Big Models, Few-Shot Learning, and Model Evaluation

Zhe Gan
Principal Researcher
6/19/2022
What has been covered so far

• The general landscape of image-text pre-training
  • OD-based methods: LXMERT, ViLBERT, UNITER, VL-BERT, OSCAR, VILLA, VinVL etc.
  • The current prevailing E2E methods: PixelBERT, SOHO, ViLT, ALBEF, CLIP-ViL, METER, BLIP etc.

• Unified image-text modeling
  • Output format: VL-T5, GPV, MDETR, UniT, UNICORN, OFA etc.
  • Architecture: UFO, VLMo, VL-BEiT etc.

• A typical academic setting for all these models
  • Model size: base (~110M) or large (~340M)
  • Pre-training data: 4M images in total (COCO+VG+SBU+CC3M)
So, what do we offer in this talk?

• **Part I: Big multimodal foundation models**
  • Beyond base/large sizes, and beyond 4M images
  • Examples include SimVLM, CoCa, and GIT

• **Part II: Multimodal few-shot learning**
  • How can we enable in-context few-shot learning?
  • Examples include Frozen, PICa, and Flamingo

• **Part III: Model evaluation**
  • What’s next for VL model evaluation?
  • Diagnostic tests, challenge sets, probing analysis
Part I: Big Models
What are big foundation models?

• Let’s take a look at big language foundation models

- BERT (base 110M, large 340M)
- GPT-3 (175B)
- PaLM (540B)
How about big multimodal models?

- Models that have either billion-level parameters or use billion-level pre-training data are considered as “big” in this context.
- First, note that foundation models are not necessarily needed to be big.
- CLIP-like dual encoders and text-to-image big models (DALLE-2, Imagen) are not considered here (will be covered in the afternoon session).
- Take VQA as an example
  - OD-based models
  - E2E models

Large model sizes and pre-training data have been the driving force for SOTA performance.

We will also briefly talk about what’s beyond SOTA chasing in later slides.
# A summary of big multimodal models

<table>
<thead>
<tr>
<th>Model</th>
<th>Model Size</th>
<th>#Pre-training image-text data</th>
<th>Pre-training tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Img Enc</td>
<td>Txt Enc</td>
<td>Fusion</td>
</tr>
<tr>
<td>CLIP ViT-L/14</td>
<td>302M</td>
<td>123M</td>
<td>0</td>
</tr>
<tr>
<td>ALIGN</td>
<td>480M</td>
<td>340M</td>
<td>0</td>
</tr>
<tr>
<td>Florence</td>
<td>637M</td>
<td>256M</td>
<td>0</td>
</tr>
<tr>
<td>SimVLM-huge</td>
<td>300M</td>
<td>39M</td>
<td>600M</td>
</tr>
<tr>
<td>METER-huge</td>
<td>637M</td>
<td>125M</td>
<td>220M</td>
</tr>
<tr>
<td>LEMON</td>
<td>147M</td>
<td>39M</td>
<td>636M</td>
</tr>
<tr>
<td>Flamingo</td>
<td>200M</td>
<td>70B</td>
<td>10B</td>
</tr>
<tr>
<td>GIT</td>
<td>637M</td>
<td>40M</td>
<td>70M</td>
</tr>
<tr>
<td>VLMo++</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>CoCa</td>
<td>1B</td>
<td>477M</td>
<td>623M</td>
</tr>
</tbody>
</table>

*Note: Some of the numbers here are based on our best estimate*

*: excluding the data used to pre-train the Florence image encoder

---

Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision
Learning Transferable Visual Models From Natural Language Supervision
Florence: A New Foundation Model for Computer Vision
SimVLM: Simple Visual Language Model Pretraining with Weak Supervision
An Empirical Study of Training End-to-End Vision-and-Language Transformers
Scaling Up Vision-Language Pre-training for Image Captioning
Flamingo: a Visual Language Model for Few-Shot Learning

GIT: A Generative Image-to-text Transformer for Vision and Language
VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts
CoCa: Contrastive Captioners are Image-Text Foundation Models
How do these models look like?

- **CLIP/ALIGN/Florence** models are dual-encoders that only use ITC loss for pre-training
- **LEMON/GIT/Flamingo** models are fusion encoders
  - **LEMON** uses a strong OD module to first extract image features offline
  - **GIT** uses a big Swin-like image encoder, but a small text decoder
  - On the other hand, **Flamingo** uses a relatively small image encoder, but a big text decoder
- Since only MLM or LM losses are used for pre-training, it is not friendly to *fast* retrieval
How do these models look like?

- **ALBEF/METER/CoCa** use an encoder-only design, but the fusion module can be used/called as a text decoder as well
  - In **METER**, ITC loss is not used; while in **ALBEF** and **CoCa**, ITC loss is used, which enables fast retrieval
  - There are also models like **BLIP** and **FIBER** that performs *fusion in the backbone*
- **SimVLM** uses an encoder-decoder design, and pre-trained with PrefixLM only
A closer look at SimVLM

- The model was pre-trained with PrefixLM on 1.8B image-text pairs as used in ALIGN (*private data*)
- **PrefixLM**: partition the text input randomly into two parts
- Additional text corpus can be naturally used for pre-training
- Strong performance after finetuning

SimVLM: Simple Visual Language Model Pretraining with Weak Supervision
A closer look at SimVLM

- Besides SOTA after finetuning, it also shows strong zero-shot generalization capability
  - Few-shot: using 1% training data
  - Using a prefix prompt “A picture of” improves the quality of decoded captions
- The generative approach also enables open-ended VQA naturally
- Also strong results on ImageNet linear probe

### Table 5: Linear evaluation on ImageNet classification, compared to state-of-the-art representation learning methods.
A closer look at LEMON

- The authors study the scaling law of VLP models for image captioning
- Scaling up model size (from 13M tiny to 675M huge) and data size (from 3M to 200M)
  - Strong performance boost for out-domain captioning with large-scale pre-training
  - Larger models benefit more from the large-scale data
A closer look at LEMON

- The authors also collect ALT200M
- No human annotation required
- Cover rich visual concepts, while some texts are not well-formed sentences, some do not exactly reflect image content

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Images (M) #cap./image</th>
<th>Unigram #unique</th>
<th>#unique in 0.1% tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCO Caption [5]</td>
<td>0.1</td>
<td>19,264</td>
<td>1,184</td>
</tr>
<tr>
<td>CC3M [40]</td>
<td>3.1</td>
<td>49,638</td>
<td>22,677</td>
</tr>
<tr>
<td>CC12M [4]</td>
<td>12.2</td>
<td>1,319,284</td>
<td>192,398</td>
</tr>
<tr>
<td>ALT200M (Ours)</td>
<td>203.4</td>
<td>2,067,401</td>
<td>1,167,304</td>
</tr>
</tbody>
</table>

Figure 3. Word cloud of the top 200 words in our pre-training dataset ALT200M, excluding the stop words, e.g., a, the, of, etc.
A closer look at LEMON

• SOTA on public benchmarks (COCO, nocaps)
• Zero-shot image captioning, capable of recognizing diverse visual contents
  • B: no pretrain
  • F: pre-training + finetuning
  • Z: pre-training only

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>Pre-training data</th>
<th>in-domain</th>
<th>near-domain</th>
<th>out-of-domain</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CIDEr</td>
<td>SPICE</td>
<td>CIDEr</td>
<td>SPICE</td>
</tr>
<tr>
<td>15</td>
<td>Human</td>
<td></td>
<td>80.6</td>
<td>15.0</td>
<td>84.6</td>
<td>14.7</td>
</tr>
<tr>
<td>16</td>
<td>SimVLsane</td>
<td>1.8B</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>SimVLMlarge</td>
<td>1.8B</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>SimVLMsane</td>
<td>1.8B</td>
<td>109.0</td>
<td>14.6</td>
<td>110.8</td>
<td>14.6</td>
</tr>
<tr>
<td>19</td>
<td>LEMONlarge</td>
<td>ALT200M</td>
<td>111.2</td>
<td>15.6</td>
<td>112.3</td>
<td>15.2</td>
</tr>
<tr>
<td>20</td>
<td>LEMONlarge</td>
<td>ALT200M</td>
<td>112.8</td>
<td>15.2</td>
<td>115.5</td>
<td>15.1</td>
</tr>
</tbody>
</table>
A closer look at GIT

- GIT is a simple generative image-to-text transformer, which is pre-trained on 800M image-text pairs \((\text{public+private})\) via a simple LM loss
  - Instead of using an OD module, the authors use a big Swin-like model as image encoder
- The image encoder is first pre-trained via image-text contrastive loss
  - The same strategy is also used in Flamingo, while in CoCa, contrastive and captioning losses are used together

VQA tasks are tackled as a text generation tasks

Video tasks are tackled via concatenating image frame features
A closer look at GIT

- GIT achieves SOTA over 12 image/video captioning and QA tasks, including the first human parity on TextCaps (no OCR engine is used)

<table>
<thead>
<tr>
<th>Image captioning</th>
<th>Image QA</th>
<th>Video captioning</th>
<th>Video QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCO*</td>
<td>nccaps*</td>
<td>VizWiz*</td>
<td>TextCaps*</td>
</tr>
<tr>
<td>ST-VQA*</td>
<td>VizWiz*</td>
<td>OCR-VQA*</td>
<td></td>
</tr>
<tr>
<td>MSVD</td>
<td>MRRRTT</td>
<td>VATEX*</td>
<td></td>
</tr>
<tr>
<td>MVSD-QA Frame</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior SOTA</td>
<td>138.7</td>
<td>120.6</td>
<td>94.1</td>
</tr>
<tr>
<td>GIT (ours)</td>
<td>148.8</td>
<td>123.4</td>
<td>114.4</td>
</tr>
<tr>
<td>Δ</td>
<td>+10.1</td>
<td>+3.7</td>
<td>+20.3</td>
</tr>
</tbody>
</table>

- GIT provides a generative scheme for image classification and scene text recognition

| Table 11: Results on scene text recognition. MJ and ST indicate the MJSynth (MJ) [36, 37] and SynthText (ST) [25] datasets used for training scene text recognition models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Fine-tuning</th>
<th>Regular Text</th>
<th>Irregular Text</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAM [58]</td>
<td>MJ-ST</td>
<td>95.3</td>
<td>90.6</td>
<td>93.9</td>
</tr>
<tr>
<td>Ro.Scanner [111]</td>
<td>MJ-ST</td>
<td>94.8</td>
<td>88.1</td>
<td>95.3</td>
</tr>
<tr>
<td>SRN [108]</td>
<td>MJ-ST</td>
<td>95.5</td>
<td>91.5</td>
<td>94.8</td>
</tr>
<tr>
<td>ABINet [16]</td>
<td>MJ-ST</td>
<td><strong>97.4</strong></td>
<td>93.5</td>
<td><strong>96.2</strong></td>
</tr>
<tr>
<td>S-GTR [29]</td>
<td>MJ-ST</td>
<td>96.8</td>
<td>94.1</td>
<td>95.8</td>
</tr>
<tr>
<td>GIT</td>
<td>TextCaps</td>
<td>94.2</td>
<td>91.5</td>
<td>92.9</td>
</tr>
<tr>
<td>GIT</td>
<td>MJ-ST</td>
<td><strong>97.3</strong></td>
<td><strong>95.2</strong></td>
<td>95.3</td>
</tr>
</tbody>
</table>

Generate class names token-by-token
A closer look at GIT

Figure 1: Example captions generated by GIT. The model demonstrates strong capability of recognizing scene text, tables/charts, food, banknote, logos, landmarks, characters, products, etc.
A closer look at CoCa

- All trained from scratch, including both image and text encoders
- Pre-trained via a combination of contrastive and generative losses
- Pre-training dataset: JFT-3B + 1.8B image-text pairs (*private*)
What are the publicly available pre-training data?

**WIT** features images and multilingual texts collected from the Wikipedia content pages.

**LAION-400M/5B**

However, it remains unclear how useful each dataset is.
Part II: In-Context Few-Shot Learning
What’s in-context few-shot learning?

- Getting SOTA via full finetuning is great. But can we train a model that can quickly adapt to different downstream tasks via only providing a few in-context examples?

The three settings we explore for in-context learning

**Zero-shot**
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

- Translate English to French:
- cheese

**One-shot**
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

- Translate English to French:
- sea otter => loutre de mer
- cheese

**Few-shot**
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

- Translate English to French:
- sea otter => loutre de mer
- peppermint => menthe poivrée
- plush giraffe => girafe peluche
- cheese

Traditional fine-tuning (not used for GPT-3)

**Fine-tuning**
The model is trained via repeated gradient updates using a large corpus of example tasks.

- sea otter => loutre de mer
- peppermint => menthe poivrée
- plush giraffe => girafe peluche
- cheese

 Aggregate Performance Across Benchmarks

- Few-shot
- One-shot
- Zero shot
Multimodal Few-Shot Learning with Frozen LM

• We need a strong language model that is kept frozen, and align an image encoder towards this language embedding space
  • This is an opposite design choice compared with LiT, since the end-goal is different
  • Frozen: in-context learning capability based on LM
  • LiT: zero-shot transfer on image classification based on CLIP

Figure 2: Gradients through a frozen language model’s self attention layers are used to train the vision encoder.

Figure 2. Design choices for contrastive-tuning on image-text data. Two letters are introduced to represent the image tower and text tower setups. L stands for locked variables and initialized from a pre-trained model, U stands for unlocked and initialized from a pre-trained model, u stands for unlocked and randomly initialized. Lu is named as “Locked-image Tuning” (LiT).
Multimodal Few-Shot Learning with Frozen LM

• Interesting in-context few-shot learning can be inherited from the frozen LM
  • Image encoder: NF-ResNet-50; however, an image is compressed into 2 global vectors
  • LM: 7B parameter transformer pre-trained on C4

Figure 1: Curated samples with about five seeds required to get past well-known language model failure modes of either repeating text for the prompt or omitting text that does not pertain to the image. These samples demonstrate the ability to generate open-ended outputs that adapt to both images and text, and to make use of facts that it has learned during language-only pre-training.
Multimodal Few-Shot Learning with Frozen LM

- How to perform in-context few-shot learning

![Multimodal Few-Shot Learning with Frozen LM](image)

Table 1: Transfer from Conceptual Captions to VQAv2. The τ column indicates whether a model uses training data from the VQAv2 training set. The row denoted Frozen_train-blind is the blind baseline described in subsection 4.1. Frozen vQA is a baseline which mixes in VQAv2 training data.

<table>
<thead>
<tr>
<th>n-shot</th>
<th>Acc.</th>
<th>n=0</th>
<th>n=1</th>
<th>n=4</th>
<th>τ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frozen scratch</td>
<td>29.5</td>
<td>35.7</td>
<td>38.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen finetuned</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen train-blind</td>
<td>24.0</td>
<td>28.2</td>
<td>29.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen vQA</td>
<td>26.2</td>
<td>33.5</td>
<td>33.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen vQA-blind</td>
<td>48.4</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Oscar [23]</td>
<td>73.8</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Transfer from Conceptual Captions to OKVQA. The τ column indicates if a model uses training data from the OKVQA training set. Frozen does not train on VQAv2 except in the baseline row, and it never trains on OKVQA.

<table>
<thead>
<tr>
<th>n-shot</th>
<th>OKVQA Acc.</th>
<th>n=0</th>
<th>n=1</th>
<th>n=4</th>
<th>τ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frozen</td>
<td>5.9</td>
<td>9.7</td>
<td>12.6</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen finetuned</td>
<td>4.0</td>
<td>5.9</td>
<td>6.6</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen train-blind</td>
<td>4.2</td>
<td>4.1</td>
<td>4.6</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen vQA</td>
<td>3.3</td>
<td>7.2</td>
<td>0.0</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Frozen vQA-blind</td>
<td>19.6</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>MAVEx [42]</td>
<td>12.5</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

![Table 1](image)

![Table 2](image)

Figure 4: Examples of (a) the Open-Ended miniImageNet evaluation (b) the Fast VQA evaluation.
PICa: How about using GPT-3 directly?

- Model size: Frozen 7B vs. GPT-3 175B
  - However, getting gradients from GPT-3 would be non-trivial, which renders training an image encoder to align the language embedding space challenging

- PICa: Prompting GPT-3 via the use of Image Captions
  - Treating GPT-3 as an *implicit* and *unstructured* KB
  - Translate images into captions/tags so that GPT-3 can understand it
  - Focus on OK-VQA that requires external knowledge to correctly answer the question
  - 4 shots outperform supervised SOTA on OK-VQA; also reasonable results on VQA-v2
Why GPT-3 are so powerful for OK-VQA?

- It encodes encyclopedia and commonsense knowledge
- GPT-3 also generates answer rationales reasonably well

An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA
Limitations of this approach

- Converting images into captions could lose important visual information

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Oscar (Li et al. 2020)</td>
<td>Feature Emb.</td>
<td>x</td>
<td>73.8</td>
</tr>
<tr>
<td>Frozen</td>
<td>Feature Emb.</td>
<td>✓</td>
<td>38.2</td>
</tr>
<tr>
<td>PICa-Base</td>
<td>Caption</td>
<td>✓</td>
<td>53.2</td>
</tr>
<tr>
<td>PICa-Base</td>
<td>Caption+Tags</td>
<td>✓</td>
<td>54.3</td>
</tr>
<tr>
<td>PICa-Full</td>
<td>Caption</td>
<td>✓</td>
<td>55.9</td>
</tr>
<tr>
<td>PICa-Full</td>
<td>Caption+Tags</td>
<td>✓</td>
<td>56.1</td>
</tr>
<tr>
<td>PICa-Full†</td>
<td>GT-Caption-5</td>
<td>✓</td>
<td>59.7</td>
</tr>
</tbody>
</table>

- How about if we have enough computing resources and go beyond this?

(f) How many giraffes are there?  
**Context:** A herd of giraffe standing next to a wooden fence.  
**Answer:** 3  
**GT Answer:** [6, 7, 8, 9, 10, 11, 12]  
**Acc.:** 0.0
Flamingo for few-shot learning

- Both the visual encoder and the LLM are kept frozen
- Besides image/video-text pairs, the pre-training data also includes interleaved image-text data (M3W), which is crucial for in-context learning
- The visual encoder is pre-trained with CLIP-like loss beforehand
Flamingo for few-shot learning

- **Perceiver resampler**: takes as input a variable number of features and outputs a fixed number of “visual tokens”
- **Gated X-attn design**: bridge vision and text modalities, the tanh gates make the training dynamics of LLM not change too much at the early stage of training
Flamingo for few-shot learning

- Each text token cross-attends to the image that precedes it in the interleaved sequence
- The model is trained by autoregressive LM loss
## Flamingo for few-shot learning

### Input Prompt

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a chinchilla. They are mainly found in Chile.</td>
<td>a flamingo. They are found in the Caribbean and South America.</td>
</tr>
<tr>
<td>This is a shiba. They are very popular in Japan.</td>
<td></td>
</tr>
<tr>
<td>This is</td>
<td></td>
</tr>
<tr>
<td>What is the title of this painting? Answer: The Hallucinogenic Toreador.</td>
<td>Arles.</td>
</tr>
<tr>
<td>Where is this painting displayed? Answer: Louvres Museum, Paris.</td>
<td></td>
</tr>
<tr>
<td>What is the name of the city where this was painted? Answer:</td>
<td></td>
</tr>
<tr>
<td>Output: &quot;Underground&quot;</td>
<td>&quot;Soullomes&quot;</td>
</tr>
<tr>
<td>Output: &quot;Congress&quot;</td>
<td></td>
</tr>
<tr>
<td>Output: 3x6=18</td>
<td></td>
</tr>
<tr>
<td>2+1=3</td>
<td></td>
</tr>
<tr>
<td>5+6=11</td>
<td></td>
</tr>
<tr>
<td>Output: A propaganda poster depicting a cat dressed as French emperor</td>
<td>A portrait of Salvador Dali with a robot head.</td>
</tr>
<tr>
<td>Napoleon holding a piece of cheese.</td>
<td></td>
</tr>
<tr>
<td>Output: A pink room with a flamingo pool float.</td>
<td></td>
</tr>
</tbody>
</table>

---

Flamingo: a Visual Language Model for Few-Shot Learning
Flamingo results highlight

- Visual dialog capability powered by LLM
Other work besides in-context learning

- **FewVLM**: train a VL-T5-like base model with PrefixLM and MLM, and found that PrefixLM is helpful for zero/few-shot captioning, while MLM is good for zero/few-shot VQA.

- **TAP-C**: CLIP is few-shot learner for VQA and visual entailment.
  - For VQA, the authors propose to reformulate it as a retrieval task.
  - For visual entailment, caption and hypothesis (text-text pairs) are used in training, while image and hypothesis (image-text pairs) are used at inference.

**Visual Question Answering**
- **Question**: What's in the bowl behind the cake?
- **Template**: The [mask] is in the bowl behind the cake.
- **Prompts**: The bread is in the bowl behind the cake. The fruit is in the bowl behind the cake.
- **Answer**: fruit

**Visual Entailment**
- **Caption**: Yachts in harbor, city in background.
- **Hypothesis**: There are no boats in the water in front of the city skyline.
- **Answer**: Contradiction
Part III: Model Evaluation
Model evaluation is difficult

• What we do at the current stage
  • We evaluate on tasks/datasets like VQAv2, NLVR2, SNLI-VE, VCR, image-text retrieval, image captioning, RefCOCO etc.
  • These benchmarks have driven tremendous progress in the field (VQA score from 70 to 82)

• They are good, but may not be good enough
  • Especially given the fact that big models are surpassing human performance on certain tasks
  • We should try to avoid both over-claiming and under-claiming
  • Not just focus on topping the leaderboard, but also testing the learned abilities
  • An open question for the community

• Some robustness analysis in the field
  • Diagnostic tests
  • Challenge sets (OOD)
  • Adversarial attacks
  • Probing
Example diagnostic tests

Winoground: Compositionality

Commonsense

Rephrasing

Logical reasoning

Image editing

Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality
Visual Commonsense in Pretrained Unimodal and Multimodal Models
Seeing past words: Testing the cross-modal capabilities of pretrained V&L models on counting tasks
Cycle-Consistency for Robust Visual Question Answering
VQA-LOL: Visual Question Answering under the Lens of Logic
Towards Causal VQA: Revealing and Reducing Spurious Correlations by Invariant and Covariant Semantic Editing
OOD generalization

Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering
Roses Are Red, Violets Are Blue... but Should VQA Expect Them To?
GRIT: General Robust Image Task Benchmark
VLUE: A Multi-Task Benchmark for Evaluating Vision-Language Models
Adversarial VQA: A New Benchmark for Evaluating the Robustness of VQA Models

Human-Adversarial Visual Question Answering
Probing (among many other works)

Fig. 1: Illustration of the proposed VALUE framework for investigating pre-trained vision-and-language models. VALUE consists of a set of well-designed probing tasks that unveil the inner mechanisms of V+L pre-trained models across: (i) Multimodal Fusion Degree; (ii) Modality Importance; (iii) Cross-modal Interaction via probing visual coreferences; (iv) Image-to-image Interaction via probing visual relations; and (v) Text-to-text Interaction via probing learned linguistic knowledge.

Behind the Scene: Revealing the Secrets of Pre-trained Vision-and-Language Models
Vision-and-Language or Vision-for-Language? On Cross-Modal Influence in Multimodal Transformers
Are Vision-Language Transformers Learning Multimodal Representations? A Probing Perspective
Take-away messages

• What has been covered
  • **Big foundation models**: compared with big LMs, the development of big VL models is still in its infant stage
  • **Few-shot in-context learning**: you need a big LM at first
  • **Model evaluation**: we discussed diagnostic tests, OOD, adversarial examples, and probing analysis

• **Future challenges**
  • What’s next beyond simple model scaling? Are larger models always better?
  • Beyond simple text output, can we train a MM model that quickly adapt to tasks that require bounding box output as well in a few-shot manner?
  • What do we mean when we say we improve VQA score by another +0.5 points? As MM foundation models become stronger and stronger, what’s the next-generation benchmarking system?
Thank you!
Any Questions?