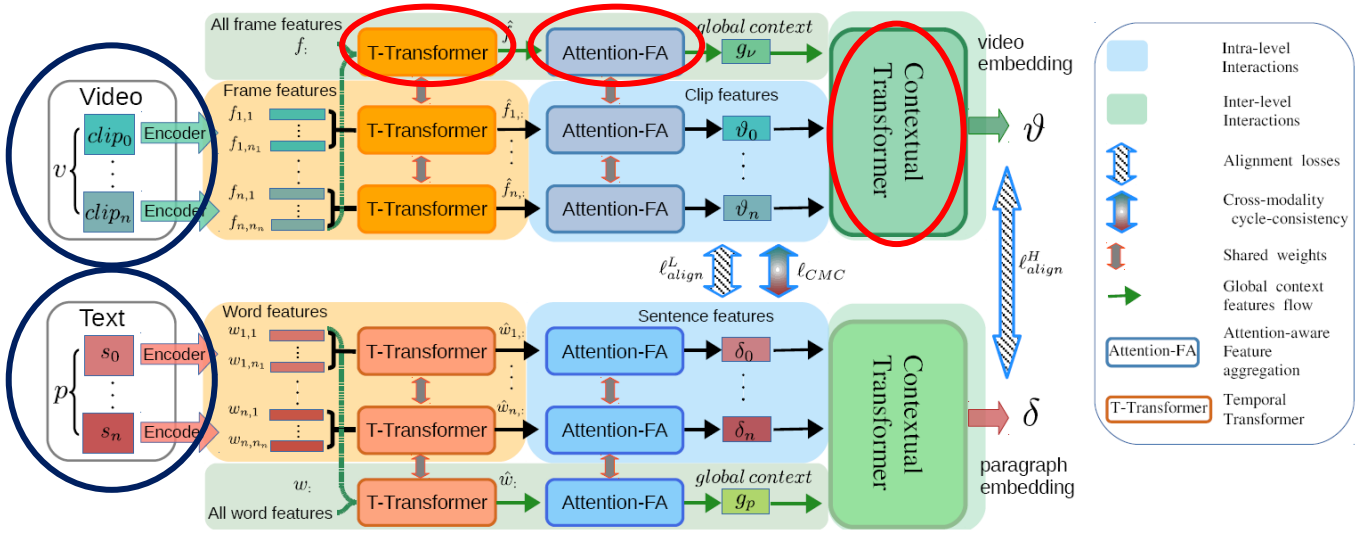
02 Cooperative Hierarchical Transformer

비디오와 텍스트를 여러 수준으로 세분화

- **텍스트 : 문단 → 문장 → 단어**
- **비디오: 비디오 → 클립 → 프레임**

=> 세분화 된 의미 구조를 포착할 수 COOT Model 제안

1.프레임/단어 feature 간의 관계를 포착하는 Temporal Transformer,
2.클립/문장 feature 생성을 위한 Attention-aware feature aggregation
3.비디오 및 텍스트 임베딩 생성을 위한 Context Transformer



02 Cooperative Hierarchical Transformer

2. 1.1 Semantic Alignment Losses

- 다른 세분화 수준의 representation을 정렬하기 위해 Zhang et al.이 제안한 alignment loss 사용
- 비디오-텍스트 정렬의 경우, contrastive loss을 활용하여, 양수 샘플은 가깝게 음수 샘플은 멀리 떨어지게 함

$$L(\mathcal{P}, \mathcal{N}, \alpha) = max(0, \alpha + D(x, y) - D(x', y)) + max(0, \alpha + D(x, y) - D(x, y'))$$

$D(x, y) = 1 - \frac{x^T y}{||x|| ||y||}$ 은 두 벡터 간의 코사인 거리

- clip-sentence , video-paragraph, global context 수준에서 표현을 정렬하기 위해 다음과 같은 손실 함수 사용

$$\ell^L_{align} = \sum_{k \in \mathcal{D}, i, k' \neq k, i' \neq i} L((\vartheta^k_i, \delta^k_i), \{(\vartheta^k_i, \delta^{k'}_{i'}), (\vartheta^{k'}_{i'}, \delta^k_i)\}, \beta)$$

$$\ell^H_{align} = \sum_{k \in \mathcal{D}, k' \neq k} L((\vartheta^k, \delta^k), \{(\vartheta^k, \delta^{k'}), (\vartheta^{k'}, \delta^k)\}, \alpha)$$

$$\ell^g_{align} = \sum_{k \in \mathcal{D}, k' \neq k} L((g^k_v, g^k_p), \{(g^k_v, g^{k'}_p), (g^{k'}_v, g^k_p)\}, \alpha_g)$$

δ^k_i: k번째 영상의 i번째 클립에 대한 임베딩
θ^k_i: k번째 단락의 i번째 문장의 임베딩
α, α_g, β는 constant margin

02 Cooperative Hierarchical Transformer

2.1.1 Semantic Alignment Losses

- 텍스트-비디오 joint embedding space에서 저차원, 고차원 의미를 클러스터링 하기 위한 추가적인 loss function 사용

$$\ell_{cluster} = \sum_{k\in\mathcal{D},i,k'\neq k,i'\neq i} L((1,1),\{(\vartheta_i^k,\vartheta_{i'}^{k'}),(\delta_{i'}^{k'},\delta_i^k)\},\gamma) + \sum_{k\in\mathcal{D},k'\neq k} L((1,1),\{(\vartheta^k,\vartheta^{k'}),(\delta^{k'},\delta^k)\},\eta)$$

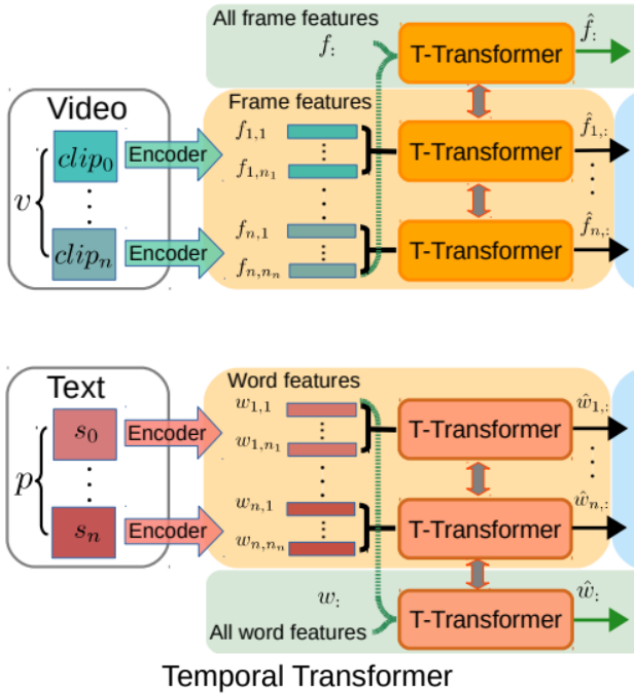
- γ, η: constant margin

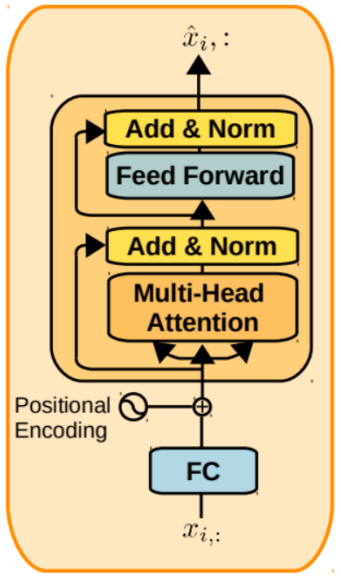
- (1,1): 양수 샘플은 변하지 않음
- 이 손실의 목표는 **음수 샘플에 대한 임베딩을 멀어지게 하는 것**

02 Cooperative Hierarchical Transformer

2.1.2 Temporal Transformer

- 프레임, 단어 representation을 배우기 위해 **표준 attention-block** 사용
- 2개의 temporal transformers을 학습: **video, text 하나씩**
- 각각의 branch의 T-Transformer는 **가중치를 공유**
- 이 모듈은 temporal features간의 관계를 포착하여 향상된 representation 생성
- 비디오 v_k가 주어지면 모든 프레임을 인코딩하여, 프레임 수준 feature $\{f_{i,:}^k\}_{i=1}^n$을 얻음
- f^k는 비디오 v_k에 대한 i번째 클립의 모든 프레임 수준 feature (주황색 부분)
- Global context 계산을 위해 비디오의 모든 프레임 feature $f_:^k$을 사용(녹색 부분)
- 최종적으로 $\{\hat{f}_{i,:}^k\}_{i=1}^n$, $\hat{f}_:^k$를 산출





02 Cooperative Hierarchical Transformer

2.2 Intra-Level Cooperation

- 일반적으로 feature fusion 방법으로 average pooling or max pooling을 활용

  -> 연관 있는 feature를 강조하기 위한 **feature 간의 관계**를 놓침

- 트랜스포머 모델은 [CLS] token, average pooling을 활용

Attention-aware feature aggregation module

- $X = \{x\_1, \ldots, x\_T\}$ (즉 $f\hat{}k\_\{i,:\} = \{f\hat{}k\_\{i,1\}, \ldots, f\hat{}k\_\{i,T\}\}$)로 표시되는 T개의 feature 벡터가 있을 때

  Attention matrix A는 $A = softmax\left(W_{2Q} + b_2\right)^T$, $Q = GELU(W_1 K^T + b_1)$

- Final feature: $\hat{x} = \Sigma_{\{i=1\}}^{T} a_i \odot x_i$, a_i: i번째 feature에 대한 A의 i-th attention vector
- **모듈의 역할**
  - (1) 쿼리(Q)와 키(K)로 학습 가능한 가중치 (W_1, W_2)로 사용하고, 계산된 score를 기반으로 집계
  - (2) 쿼리는 변환된 키(K)와 같으며, 활성화 함수 GELU를 적용
- $\{f^k\}_{i=1}^n$과 f^k을 입력으로 받아, 클립 수준$a\{\theta_i^k\}_{i=1}^n$ feature와 비디오의 global context g_v를 출력

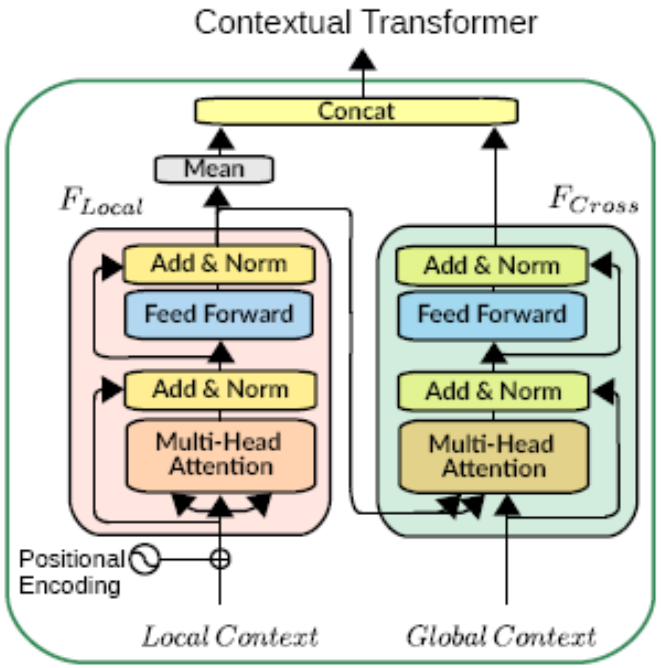
02 Cooperative Hierarchical Transformer

2. 3 Inter-Level Cooperation

- Local context와 global context간 상호 작용을 모델링
- 비디오의 general context와 관련된 의미는 강조하고, 관련 없는 의미를 억제하는 방향으로 학습

Contextual Transformer

- Low-level semantics와 High-level semantics 사이의 관계를 모델링하기 위해 사용
- 2개의 모듈 사용: $\boldsymbol{F_{Local}}$, $\boldsymbol{F_{global}}$
- F_{Local} : Low-level semantics간의 단기 상호작용을 모델링
- F_{global}: 중요한 semantics를 강조하기 위해 local context와 global context간의 상호작용을 모델링



02 Cooperative Hierarchical Transformer

2. 3 Inter-Level Cooperation

- F_Local 은 Local representation $\{\theta_i^k\}_{i=1}^n \in R^{\{n \times d\}}$ (n: number of clip, d: feature dimension) 에 multi-head attention, feed-forward를 순차적으로 적용해 임베딩 생성 $\{\boldsymbol{h_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$
- 임베딩 h_i 과 global context g_v를 기반으로 하여 key-value pairs와 query를 계산
- F_Global은 아래와 같은 attention output을 생성

$$H_{attn} = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d}})\mathbf{V}, \qquad Q = \mathcal{W}_q g_v,\ K = \mathcal{W}_k\{h_i\}_{i=1}^n,\ V = \mathcal{W}_v\{h_i\}_{i=1}^n$$

- H_attn는 feed-forward를 거쳐 contextual 임베딩 $H_{context}$ 생성
- 최종적인 비디오 임베딩 $\theta_k = concat(mean(\{h_i\}_{i=1}^n,\ H_{\{context\}})$

03 Cross-Modal Cycle Consistency

Cross-Modal Cycle Consistency

- 클립과 문장 간의 의미적 정렬을 강화하기 위한 목적
- 클립과 문장이 semantic하게 정렬되었다 -> 한 쌍의 클립과 문장이 학습된 공통 공간에서 가장 가까운 이웃인 경우

클립 임베딩: $\{\boldsymbol{\theta_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}} = \{\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_n}\}$**, 문장 임베딩:** $\{\boldsymbol{\delta_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{m}} = \{\boldsymbol{\delta_i}, \ldots, \boldsymbol{\delta_m}\}$

- 계산 과정

1) soft nearest neighbor $\bar{\theta}_{\delta_i}$를 찾음

$$\bar{\vartheta}_{\delta_i} = \sum_{j=1}^{n} \alpha_j \vartheta_j \quad \text{where} \quad \alpha_j = \frac{\exp(-\|\delta_i - \vartheta_j\|^2)}{\sum_{k=1}^{n} \exp(-\|\delta_i - \vartheta_k\|^2)}$$

α_j: 클립임베딩 θ_i, 문장임베딩 δ_i사이의 유사도 점수

2) 소프트 최근접 이웃인 $\bar{\theta}_{\{\delta_i\}}$에서부터 문장 시퀀스 $\{\delta_i\}_{i=1}^{m}$ 순환하고 **소프트 위치(soft location)**를 계산

$$\mu = \sum_{j=1}^{m} \beta_j j \quad \text{where} \quad \beta_j = \frac{\exp(-\|\bar{\vartheta} - \delta_j\|^2)}{\sum_{k=1}^{m} \exp(-\|\bar{\vartheta} - \delta_k\|^2)}.$$

- 문장 임베딩 δ_i가 semantically cycle consistent하다는 것은 원래 위치, 즉 i = μ로 다시 순환하는 것을 의미

03 Cross-Modal Cycle Consistency

- 샘플링된 클립과 문장 세트에 대한 cycle-consistency의 편차에 페널티를 부여하여 **모델이 의미론적으로 일관된 표현을 학습**하게 함

=> **Objective**: source location i 와 soft destination location μ 사이의 거리

$$\ell_{CMC} = \|i - \mu\|^2$$

- Nearest neighbors을 soft nearest neighbors으로 계산하면 손실을 미분할 수 있음
- supervised 시나리오와 self-supervised 시나리오 모두에서 사용 가능
- 과정
  - 각 비디오를 여러 클립으로 균일하게 분할, 각 단락을 문장으로 분할
  - **텍스트에서 비디오로의 cycle-consistency 과 비디오에서 텍스트로의 cycle-consistency** 를 계산합니다.

=> **최종 손실 함수:** $\ell_{final} = \ell^L_{align} + \ell^H_{align} + \ell^g_{align} + \ell_{cluster} + \lambda\ell_{CMC}$

04 Experiments

- Datasets
  - ActivityNet-captions
    - 평균 길이가 2분인 20,000개의 YouTube 동영상, 72,000개의 클립-문장 쌍으로 구성
    - Train: ~10k Val1: ~5k, Val2: ~5k
  - Youcook2
    - 14000개 클립으로 이루어진 2000개의 비디오, 89가지 유형의 레시피를 다룸
    - 각 클립에는 수동으로 주석이 달린 텍스트 설명이 있음
    - Train: ~9.6k Val: ~3.2k
- Evaluation Metrics
  - 표준 검색 메트릭 : K(R@K e.g. R@1, R@5, R@10)에서의 recall, Median Rank(MR)
- Text encoding
  - Token을 BERT-Base, Uncased 에 넣어 마지막 2개 레이어 결과를 사용. 1536-d
- Video encoding
  - ActivityNet-captions : 2048-d feature (Zhang et al.)
  - Youcook2

    1) 4096-d features at 3FPS: 2D(ImageNet에서 사전 훈련된 Resnet-152)와 3D(Kinetics에서 사전 훈련된 ResNext-101 모델[42]) 를 concat (Miech et al., 2019)

    2) 512-d features at 0.6FPS: Howto100m으로 비디오 텍스트 학습에 대해 사전 훈련된 비디오 임베딩 네트워크(Miech et al., 2020)
  - 각 클립을 80 frame features로 샘플링

04 Experiments

- **Training**
  - $\alpha = \alpha_g = \beta = \gamma = \mu = 0.2$
  - Mini-batch: 64
  - Activation function: GELU
  - Hidden size: 384
  - T-Transformer: one self-attention layer
  - Contextual Transformer: 1 self-attention layer, 1 cross-attention layer

- **Video-Language Retrieval**
  - 쿼리로 paragraph를 주고, 데이터베이스에서 가장 관련성이 높은 비디오를 찾는 것
  - 쿼리가 비디오일 수 있으며 작업은 가장 관련성이 높은 단락을 검색하는 것
- **Clip-sentence retrieval**
  - Youcook2: 하나의 문장이 주어진 짧은 비디오 클립을 검색

04 Experiments

Result 1: Ablation study

Table 1: **Ablation study on ActivityNet-captions (val1).** We quantify the individual contributions of the attention-aware feature aggregation (AF), the Contextual Transformer (CoT), and the cross-modal cycle-consistency loss (CMC). HSE results are reproduced by us. Disabling CoT means removing the cross-attention layer between local and global context.

Model	Pooling	CMC	CoT	Paragraph ⟹ Video			Video ⟹ Paragraph			Param (M)
	Lowlvl			R@1	R@5	R@50	R@1	R@5	R@50	
HSE	Max	✗	✗	45.6±0.3	76.1±0.7	96.0±0.3	44.9±0.5	75.8±1.2	95.8±0.4	26.1
HSE	Max	✓	✗	46.6±0.4	78.1±0.3	97.3±0.1	46.4±0.3	77.6±0.3	97.1±0.3	26.1
COOT	CLS	✗	✗	49.4±1.4	77.7±1.3	95.7±0.2	49.7±1.9	77.8±0.9	95.8±0.3	4.9
COOT	AVG	✗	✗	52.6±0.6	80.6±0.4	97.0±0.2	52.1±0.4	80.8±0.2	97.0±0.2	4.9
COOT	Max	✗	✗	58.2±0.5	84.9±0.2	98.1±0.1	58.7±0.5	86.0±0.2	98.2±0.1	4.9
COOT	AFA	✗	✗	59.0±0.5	85.4±0.2	98.2±0.0	59.8±0.6	85.8±0.8	98.2±0.1	5.8
COOT	Max	✓	✓	59.4±0.9	86.1±0.6	98.3±0.0	60.5±0.1	87.1±0.2	98.5±0.1	6.7
COOT	AFA	✗	✓	59.8±1.1	86.3±0.3	98.5±0.1	60.1±0.1	87.1±0.4	98.5±0.1	7.6
COOT	AFA	✓	✗	59.5±0.5	85.5±0.4	98.1±0.0	60.5±0.7	86.2±0.5	98.2±0.1	5.8
COOT	AFA	✓	✓	**60.8**±0.6	**86.6**±0.4	**98.6**±0.1	**60.9**±0.3	**87.4**±0.5	**98.6**±0.0	7.6

- **CMC 손실**은 HSE와 COOT 모두에 대한 성능을 크게 향상
- **Attention-FA** 모듈은 일반적인 avg-pooling 보다 더 나은 성능
- **Contextual Transformer**를 사용하면 성능이 향상
- HSE 방법보다 **60% 적은 10.6M의 매개변수**

04 Experiments

Result 2: paragraph to video and video to paragraph retrieval task (ActivityNet-captions)

Table 2: **Video-paragraph retrieval results on AcitvityNet-captions dataset (val1).**

	Paragraph ⟹ Video				Video ⟹ Paragraph			
Method	R@1	R@5	R@50	MR	R@1	R@5	R@50	MR
LSTM-YT [52]	0.0	4.0	24.0	102.0	0.0	7.0	38.0	98.0
No Context [53]	5.0	14.0	32.0	78.0	7.0	18.0	45.0	56.0
DENSE [39]	14.0	32.0	65.0	34.0	18.0	36.0	74.0	32.0
VSE [54]([5])	11.7	34.7	85.7	10	-	-	-	-
FSE [21]	18.2	44.8	89.1	7	16.7	43.1	88.4	7
HSE [21]	44.4±0.5	76.7±0.3	97.1±0.1	2	44.2±0.6	76.7±0.3	97.0±0.3	2
COOT	**60.8**±0.6	**86.6**±0.4	**98.6**±0.1	**1**	**60.9**±0.3	**87.4**±0.5	**98.6**±0.0	**1**

- 다양한 평가 메트릭에서 이전의 모든 방법을 훨씬 능가
- COOT는 HSE에 비해 평균 16.6% 더 나은 R@1을 얻었지만 매개변수는 더 적었음

04 Experiments

Result 3: paragraph to video and video to paragraph retrieval task (YouCook2)

Table 3: **Retrieval results on YouCook2 dataset.** Results with * are computed by us. △ we use features of a video-text model [17] pretrained on the HowTo100m dataset.

		Paragraph ⟹ Video				Sentence ⟹ Clip			
Method	TrainSet	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR
Random	-	0.21	1.09	2.19	229	0.03	0.15	0.3	1675
Miech et al. [16]	HowTo100M	43.1*	68.6*	79.1*	2*	6.1	17.3	24.8	46
ActBERT [8]	HowTo100M	-	-	-	-	9.6	26.7	38.0	19
MIL-NCE [17]	HowTo100M	61.9*	89.4*	98.9*	**1***	15.1	38.0	51.2	10
HGLMM [44]	YouCook2	-	-	-	-	4.6	14.3	21.6	75
Miech et al. [16]	YouCook2	32.3*	59.2*	70.9*	4*	4.2	13.7	21.5	65
COOT	YouCook2	50.4±2.6	79.4±0.6	87.4±0.8	1.3±0.6	5.9±0.7	16.7±0.6	24.8±0.8	49.7±2.9
Miech et al. [16]	HowTo100M+ YouCook2	59.6*	86.0*	93.6*	**1***	8.2	24.5	35.3	24
COOT	HowTo100M△+ YouCook2	**77.2**±1.0	**95.8**±0.8	**97.5**±0.3	**1.0**±0.0	**16.7**±0.4	**40.2**±0.3	**52.3**±0.5	**9.0**±0.0

다른 pretrained features 사용 여부에 따라

- Without HowTo100M: 단락-비디오 및 문장-클립 task 모두 이전보다 훨씬 좋은 성능

  ⇒서로 다른 계층 수준 간의 상호 작용을 모델링하는 것이 장기적인 의미를 포착하는 데 중요

- With HowTo100M: 이전의 SOTA를 능가하는 성능 ⇒ large-scale pretraining을 보완

04 Experiments

Result 4: Video Captioning

Table 4: **Captioning results on the YouCook2 dataset (val split).** Results with * are computed by us. $^{\triangle}$ we use features of a video-text model [17] pretrained on the HowTo100m dataset. "MART w/o re" denotes a MART variant without recurrence.

Features	Method	TrainSet	B@3	B@4	RougeL	METEOR	CIDEr-D	R@4↓
RGB+Flow	VTransformer [55]	YouCook2	13.08*	7.62	32.18*	15.65	32.26	7.83
RGB+Flow	TransformerXL [56]	YouCook2	11.46*	6.56	30.78*	14.76	26.35	6.30
RGB+Flow	MART [45]	YouCook2	12.83*	8.00	31.97*	15.90	35.74	**4.39**
COOT clip	MART	YouCook2	14.17	8.69	33.01	16.11	38.28	8.07
COOT video+clip	MART	YouCook2	15.75	9.44	34.32	18.17	46.06	6.30
COOT clip	MART	H100M$^{\triangle}$+YC2	17.12	10.91	37.59	18.85	54.07	5.11
COOT clip	MART w/o re.	H100M$^{\triangle}$+YC2	17.16	10.69	37.43	19.18	54.85	5.45
COOT clip	VTransformer	H100M$^{\triangle}$+YC2	17.62	11.09	37.63	19.34	54.67	4.57
COOT video+clip	VTransformer [55]	H100M$^{\triangle}$+YC2	17.79	11.05	37.51	19.79	55.57	5.69
COOT video+clip	MART	H100M$^{\triangle}$+YC2	**17.97**	**11.30**	**37.94**	**19.85**	**57.24**	6.69

Table 5: **Captioning results on the ActivityNet-Captions dataset (ae-test split of MART [45]).** Results with * are computed by us. "MART w/o re" denotes a MART variant without recurrence.

Features	Method	TrainSet	B@3	B@4	RougeL	METEOR	CIDEr-D	R@4↓
RGB+Flow	VTransformer [55]	ActivityNet	16.27*	9.31	29.18*	15.54	21.33	7.45
RGB+Flow	TransformerXL [56]	ActivityNet	16.71*	10.25	30.53*	14.91	21.71	8.79
RGB+Flow	MART	ActivityNet	16.43*	9.78	30.63*	15.57	22.16	5.44
COOT video+clip	TransformerXL [56]	ActivityNet	16.94	10.57	30.93	14.76	22.04	15.85
COOT video+clip	VTransformer [55]	ActivityNet	16.80	10.47	30.37	15.76	25.90	19.14
COOT clip	MART w/o re.	ActivityNet	15.41	9.37	28.66	15.61	22.05	12.03
COOT video+clip	MART w/o re.	ActivityNet	16.59	10.33	29.93	15.64	25.41	17.03
COOT clip	MART	ActivityNet	16.53	10.22	30.68	15.91	23.98	**5.35**
COOT video+clip	MART	ActivityNet	**17.43**	**10.85**	**31.45**	**15.99**	**28.19**	6.64

- 학습된 representation이 다른 task에서도 작동하는 것을 보이기 위해 MART에 적용
- COOT을 통해 학습된 representation을 사용한 MART 방법이 RGB+Flow를 사용한 방법보다 성능이 향상되었음

05 Conclusions

- 의미론적으로 잘 정렬된 **joint embedding space를 학습하기 위한** COOT(cooperative hierarchical transformer) 구조 제시
- 다양한 수준에서의 long-range temporal context 사용을 장려하도록 설계
- 2개의 새로운 component 제시
  - **attention-aware feature aggregation module**: 프레임과 단어 간의 상호 작용 모델링
  - **contextual transformer:** local contexts와 global context간의 상호작용 모델링
- 새로운 **cross-modal cycle-consistency loss** 제시: 클립과 문장의 의미론적 정렬을 강화
- **retrieval, captioning task에서 SOTA** 달성

Q&A