mPLUG

Effective and Efficient Vision-Language Learning by Cross-modal Skip connections

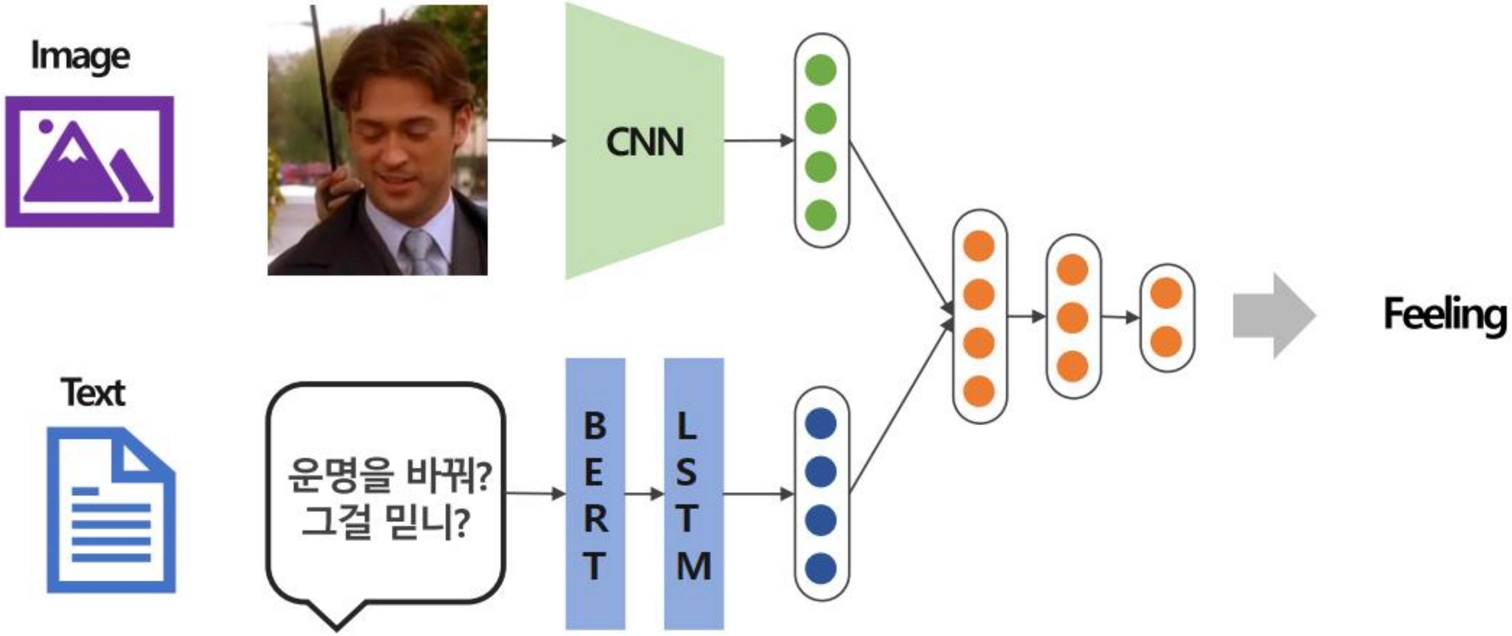
발표자: 김지환

Contents

Overview of Vision-Language models

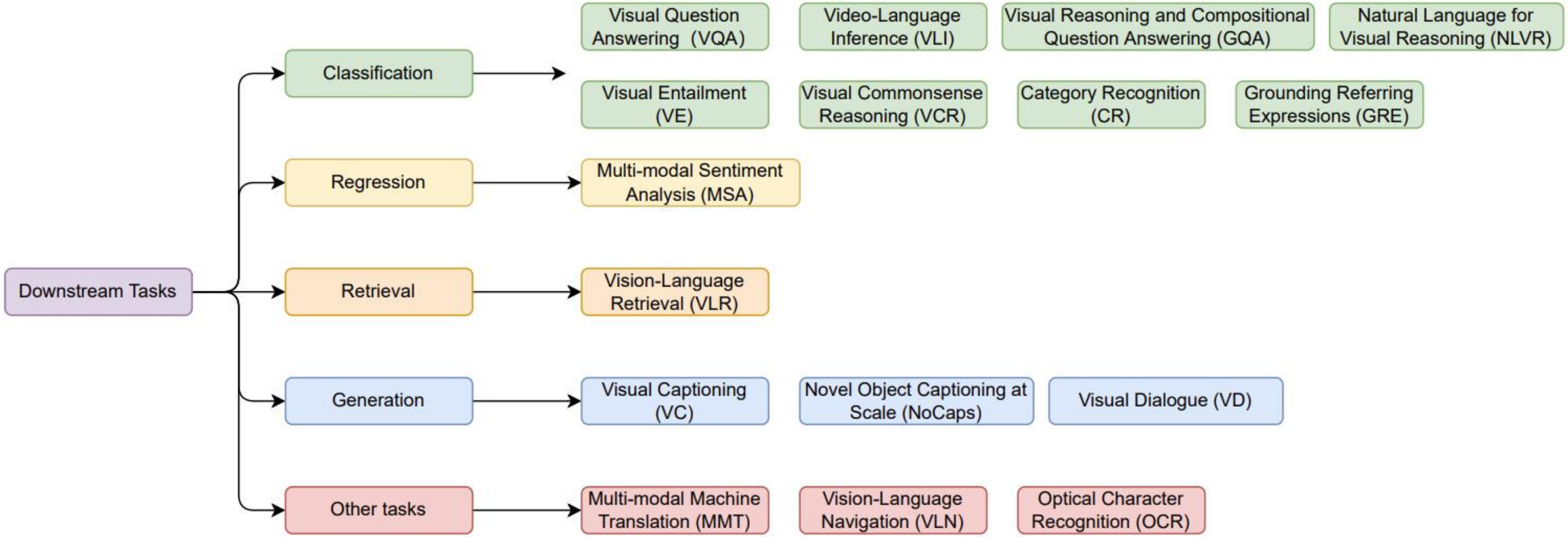
Multi-Modal Model?

두개 이상의 Modality를 활용하여 풀고자 하는 문제를 해결하는 모델



1. Overview of vision-language models

Vision-language Tasks



VLP: A Survey on Vision-language Pre-training. Fei-Long Chen

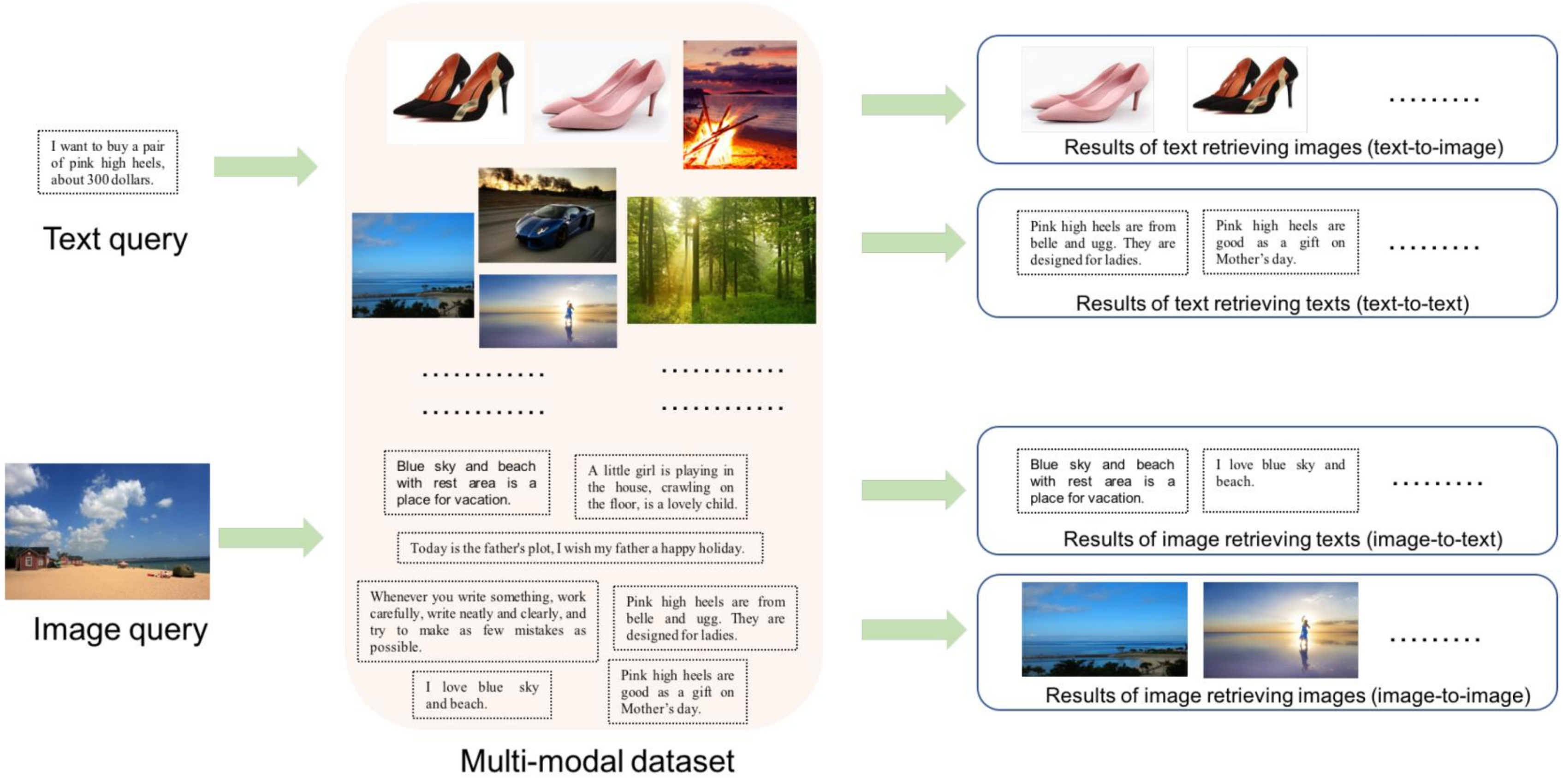
1. Overview of vision-language models

Vision-language Tasks

Elevating the Role of Image Understanding in Visual Question Answering. Yash Goyal

1. Overview of vision-language models

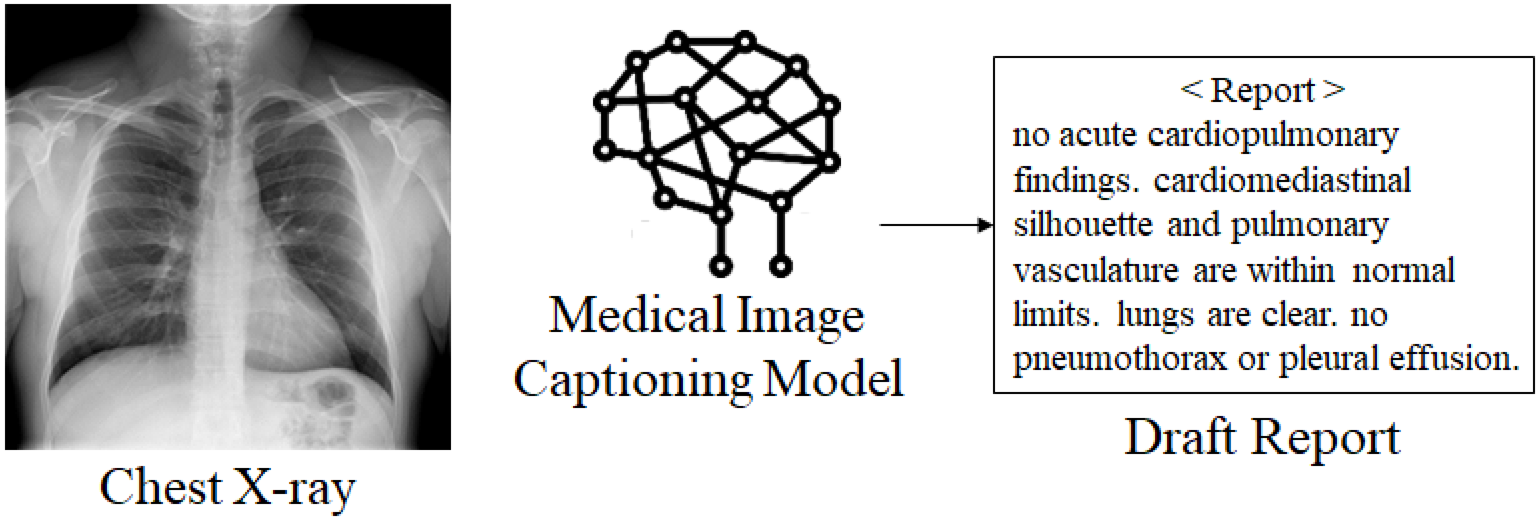
Vision-language Tasks



Deep Multi-Modal Metric Learning with Multi-Scale Correlation for Image-Text Retrieval. Yan Hua

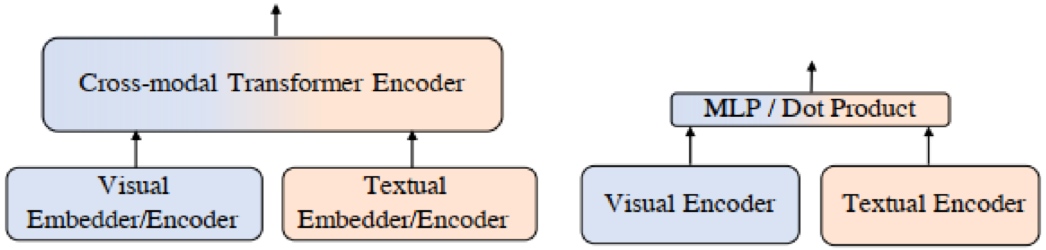
1. Overview of vision-language models

Vision-language Tasks



http://medinfolab.snu.ac.kr/research/image.html

1. Overview of vision-language models



(a) Fusion-encoder Model (b) Dual-encoder Model

(a) Fusion-encoder model
Simultaneously encode visual and textual inputs via modal-specific embedders/encoders and employ a cross-modal Transformer encoder to fuse representations.

(b) Dual-encoder model
Encode images/text separately and adopt an extreme lightweight module (*e.g.*, MLP) for cross-modal interactions.

1. Overview of vision-language models

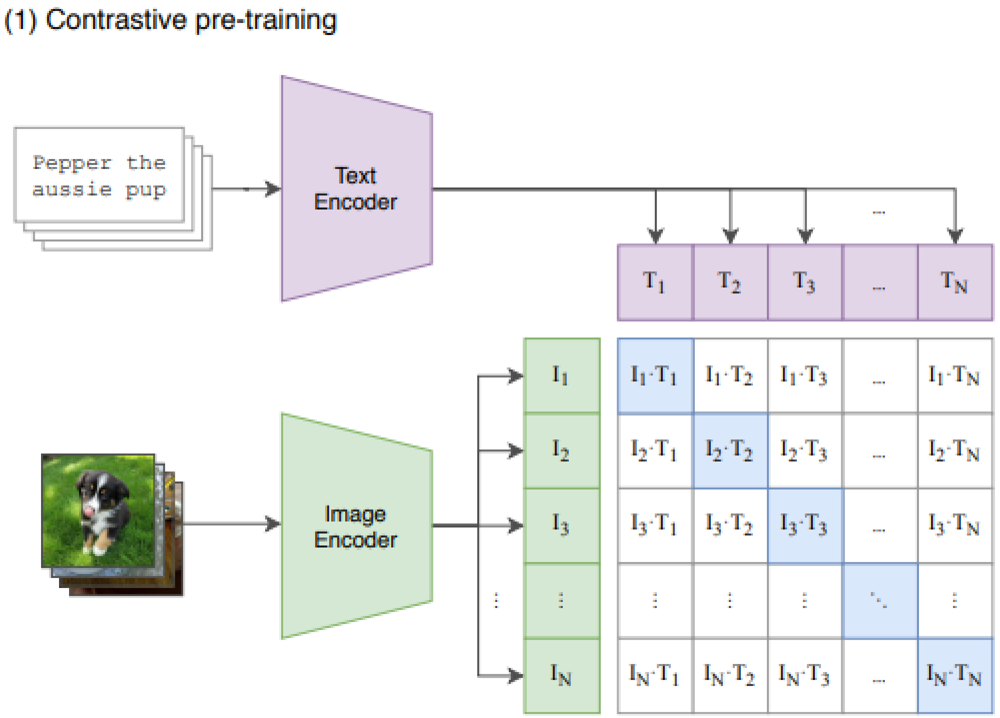
VL-PTM	Text encoder	Vision encoder	Fusion scheme	Pre-training tasks	Multimodal datasets for pre-training
Fusion Encoder					
VisualBERT [2019]	BERT	Faster R-CNN	Single stream	MLM+ITM	COCO
Uniter [2020]	BERT	Faster R-CNN	Single stream	MLM+ITM+WRA+MRFR+MRC	CC+COCO+VG+SBU
OSCAR [2020c]	BERT	Faster R-CNN	Single stream	MLM+ITM	CC+COCO+SBU+Flickr30k+VQA
InterBert [2020]	BERT	Faster R-CNN	Single stream	MLM+MRC+ITM	CC+COCO+SBU
ViLBERT [2019]	BERT	Faster R-CNN	Dual stream	MLM+MRC+ITM	CC
LXMERT [2019]	BERT	Faster R-CNN	Dual stream	MLM+ITM+MRC+MRFR+VQA	COCO+VG+VQA
VL-BERT [2019]	BERT	Faster R-CNN+ ResNet	Single stream	MLM+MRC	CC
Pixel-BERT [2020]	BERT	ResNet	Single stream	MLM+ITM	COCO+VG
Unified VLP [2020]	UniLM	Faster R-CNN	Single stream	MLM+seq2seq LM	CC
UNIMO [2020b]	BERT, RoBERTa	Faster R-CNN	Single stream	MLM+seq2seq LM+MRC+MRFR+CMCL	COCO+CC+VG+SBU
SOHO [2021]	BERT	ResNet + Visual Dictionary	Single stream	MLM+MVM+ITM	COCO+VG
VL-T5 [2021]	T5, BART	Faster R-CNN	Single stream	MLM+VQA+ITM+VG+GC	COCO+VG
XGPT [2021]	transformer	Faster R-CNN	Single stream	IC+MLM+DAE+MRFR	CC
Visual Parsing [2021]	BERT	Faster R-CNN + Swin transformer	Dual stream	MLM+ITM+MFR	COCO+VG
ALBEF [2021a]	BERT	ViT	Dual stream	MLM+ITM+CMCL	CC+COCO+VG+SBU
SimVLM [2021b]	ViT	ViT	Single stream	PrefixLM	C4+ALIGN
WenLan [2021]	RoBERTa	Faster R-CNN + EffcientNet	Dual stream	CMCL	RUC-CAS-WenLan
ViLT [2021]	ViT	Linear Projection	Single stream	MLM+ITM	CC+COCO+VG+SBU
Dual Encoder					
CLIP [2021]	GPT2	ViT, ResNet		CMCL	self-collected
ALIGN [2021]	BERT	EffcientNet		CMCL	self-collected
DeCLIP [2021b]	GPT2, BERT	ViT, ResNet, RegNetY-64GF		CMCL+MLM+CL	CC+self-collected
Fusion Encoder+ Dual Encoder					
VLMo [2021a]	BERT	ViT	Single stream	MLM+ITM+CMCL	CC+COCO+VG+SBU
FLAVA [2021]	ViT	ViT	Single stream	MMM+ITM+CMCL	CC+COCO+VG+SBU+RedCaps

A Survey of Vision-Language Pre-Trained Models, Yifan Du

1. Overview of vision-language models

CLIP *Dual Encoder

Pretraining Task - Image captioning pair



$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{\exp\left(f(x)^T f\left(x^+\right)\right)}{\exp\left(f(x)^T f\left(x^+\right)\right) + \sum_{j=1}^{N-1} \exp\left(f(x)^T f\left(x_j\right)\right)} \right]$$

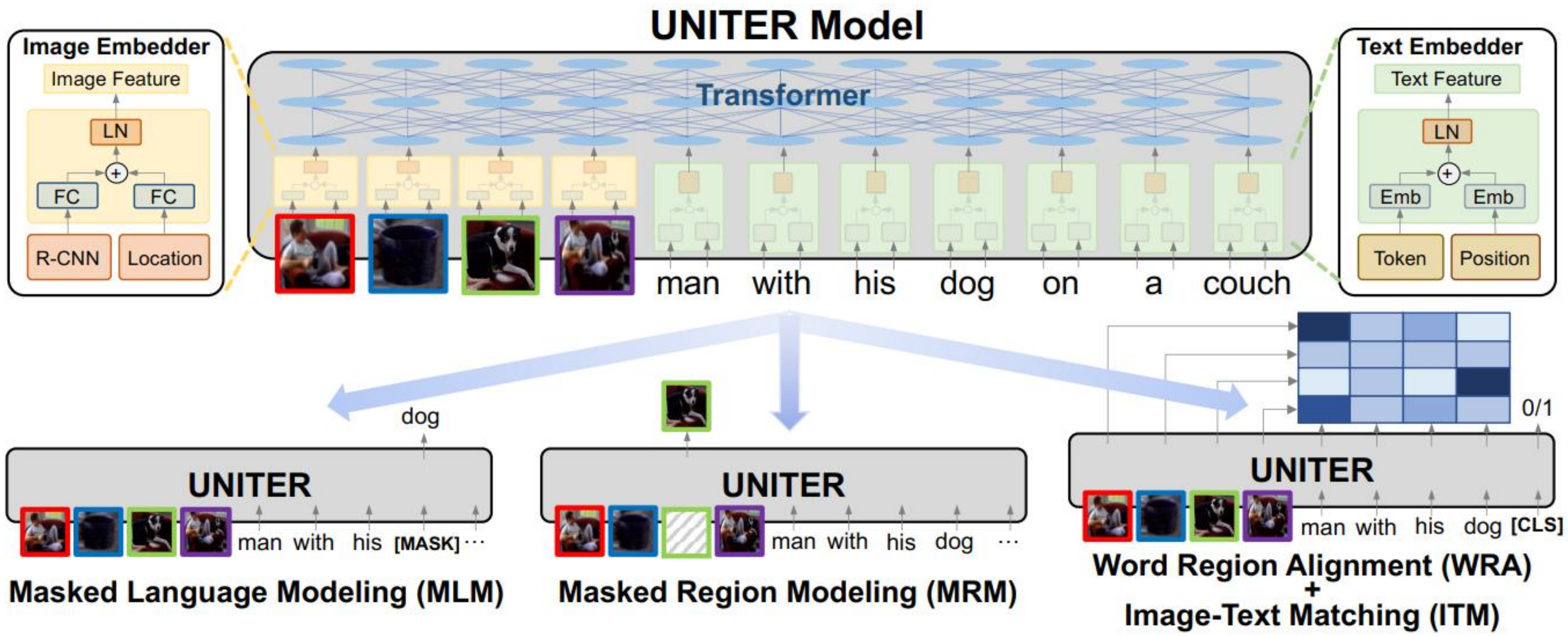
```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

$$\cos\theta = \frac{A \cdot B}{\|A\|\|B\|}$$

UNITER *Fusion Encoder

Pretraining Tasks

① Masked Language Modeling (**MLM**)
② Image-Text Matching (**ITM**)
③ Word-Region Alignment (**WRA**)
④ Masked Region Modeling (**MRM**)



$$\mathcal{L}_{\mathrm{MLM}}(\theta) = -\mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} \log P_\theta(\mathbf{w_m}|\mathbf{w}_{\backslash\mathbf{m}}, \mathbf{v})$$

$$\mathcal{L}_{\mathrm{MRM}}(\theta) = \mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D} f_\theta(\mathbf{v_m}|\mathbf{v}_{\backslash\mathbf{m}}, \mathbf{w})$$

$$\mathcal{L}_{\mathrm{WRA}}(\theta) = \mathcal{D}_{ot}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T}\in\Pi(\mathbf{a},\mathbf{b})} \sum_{i=1}^{T}\sum_{j=1}^{K} \mathbf{T}_{ij} \cdot c(\mathbf{w}_i, \mathbf{v}_j),$$

$$\mathcal{L}_{\mathrm{ITM}}(\theta) = -\mathbb{E}_{(\mathbf{w},\mathbf{v})\sim D}[y \log s_\theta(\mathbf{w},\mathbf{v}) + (1-y)\log(1-s_\theta(\mathbf{w},\mathbf{v}))])$$

UNITER: UNiversal Image-Text Representation Learning. Yen-Chun Chen

UNITER *Fusion Encoder

Pre-training Data		Pre-training Tasks	Meta-Sum	VQA	IR (Flickr)	TR (Flickr)	NLVR^2	Ref-COCO+
				test-dev	val	val	dev	val^d
None	1	None	314.34	67.03	61.74	65.55	51.02	68.73
Wikipedia + BookCorpus	2	MLM (text only)	346.24	69.39	73.92	83.27	50.86	68.80
In-domain (COCO+VG)	3	MRFR	344.66	69.02	72.10	82.91	52.16	68.47
	4	ITM	385.29	70.04	78.93	89.91	74.08	72.33
	5	MLM	386.10	71.29	77.88	89.25	74.79	72.89
	6	MLM + ITM	393.04	71.55	81.64	91.12	75.98	72.75
	7	MLM + ITM + MRC	393.97	71.46	81.39	91.45	76.18	73.49
	8	MLM + ITM + MRFR	396.24	71.73	81.76	92.31	76.21	74.23
	9	MLM + ITM + MRC-kl	397.09	71.63	82.10	92.57	76.28	74.51
	10	MLM + ITM + MRC-kl + MRFR	399.97	71.92	83.73	92.87	76.93	74.52
	11	MLM + ITM + MRC-kl + MRFR + WRA	400.93	72.47	83.72	93.03	76.91	74.80
	12	MLM + ITM + MRC-kl + MRFR (w/o cond. mask)	396.51	71.68	82.31	92.08	76.15	74.29
Out-of-domain (SBU+CC)	13	MLM + ITM + MRC-kl + MRFR + WRA	396.91	71.56	84.34	92.57	75.66	72.78
In-domain + Out-of-domain	14	MLM + ITM + MRC-kl + MRFR + WRA	**405.24**	**72.70**	**85.77**	**94.28**	**77.18**	**75.31**

mPLUG

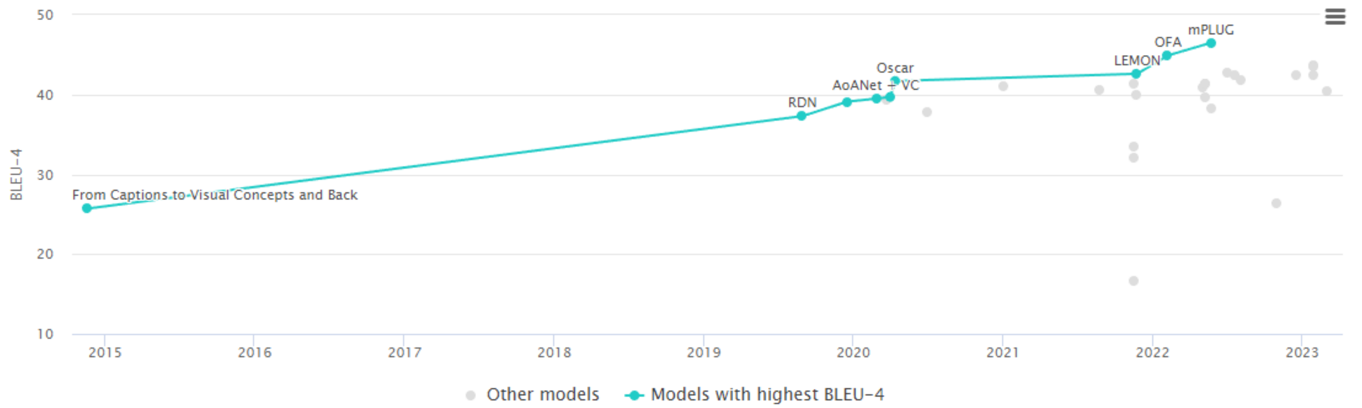
2. mPLUG

Image Captioning SOTA

Image Captioning on COCO Captions

Leaderboard Dataset

View BLEU-4 by Date



Filter: ResNet untagged

Edit Leaderboard

Rank	Model	BLEU-4	CIDER	METEOR	SPICE	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	Paper	Code	Result	Year	Tags
1	**mPLUG**	46.5	155.1	32.0	26.0					mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-			2022	

https://paperswithcode.com/sota/image-captioning-on-coco-captions

2. mPLUG

Abstract

기존의 대부분의 사전 훈련된 Vision-Language 모델은 Cross-modal 정렬에서 긴 시각적 시퀀스로 인해 낮은 계산 효율성과 정보 비대칭이라는 문제를 안고 있다.

이러한 문제를 해결하기 위해 mPLUG는 새로운 Cross-modal skip connection을 통해 효율적인 Vision-Language 아키텍처를 도입하여 vision side에서 시간소모가 심한 full self-attention에서 특정 레이어들을 건너뛰는 inter-layer shortcut을 활용한다.

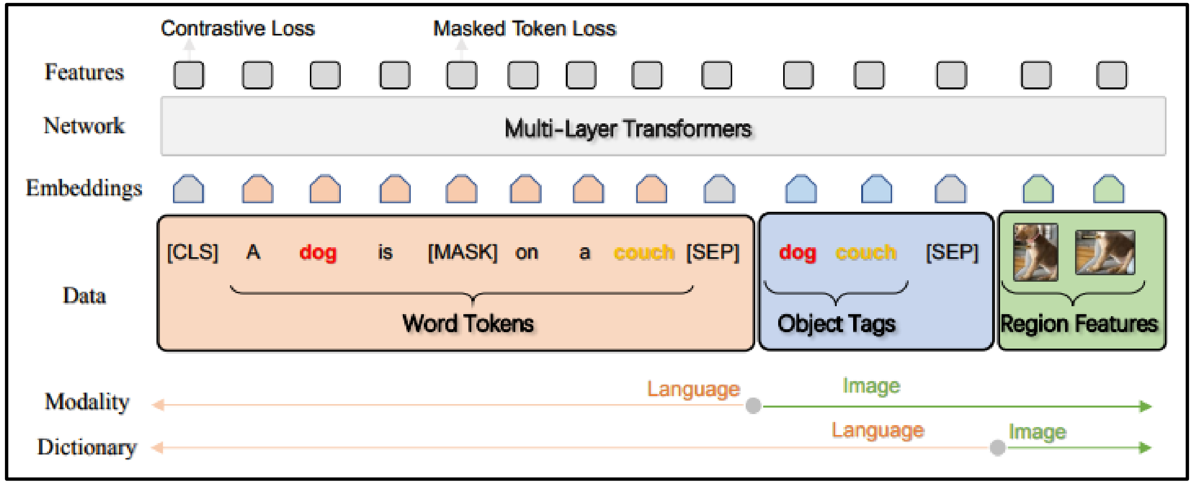
Image captioning, Image-text retrieval, Visual grounding and VQA 같은 광범위한 Vision-Language 다운스트림 작업에서 SOTA 결과를 냈다. 또한 mPLUG는 여러 Video-Language Task로 전이학습 할 때 강력한 제로 샷 성능을 보인다.

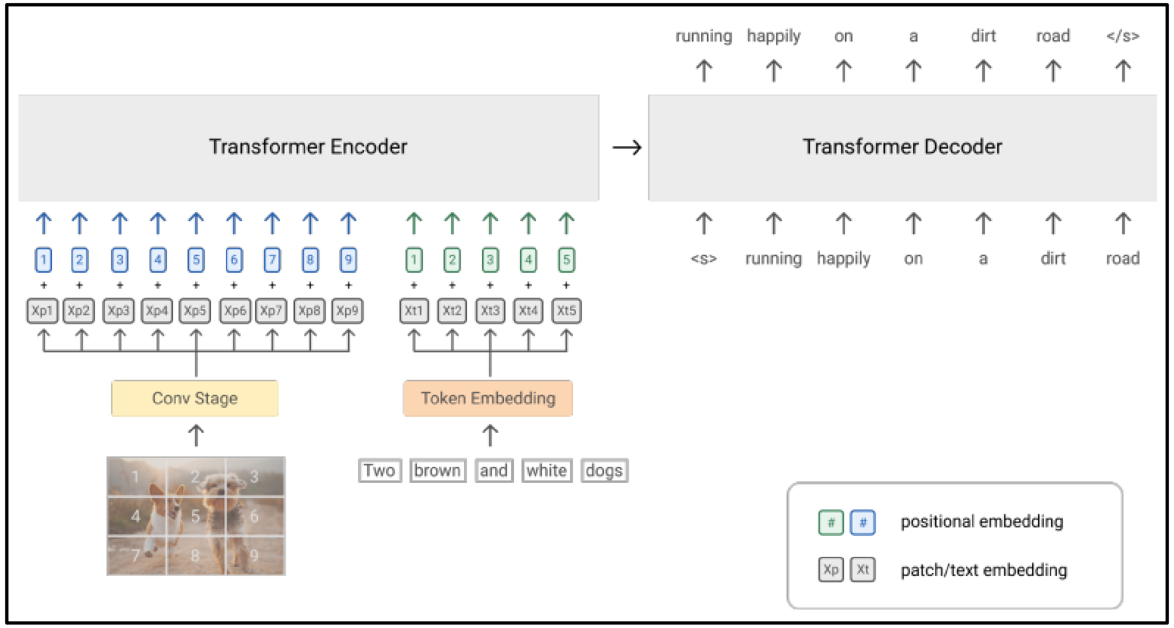
Introduction: 기존 모델

Vision-Language 모델을 학습할 때 가장 큰 어려움은 두 가지 양식을 적절히 조화시켜 그 사이의 의미적 간극을 좁히는 것이다.

이전 연구에서는 Cross-modal alignment을 위해 사전 훈련된 object detector를 사용해 이미지 구역을 추출하고 그에 맞는 언어 쌍을 나열하는 방식을 택했는데 object detector의 성능에 제한을 받을 뿐 만 아니라 사용 가능한 annotation의 양에 제한을 받는다.

보다 최근 연구에서 더 좋은 성능을 위해 사전 훈련된 object detector를 제거하고 이미지와 텍스트의 표현을 직접 정렬해 end-to-end 방식으로 학습.





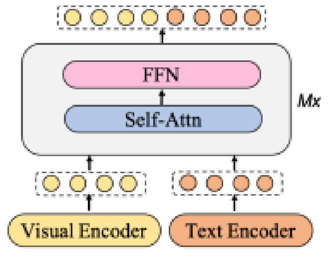
Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. Xiujun Li
SimVLM: Simple Visual Language Model Pretraining with Weak Supervision. Zirui Wang

2. mPLUG

Introduction: 기존 모델의 문제점

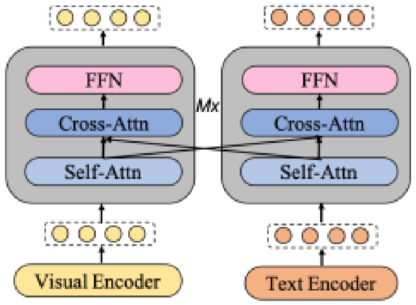
① Efficiency
긴 시각적 시퀀스에서의 Full Self-Attention은 텍스트 시퀀스보다 훨씬 더 많은 계산이 필요하다.

② Information asymmetry
널리 사용되는 Image-text 사전 학습 데이터의 캡션 텍스트는 일반적으로 짧고 매우 추상적인 반면,
이미지에서 더 상세하고 다양한 정보를 추출할 수 있다.
이러한 비대칭성으로 인해 효과적인 multi-modal fusion 에 어려움이 있다.



(a) Connected-attention Network.

Single stream fusion
이미지와 텍스트의 정보량이 달라 비대칭성이 생기고
Full Self-attention 의 비용이 큼

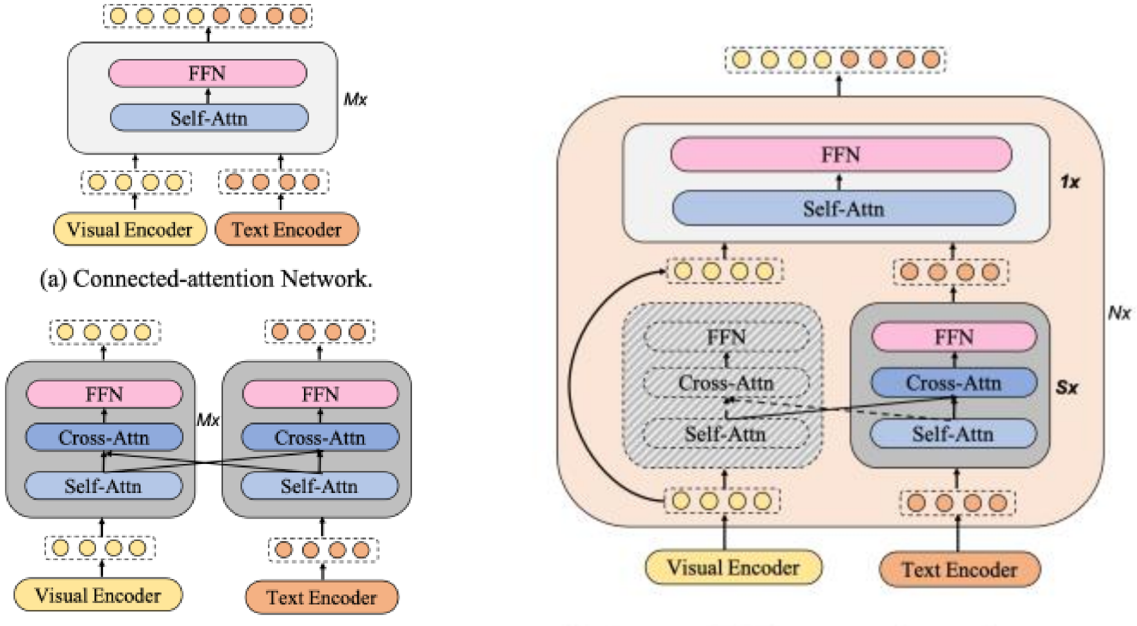


(b) Co-attention Network.

Dual stream fusion
정보 비대칭성이 줄지만 2개의 Transformer를
사용해 파라미터가 많음

Introduction: mPLUG 제안

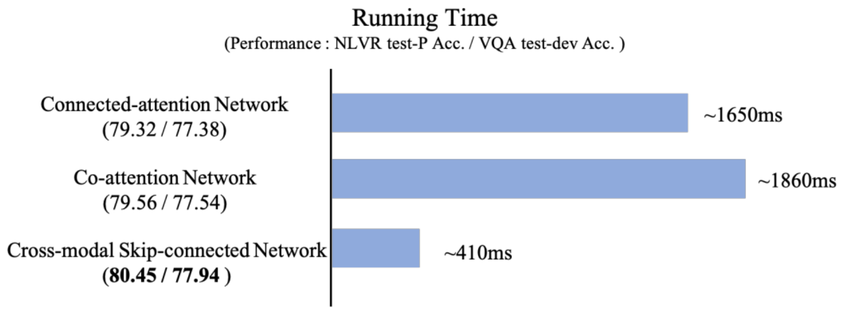
mPLUG: **M**ulti-modal **P**re-training framework for both vision-**L**anguage **U**nderstanding and **G**eneration.



(a) Connected-attention Network.

(b) Co-attention Network.

(c) Cross-modal Skip-connected Network.



4배 이상 빠른 속도로 우수한 성능을 달성!

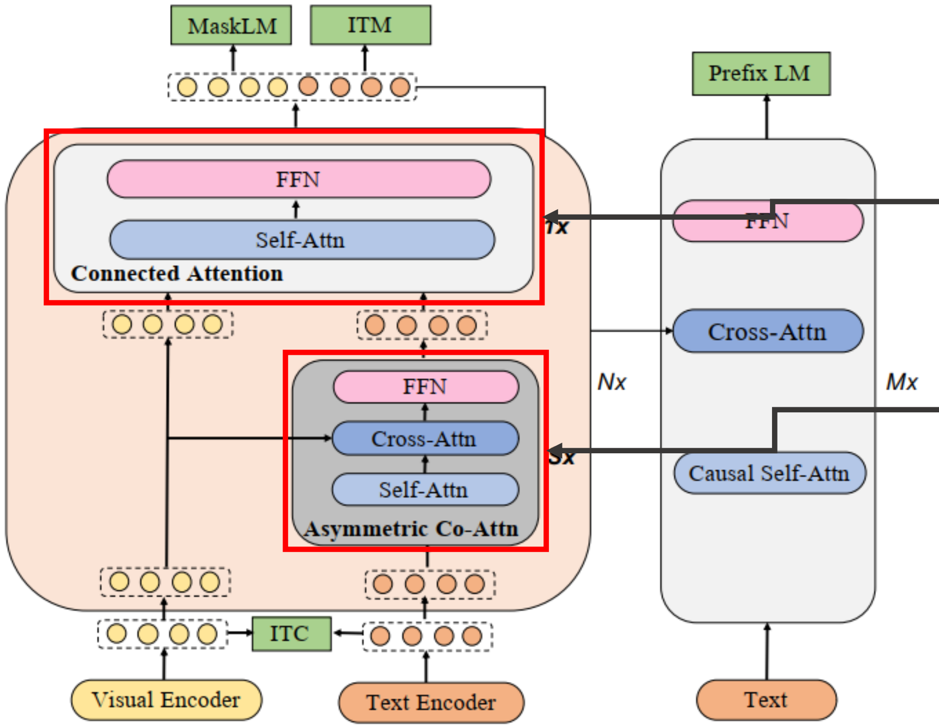
Removing the co-attention on vision-side
Concatenating the original visual representation and the co-attention output on the language side

2. mPLUG

Model Architecture

We use a visual transformer directly on the image patches as the visual encoder $\{v_{cls}, v_1, v_2, ..., v_M\}$

The input text is fed to the text encoder and represented as a sequence of embeddings $\{l_{cls}, l_1, l_2, ..., l_N\}$



Algorithm 1: Pseudocode of Cross-modal Skip-connected Network.

```
# image, text.ids, text.mask: paired {image, text} pairs.
# image_encoder: vision transformer based encoder.
# text_encoder: language transformer based encoder.
# S: the number of skipped layers in the asymmetric co-attention
# T: total layers of cross-modal skip-connections

def connected_layer(img_feature, txt_feature):
    fusion_feature = concat(img_feature, txt_feature)
    fusion_feature = norm(self_attn(fusion_feature) + fusion_feature)
    fusion_feature = norm(ffn(fusion_feature) + fusion_feature)
    img_feature, txt_feature = split(fusion_feature)
    return img_feature, txt_feature

# asymmetric co-attention architecture
def cross_layer(img_feature, txt_feature):
    txt_feature = norm(self_attn(txt_feature) + txt_feature)
    txt_feature = norm(cross_attn(txt_feature, img_feature)+
        txt_feature)
    txt_feature = norm(ffn(txt_feature) + txt_feature)
    return img_feature, txt_feature

def skip_connected_network(img_feature, txt_feature, S):
    for i in range(1, T+1):
        encoder = connected_layer if (i % (S+1) == 0)
            else cross_layer
        img_feature, txt_feature = encoder(img_feature, txt_feature)
    fusion_feature = concat(img_feature, txt_feature)
    return fusion_feature

img_feature = image_encoder(image)
txt_feature = text_encoder(text.ids, text.mask)
fusion_feature = skip_connected_network(img_feature, txt_feature, S)
```

2. mPLUG

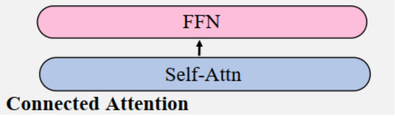
Model Architecture

Algorithm 1: Pseudocode of Cross-modal Skip-connected Network.

```
# image, text.ids, text.mask: paired {image, text} pairs.
# image_encoder: vision transformer based encoder.
# text_encoder: language transformer based encoder.
# S: the number of skipped layers in the asymmetric co-attention
# T: total layers of cross-modal skip-connections

def connected_layer(img_feature, txt_feature):
    fusion_feature = concat(img_feature, txt_feature)
    fusion_feature = norm(self_attn(fusion_feature) + fusion_feature)
    fusion_feature = norm(ffn(fusion_feature) + fusion_feature)
    img_feature, txt_feature = split(fusion_feature)
    return img_feature, txt_feature

# asymmetric co-attention architecture
def cross_layer(img_feature, txt_feature):
    txt_feature = norm(self_attn(txt_feature) + txt_feature)
    txt_feature = norm(cross_attn(txt_feature, img_feature)+
        txt_feature)
    txt_feature = norm(ffn(txt_feature) + txt_feature)
    return img_feature, txt_feature

def skip_connected_network(img_feature, txt_feature, S):
    for i in range(1, T+1):
        encoder = connected_layer if (i % (S+1) == 0)
            else cross_layer
        img_feature, txt_feature = encoder(img_feature, txt_feature)
    fusion_feature = concat(img_feature, txt_feature)
    return fusion_feature

img_feature = image_encoder(image)
txt_feature = text_encoder(text.ids, text.mask)
fusion_feature = skip_connected_network(img_feature, txt_feature, S)
```

$$[v^n_{SA}; l^n_{SA}] = LN(SA([v^{n-1}; l^{n-1}]) + [v^{n-1}; l^{n-1}]) \quad (4)$$

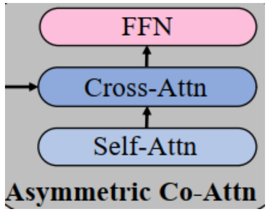
$$[v^n; l^n] = LN(FFN([v^n_{SA}; l^n_{SA}]) + [v^n_{SA}; l^n_{SA}]) \quad (5)$$



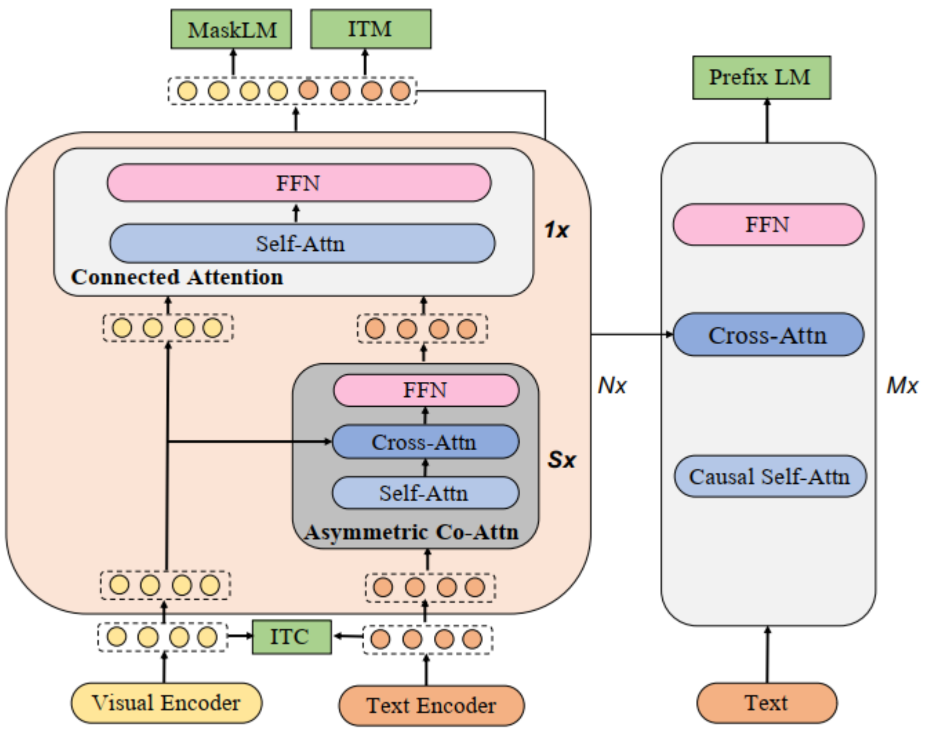
$$l^n_{SA} = LN(SA(l^{n-1}) + l^{n-1}) \quad (1)$$

$$l^n_{CA} = LN(CA(l^n_{SA}, v^{n-1}) + l^n_{SA}) \quad (2)$$

$$l^n = LN(FFN(l^n_{CA}) + l^n_{CA}) \quad (3)$$



Pretraining Tasks

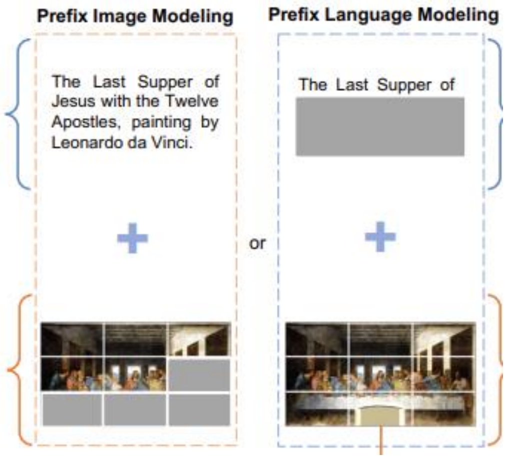


① Image-Text Contrastive (ITC): CLIP과 유사한 훈련

② Image-Text Matching (ITM): 이미지와 문장이 서로 매치되는지 훈련

③ Masked Language Modeling (MLM): BERT와 유사한 훈련

④ Prefix Language Modeling (Prefix LM): 이미지 캡션 생성



Prefix Language Models are Unified Modal Learners. Shizhe Diao

Downstream Tasks - Image Captioning, VQA

Models	Data	COCO Caption Cross-entropy Optimization B@4	M	C	S	CIDEr Optimization B@4	M	C	S	NoCaps C	S
Encoder-Decoder	CC12M	-	-	110.9	-	-	-	-	-	90.2	12.1
E2E-VLP [19]	4M	36.2	-	117.3	-	-	-	-	-	-	-
VinVL [9]	5.65M	38.5	30.4	130.8	23.4	41.0	31.1	140.9	25.2	97.3	13.8
OSCAR [4]	6.5M	-	-	-	-	41.7	30.6	140.0	24.5	83.4	11.4
SimVLM_{large} [7]	1.8B	40.3	**33.4**	**142.6**	**24.7**	-	-	-	-	-	-
LEMON_{large} [33]	200M	40.6	30.4	135.7	23.5	42.3	31.2	144.3	25.3	113.4	**15.0**
BLIP [34]	129M	40.4	-	136.7	-	-	-	-	-	113.2	14.8
OFA [35]	18M	-	-	-	-	43.5	31.9	149.6	**26.1**	-	-
mPLUG	14M	**43.1**	31.4	141.0	24.2	**46.5**	**32.0**	**155.1**	26.0	**114.8**	14.8

Table 1: Evaluation Results on COCO Caption "Karpathy" test split and NoCaps validation set. B@4: BLEU@4, M: METEOR, C: CIDEr, S: SPICE.

We first fine-tune mPLUG with cross-entropy loss and then with CIDEr optimization for extra 5 epochs.

Models	Data	Test-dev	Test-std
Pretrained on COCO, VG, SBU and CC datasets			
VLBERT [43]	4M	71.16	-
E2E-VLP [19]	4M	73.25	73.67
VL-T5 [44]	4M	-	71.30
UNITER[2]	4M	72.70	72.91
OSCAR[4]	4M	73.16	73.44
CLIP-ViL[26]	4M	76.48	76.94
METER[11]	4M	77.68	77.64
ALBEF[6]	4M	74.54	74.70
$\text{mPLUG}_{\text{ViT-B}}$	4M	**77.94**	**77.96**
Models Pretrained on More Data			
ALBEF [6]	14M	75.84	76.04
BLIP [34]	129M	78.25	78.32
SimVLM [7]	1.8B	80.03	80.34
Florence [45]	0.9B	80.16	80.36
OFA [35]	18M	79.87	80.02
VLMo [20]	-	79.94	79.98
$\text{mPLUG}_{\text{ViT-B}}$	14M	79.79	79.81
$\text{mPLUG}_{\text{ViT-L}}$	14M	**81.27**	**81.26**

SimVLM, Florence 대비 적은 수의
사전 훈련 데이터로 높은 성능을 당성

2. mPLUG

Downstream Tasks - ITR, VG

Models	# Pretrain data	MSCOCO (5K test set) TR R@1	R@5	R@10	IR R@1	R@5	R@10	Flickr30K (1K test set) TR R@1	R@5	R@10	IR R@1	R@5	R@10
E2E-VLP [19]	4M	-	-	-	-	-	-	86.2	97.5	98.92	73.6	92.4	96.0
UNITER [2]	4M	65.7	88.6	93.8	52.9	79.9	88.0	87.3	98.0	99.2	75.6	94.1	96.8
OSCAR [4]	4M	70.0	91.1	95.5	54.0	80.8	88.5	-	-	-	-	-	-
UNIMO [46]	4M	-	-	-	-	-	-	89.4	98.9	99.8	78.0	94.2	97.1
VLMo [20]	4M	78.2	94.4	97.4	60.6	84.4	91.0	95.3	99.9	100.0	84.5	97.3	98.6
ALIGN [18]	1.8B	77.0	93.5	96.9	59.9	83.3	89.8	95.3	99.8	100.0	84.9	97.4	98.6
ALBEF [6]	14M	77.6	94.3	97.2	60.7	84.3	90.5	95.9	99.8	100.0	85.6	97.5	98.9
Florence [45]	0.9B	81.8	95.2	-	63.2	85.7	-	97.2	99.9	-	87.9	98.1	-
BLIP [34]	14M	80.6	95.2	97.6	63.1	85.3	91.1	96.6	99.8	100.0	87.2	97.5	98.8
BLIP [34]	129M	82.4	95.4	97.9	65.1	86.3	91.8	97.4	99.8	99.9	87.6	97.7	99.0
mPLUG	14M	**82.8**	**96.1**	**98.3**	**65.8**	**87.3**	**92.6**	**97.6**	**100.0**	**100.0**	**88.4**	**97.9**	**99.1**

Table 3: Image-text retrieval results on Flickr30K and COCO datasets.

Model	RefCOCO val	testA	testB	RefCOCO+ val	testA	testB	RefCOCOg val-u	test-u
VLBERT [43]	-	-	-	72.59	78/57	62.30	-	-
UNITER [2]	81.41	87.04	74.17	75.90	81.45	66.70	74.86	75.77
VILLA [50]	82.39	87.48	74.84	76.17	81.54	66.84	76.18	76.71
MDETR [51]	86.75	89.58	81.41	79.52	84.09	70.62	81.64	80.89
UNICORN [52]	88.29	90.42	83.06	80.30	85.05	71.88	83.44	83.93
OFA [35]	90.05	92.93	85.26	84.49	90.10	77.77	84.54	85.20
mPLUG	**92.40**	**94.51**	**88.42**	**86.02**	**90.17**	**78.17**	**85.88**	**86.42**

Table 4: Visual grounding results (Acc@0.5) on ReferCOCO, ReferCOCO+, and ReferCOCOg.

Zero-shot Transferability

Model	TR R@1	TR R@5	IR R@1	IR R@5
Zero-Shot				
CLIP [17]	88.0	98.7	68.7	90.6
ALIGN [18]	88.6	98.7	75.7	93.8
FLIP [56]	89.8	99.2	75.0	93.4
Florence [45]	90.9	99.1	76.7	93.6
ALBEF† [6]	94.1	99.5	82.8	96.3
BLIP† [34]	94.8	99.7	84.9	96.7
mPLUG	**93.0**	**99.5**	**82.2**	**95.8**
mPLUG†	**95.8**	**99.8**	**86.4**	**97.6**

Table 8: Zero-shot image-text retrieval results on Flickr30K. † denotes the models finetuned on COCO.

Model	# Pretrain data	MSRVTT-Retrieval R@1	R@5	R@10
Zero-Shot				
MIL-NCE [57]	How100M	9.9	24.0	32.4
VideoCLIP [58]	How100M	10.4	22.2	30.0
VATT [59]	How100M, AudSet	-	-	29.7
ALPRO [60]	W2M, C3M	24.1	44.7	55.4
VIOLET [61]	Y180M, W2M, C3M	25.9	49.5	59.7
CLIP [17]	WIT400M	26.0	49.4	60.7
Florence [45]	FLD900M	37.6	63.8	72.6
BLIP † [34]	129M	43.3	65.6	74.7
mPLUG	14M	38.1	59.2	68.2
mPLUG †	14M	**44.3**	**66.4**	**75.4**
Fine-Tuning				
VideoCLIP [58]	How100M	30.9	55.4	66.8
ALPRO [60]	C3M, W2M	33.9	60.7	73.2
VIOLET [61]	Y180M, C3M, W2M	34.5	63.0	73.4

Table 9: Zero-shot video-language results on text-to-video retrieval on the 1k test split of the MSRVTT dataset. † denotes the models finetuned on COCO. Video datasets include HowTo100M [62], WebVid-2M(W2M) [63], YT-Temporal-180M(Y180M) [64]. Image datasets include CC3M(C3M) [38], FLD900M [45], WIT400M [17]. Audio datasets include AudioSet(AudSet) [65].

Model	MSRVTT-QA Acc	MSVD-QA Acc	VATEX-Cap CIDEr
Zero-Shot			
VQA-T [66]	2.9	7.5	-
BLIP [34]	19.2	35.2	37.4
mPLUG	**21.1**	**37.2**	**42.0**

Table 10: Zero-shot video-language results on Question-Answer and Caption tasks.

Q&A

Thank you