14주차 논문 리뷰 - Flamingo: a Visual Language Model for Few-Shot Learning

0. Abstract

- **목표**: 새로운 작업에 빠르게 적응하는 **Multimodal 모델**을 구축하는 것
- Flamingo라는 Visual Language Model (VLM)을 소개
  1. **Vision-only** 및 **Language-only 모델**을 연결
  2. 임의로 섞인 시각적 및 텍스트 데이터를 효율적으로 처리
  3. 이미지와 텍스트를 매끄럽게 통합해 입력으로 사용
- **유연성**: Flamingo는 **Multimodal 웹 데이터**를 활용해 학습되며, **in-context few-shot learning**(소수의 예시를 기반으로 빠르게 학습)이 가능함
- **결과**
  - 다양한 이미지/비디오 작업에서 평가됨
  - **Open-ended 질문**, **Captioning**, **Multiple-choice 질문** 등에서 높은 성능을 보여줌
  - **Few-shot learning** 기반으로 소량의 작업별 예시만으로도 성능을 발휘함

1. Introduction

- **배경**
  - **Intelligence**(지능)핵심: 짧은 지시를 통해 새로운 작업을 빠르게 학습하는 것
  - 기존의 컴퓨터 비전 모델은 대규모 Supervised learning 데이터로 학습된 후 특정 작업에 맞게 **Fine-tuning**이 필요했음

    ⇒ Fine-tuning은 많은 **데이터**와 **리소스**가 필요하기 때문에 비효율적
- **해결 방향**

    - 최근 연구된 **Multimodal Vision-Language Model**은 적은 데이터로 다양한 작업을 수행할 수 있도록 설계됨
    - 하지만 기존 모델은 시각과 텍스트 간의 단순한 유사성 점수만 제공해 **Open-ended**(개방형) 작업에는 적합하지 않음
- **Flamingo의 특징**
    - Flamingo는 Visual Language Model (VLM)으로, 텍스트와 시각적 데이터를 결합하여 **Few-shot learning**을 수행
    - **강점**
        1. 임의의 **멀티모달 입력**(이미지/텍스트)을 처리함
        2. Vision-Text 데이터를 활용해 **Autoregressive**(자동회귀 방식)로 텍스트를 생성
    - **구조**: Flamingo는 두 개의 모델을 결합해 작동
        - **Pre-trained**된 Vision 모델
        - **Frozen Language 모델**
- **결과**: Flamingo는 소수의 예시만으로 **16가지 이미지/비디오 이해 작업**을 수행하며, 기존 Fine-tuned 모델을 능가하는 성능을 보여줌
    - 특히 **Few-shot learning**의 성능이 뛰어나고, 적은 양의 작업별 학습 데이터만 사용했음에도 높은 효율성을 달성

2. Approach

구성 요소

1. **Vision Encoder → Perceiver Resampler**
    - **Vision Encoder**: NFNet (F6)을 기반으로 하며, 이미지와 비디오 프레임을 spatio-temporal features로 변환
        - 이미지 → 2D feature map
        - 비디오 → 1FPS로 샘플링 후 3D feature map에 시간 임베딩 추가
    - **Perceiver Resampler**
        - **다양한 크기의 시각 특징**을 받아 고정된 수(64)의 시각 토큰으로 변환
        - **Transformer 구조**를 사용하여 시각 특징을 요약

    - **목적**: vision-text 교차 계산량을 줄임
2. **GATED XATTN-DENSE Layers**
    - 기존 언어 모델 (LM)과 **새로운 cross-attention 레이어**를 교차 배치
    - **Frozen Language Model**에 시각 정보를 추가로 제공하여 텍스트 예측 시 시각적 특징을 활용
    - **Gating 메커니즘**: 모델 초기화 시 LM 성능이 그대로 유지되도록 안정성을 보장함
3. **Interleaved Multi-Visual Input**
    - 이미지/비디오와 텍스트가 섞인 입력에서 개별 시각적 토큰만 조건으로 사용
    - **단일 이미지 주의 메커니즘**: 특정 텍스트 토큰이 직전에 입력된 이미지/비디오에만 주의를 집중
        - **장점**: 시각적 입력 개수와 관계없이 일반화가 가능하며 **최대 32쌍**까지 확장 가능
4. **데이터셋 구성**
    - **M3W**: 웹페이지에서 수집된 **이미지와 텍스트의 혼합 데이터**
    - **LTIP**: 고품질 이미지-텍스트 쌍 데이터셋 (312M 쌍)
    - **VTP**: 비디오와 텍스트 쌍 데이터셋 (27M 쌍, 22초 평균)
    - **Multi-Objective Training**:
        - 여러 데이터셋에서 **음의 로그 가능도**를 최소화하는 방식으로 학습
        - 각 데이터셋 가중치(λ)를 조정하여 성능 최적화

Few-shot In-context Learning

- **Flamingo**는 학습된 후 새로운 시각적 작업에 **Few-shot learning**을 통해 빠르게 적응
- **입력 형식**:
    - 이미지/비디오와 텍스트 쌍 → **Prompt**로 구성 (예: 이미지 5쌍, 질문)
- **평가 방식**:
    - **Open-ended Task**: Beam search로 텍스트 생성
    - **Close-ended Task**: **Log-likelihood** 기반으로 정답 확률 평가

결과

- **Few-shot Learning 성능**:
  - Flamingo는 4~32개의 예시만으로도 기존 **Fine-tuning 모델**보다 우수한 성능을 보여줌
  - 특히, **32-shot**에서는 수천 개의 주석 데이터로 학습된 모델을 능가하는 결과를 달성함

3. Experiments

목적

- 다양한 멀티모달 이미지/비디오 및 언어 벤치마크를 사용하여 모델 성능을 평가하고 검증하는 것을 목표로 함
- 총 16개의 벤치마크 중 **5개**는 모델 설계 및 하이퍼파라미터 조정을 위해 **DEV set**으로 사용됨:
  - **COCO, OKVQA, VQAv2, MSVDQA, VATEX**
- 나머지 **11개** 벤치마크는 **few-shot** 학습 성능을 검증하는 데만 사용:
  - 다양한 작업을 포함 (이미지 캡셔닝, 비디오 질의응답, 멀티-초이스 질의응답 등)
- **평가 기준**:
  - 모든 벤치마크에 대해 고정된 하이퍼파라미터를 사용
  - 네 가지 **few-shot prompt templates**를 사용

- DEV set 벤치마크는 모델 설계 조정에 사용되므로 결과가 편향될 수 있으나, 나머지 11개는 편향 없는 성능 평가를 위해 활용

3-1. Few-shot learning on vision-language tasks

Few-shot 결과

Method	FT	Shot	OKVQA (I)	VQAv2 (I)	COCO (I)	MSVDQA (V)	VATEX (V)	VizWiz (I)	Flick30K (I)	MSRVTTQA (V)	iVQA (V)	YouCook2 (V)	STAR (V)	VisDial (I)	TextVQA (I)	NextQA (I)	HatefulMemes (I)	RareAct (V)
Zero/Few shot SOTA	✗	(X)	[34] 43.3 (16)	[114] 38.2 (4)	[124] 32.2 (0)	[58] 35.2 (0)	-	-	-	[58] 19.2 (0)	[135] 12.2 (0)	-	[143] 39.4 (0)	[79] 11.6 (0)	-	-	[85] 66.1 (0)	[85] 40.7 (0)
Flamingo-3B	✗	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	✗	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	✗	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	47.3	30.6	26.1	56.3	-
Flamingo-9B	✗	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	✗	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	**42.8**	50.4	33.6	24.7	62.7	-
	✗	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	50.4	32.6	28.4	63.5	-
Flamingo	✗	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	**60.8**
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	**55.6**	36.5	30.8	68.6	-
	✗	32	**57.8**	**67.6**	**113.8**	**52.3**	**65.1**	**49.8**	**75.4**	**31.0**	**45.3**	**86.8**	42.2	**55.6**	**37.9**	**33.5**	**70.0**	-
Pretrained FT SOTA	✓	(X)	54.4 [34] (10K)	80.2 [140] (4448K)	143.3 [124] (500K)	47.9 [28] (27K)	76.3 [153] (500K)	57.2 [65] (20K)	67.4 [150] (30K)	46.8 [51] (130K)	35.4 [135] (6K)	138.7 [132] (10K)	36.7 [128] (46K)	75.2 [79] (123K)	54.7 [137] (20K)	25.2 [129] (38K)	79.1 [62] (9K)	-

Table 1: **Comparison to the state of the art.** A *single* Flamingo model reaches the state of the art on a wide array of image **(I)** and video **(V)** understanding tasks with few-shot learning, significantly outperforming previous best zero- and few-shot methods with as few as four examples. More importantly, using only 32 examples and without adapting any model weights, Flamingo *outperforms* the current best methods – fine-tuned on thousands of annotated examples – on seven tasks. Best few-shot numbers are in **bold**, best numbers overall are underlined.

- **Flamingo** 모델은 기존 SOTA 성능을 대부분 뛰어넘으며 새로운 few-shot 기준을 수립
  - **Few-shot**: 모델이 아주 적은 양의 학습 샘플만 사용해 테스트하는 방식
- 모델 크기가 클수록 성능이 향상되며, **더 많은 shots**를 사용할수록 성능이 추가적으로 개선됨

Few-shot 성능의 주요 포인트

- Flamingo는 **대형 언어 모델**(GPT-3와 유사)의 few-shot 학습 방식을 따르며, 더 많은 shots에서 특히 강력한 성능을 보여줌
- 특히 **M3W** 데이터셋의 경우, 5개의 이미지로 학습된 모델도 32개의 이미지나 비디오에서 좋은 성능을 보임:
  - 이는 **Flamingo 아키텍처**의 유연성을 입증
  -

3.2 Fine-tuning Flamingo as a pretrained vision-language model

Fine-tuning 성능

- Flamingo 모델은 fine-tuning을 통해 few-shot 결과를 더욱 개선할 수 있음
- **Table 2**에서 보여주듯 Flamingo는 **9개 벤치마크**에서 fine-tuning을 통해 새로운 SOTA를 달성:
  - 기존 SOTA 모델들과의 비교에서 **5개 벤치마크**를 초과 달성함

- 일부 벤치마크는 **CIDEr**와 같은 특화된 최적화 기준을 사용하는 모델보다 우수

3.3 Ablation studies

	Ablated setting	*Flamingo*-3B original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
		***Flamingo*-3B model**		3.2B	1.74s	86.5	42.1	55.8	36.3	53.4	**70.7**
(i)	Training data	All data	w/o Video-Text pairs	3.2B	1.42s	84.2	43.0	53.9	34.5	46.0	67.3
			w/o Image-Text pairs	3.2B	0.95s	66.3	39.2	51.6	32.0	41.6	60.9
			Image-Text pairs→ LAION	3.2B	1.74s	79.5	41.4	53.5	33.9	47.6	66.4
			w/o M3W	3.2B	1.02s	54.1	36.5	52.7	31.4	23.5	53.4
(ii)	Optimisation	Accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	33.2	40.8	62.9
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	47.5	66.5
(iv)	Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	50.7	66.9
			GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	47.8	63.1
(v)	Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	29.1	42.3	59.8
			Every 4th	2.3B	1.02s	82.3	42.7	55.1	34.6	50.8	68.8
			Every 2nd	2.6B	1.24s	83.7	41.0	55.8	34.5	49.7	68.2
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	44.7	66.6
			Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	48.3	66.7
(vii)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	44.5	64.9
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	42.9	62.7
(viii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	50.1	57.8
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	53.9	62.7

Table 3: **Ablation studies.** Each row should be compared to the baseline Flamingo run (top row). Step time measures the time spent to perform gradient updates on all training datasets.

Flamingo-3B 모델을 사용한 **ablation study** 결과를 요약한 것

1. **Training data mixture의 중요성 (i)**
   - **M3W** 데이터셋 제거 시 성능이 **17% 이상 감소**
   - **Image-Text pair**(ITP) 제거는 성능 저하를 유발함
   - 다양한 데이터셋의 혼합이 중요함을 입증
2. **Gradient Accumulation 방식 (ii)**
   - **Round-Robin** 대신 효율적인 gradient accumulation 전략을 사용해야 성능이 유지됨
3. **Frozen LM에 대한 시각적 조건부 설정 (iii)**
   - **0-initialized tanh gating**을 비활성화할 경우 성능이 **4.2% 감소**
   - **XATTN-DENSE** 구조가 **가장 우수**한 성능을 보여줌
4. **Compute/Memory vs. Performance trade-off (iv)**
   - **Cross-Attention Layer**를 **매 4번째 층**에 추가할 경우 메모리와 연산량이 절약되지만, 성능 저하를 최소화함

- **Perceiver Resampler**와 **Transformer** 비교 시, Resampler는 느리지만 성능이 비슷함

5. **Vision Encoder의 역할 (vi)**
   - **NFNet-F6** 인코더와 CLIP 기반 모델 비교:
     - **NFNet-F6**는 CLIP ViT-L/14보다 성능이 **5.8%~8% 향상**
   - 이는 Flamingo의 비디오/이미지 입력 처리 성능을 높이는 데 기여
6. **Freezing LM components prevents catastrophic forgetting (viii)**
   - **Catastrophic Forgetting** 방지를 위해 언어 모델(LM) 구성 요소를 동결하는 것이 필요함을 검증함
   - **Scratch**(처음부터 학습)로 훈련하면 **성능이 –12.9% 감소**
   - 사전 학습된 LM을 **fine-tuning**하면 성능이 **–8.0% 감소**
     - 이는 **catastrophic forgetting**의 사례로, 새로운 목표로 훈련하면서 모델이 사전 학습 내용을 점진적으로 잊어버리기 때문
   - 따라서 **언어 모델을 동결**하는 것이 사전 학습 데이터셋(MassiveText)을 포함하여 학습하는 것보다 더 나은 대안임

4. Related Work

- **언어 모델링과 Few-Shot 적응**
  - Transformer 도입 이후 언어 모델링은 크게 발전했으며, **사전 학습(pretraining)** 후 다운스트림 작업에 적응하는 접근이 표준화됨
  - Flamingo는 **Chinchilla (70B)** 언어 모델을 기반으로 구축됨
  - Few-shot 학습에 대한 기존 연구들은 **Adapter 추가, 부분 fine-tuning**, 또는 **In-context learning**(프롬프트 내 예제 제시)을 주로 활용해 왔음
  - Flamingo는 **In-context few-shot 학습**을 따르며, **metric learning**이나 **meta-learning** 기반의 복잡한 few-shot 학습 기법과 차별화됨
- **언어 모델과 비전(시각)의 융합**
  - BERT와 같은 LM 혁신이 **Vision-Language 모델(VLM)** 연구에 큰 영향을 미침
  - 기존 VLM 연구들은 **fine-tuning**을 필요로 하나, Flamingo는 **fine-tuning 없이** 다양한 작업에 적응할 수 있음
  - Flamingo는 **Contrastive Learning** 기반 모델과 달리 텍스트 생성이 가능함

    - 비슷한 연구로, autoregressive 텍스트 생성 모델이 존재하며, Flamingo는 이를 확장해 **임의의 이미지, 비디오, 텍스트를 결합**할 수 있는 첫 LM을 도입
- **웹 스케일 비전-언어 학습 데이터셋**
    - 수작업으로 레이블링된 데이터셋은 비용이 높고 규모가 작음(10k-100k)
    - 따라서 자동으로 수집한 **대규모 vision-text 데이터**가 중요하며, Flamingo는 **웹 페이지 전체**를 단일 시퀀스로 학습 데이터로 사용
    - **CM3**는 HTML 생성 방식을 사용하지만, Flamingo는 더 간단히 **plain text**를 예측함

5. Discussion

- **한계점**
    1. **언어 모델의 한계 상속**:Flamingo는 사전 학습된 LM의 약점을 그대로 가져옴
        - 예시: **할루시네이션(hallucinations)**, **긴 시퀀스 처리** 어려움, 낮은 샘플 효율성
    2. **분류 성능 한계**:Flamingo는 **Contrastive Learning 기반 모델**에 비해 분류 성능이 낮으나, **더 광범위한 작업**에 적용 가능
    3. **In-context learning의 한계**:
        - 프롬프트 예시 특성에 따라 성능이 민감하게 변함
        - 많은 **shots**를 사용할 때 **추론 비용**과 성능이 비효율적임
- **사회적 영향**
    - Flamingo의 범용성은 긍정적 및 부정적 영향을 동시에 가질 수 있음
    - 주요 위험:
        - **언어 모델**과 마찬가지로 편향된 결과, 공격적 언어, 프라이버시 침해 가능성 존재
        - **시각 입력**의 추가로 **인종적, 성별 편향**의 위험이 더 커질 수 있음