Thinking Fast and Slow: Efficient Text-to-Visual Retrieval with Transformers

Antoine Miech, Jean-Baptiste Alayrac,
Ivan Laptev, Josef Sivic, Andrew Zisserman

[CVPR 2021]

Presenters:

Sohan Patnaik and Siba Smarak Panigrahi

Reading Session XIV

Kharagpur Data Analytics Group
IIT Kharagpur

February 13, 2022

Outline

- Introduction
  - Contribution
- Thinking Fast and Slow For Retrieval
  - Thinking Slow with cross-attention
    - A novel architecture for fine-grained vision-text cross-attention
    - Bi-directional captioning objective for retrieval
  - Thinking Faster and better for retrieval
    - Fast indexable dual encoder models
    - Distilling the Slow model into the Fast model
    - Re-ranking the Fast results with the Slow model.
- Results
  - Improving cross-attention for retrieval
  - Benefits of re-ranking
  - Comparison to the state of the art
- Conclusion

Introduction

- **Task**: Language-based search of large-scale image and video datasets.

- **Approach**: Independently map text and vision to a joint-embedding space
  - **Dual-encoders (DE)**: fast, easily scalable to large number of images
  - **Vision-text transformer with cross-attention (CA)**: slow, improved retrieval accuracy, inapplicable in practice due to scaling problems

Contribution

- DE models with a novel distillation objective to transfer knowledge from accurate CA models

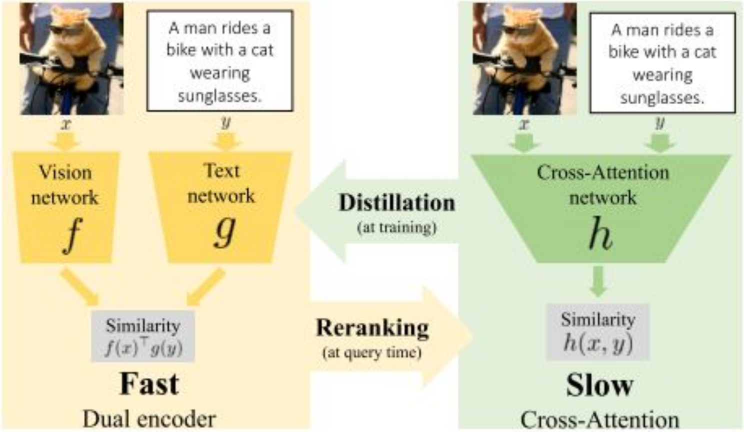
- DE and CA models combined with re-ranking where a few most promising candidates obtained with the Fast model are re-ranked using the Slow model

- Increased inference speed and competitive retrieval accuracy on both image and video domains
  - Flickr30k for Image domain
  - VATEX for Video domain

Thinking Fast and Slow for Retrieval

- **The Dual Encoder** (fast model) consists of extracting modality-specific embeddings.
  - The similarity between the image and text can be computed using dot product.

- **The Cross-Attention** (slow model) approach assumes that the similarity cannot be decomposed as simple dot product.
  - Richer interactions are allowed between the image and the text representations for better and computationally expensive scoring.



Thinking Slow with cross-attention

- Given an image x and and a text description y, the similarity h is computed as

$$h(x, y) = A(\phi(x), y)$$

where A denotes a network that uses cross-attention mechanism,
ϕ denotes the visual encoder (CNN).

- Above equation depicts how the text attends to the image or vice versa via multiple nonlinear functions.

- The authors proposed two novel methods
  - A fine-grained visual-text-cross-attention is used by increasing resolution of attended high-level image features.
  - A captioning loss is used to train the retrieval model instead of a ranking loss.

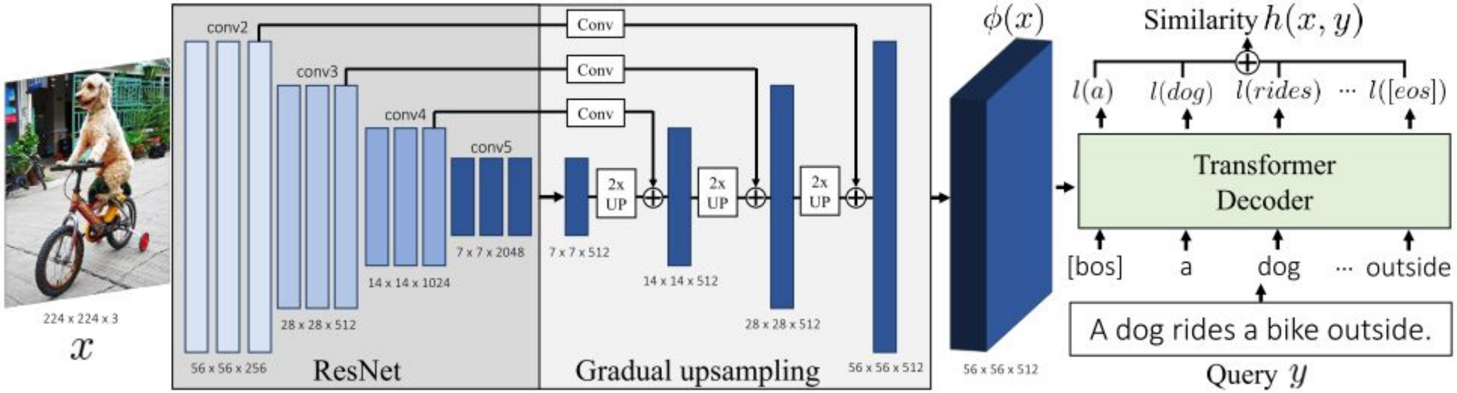
Novel fine-grained vision-text cross-attention

- **Visual features**: last convolutional layer of CNN. The feature map is flattened into a set of feature vectors that are used as input to vision-language cross-attention modules.
    - For example, a 224×224 input image passed through a ResNet-50 outputs a 7×7 feature map that is flattened into 49 vectors.

- While the last feature map produces high-level semantic information crucial for grounding text description into images, this last feature map is also severely downsampled.

Novel fine-grained vision-text cross-attention

- Solutions:
  - Increase the input image resolution
    - Increases the cost of running the visual backbone.
  - Gradually upsample the last convolutional feature map with a lightweight architecture conditioned on earlier higher resolution feature maps



Bi-directional captioning objective for retrieval

- Use of captioning model for retrieval

- Cross-attention module A as a stack of Transformer decoders
  - takes visual feature map $\phi(x)$ as an encoding state
  - each layer of the decoder consists of a masked text self-attention layer
  - a cross-attention layer that enables the text to attend to the visual features
  - a feed forward layer
  - **Note**: absence of self-attention layers on visual features
    - Allows visual feature map $\phi(x)$ to scale to thousands of vectors

Bi-directional captioning objective for retrieval

- Input text (y) of length L $y = [y^1, \ldots, y^L]$

$$h(x, y) = h_{fwd}(x, y) + h_{bwd}(x, y)$$

$$h_{fwd}(x, y) = \sum_{l=1}^{L} \log(p(y^l | y^{l-1}, \ldots, y^1, \phi(x); \theta_{fwd}))$$

- The parameters of the visual backbone, the forward and backward transformer models are obtained by minimizing $\mathcal{L}_{\mathrm{CA}} = -\sum_{i=1}^{n} h(x_i, y_i)$
- **Captioning loss and Contrastive loss**: for each ground truth token of the sequence a cross entropy loss is taken which effectively means that all other tokens in the vocabulary are considered as negatives!

Thinking Faster and better for retrieval

- The authors propose to distill the knowledge of the Slow cross-attention model into a Fast dual-encoder model that can be efficiently indexed.

- They combine the Fast dual-encoder model with the Slow cross-attention model via a re-ranking mechanism.

Fast indexable dual encoder models.

- The approach relies on calculation of the similarity between the text and the image as the dot product between their representations.
- The objective is to learn semantic embeddings f(x) for image and g(y) for the text where semantically relevant image-text pairs have higher similarity.
- For this, the authors use the standard noise contrastive loss as shown below.

$$\mathcal{L}_{\text{DE}} = -\sum_{i=1}^{n} \log \left(\frac{e^{f(x_i)^\top g(y_i)}}{e^{f(x_i)^\top g(y_i)} + \sum_{(x',y') \in \mathcal{N}_i} e^{f(x')^\top g(y')}} \right)$$

Distilling the Slow model into the Fast model.

- The authors propose an extension of the distillation approach
- Given an image-text pair (xi, yi), the authors sample a finite subset

  B = {(xi, yi)} U {(x, yi) | x != xi}.
- The probability distribution measuring the likelihood of the pair (x, y) ∈ Bi according to the Slow teacher model h(x, y) is

$$p(\mathcal{B}_i)(x, y) = \frac{\exp(h(x, y)/\tau)}{\sum_{(x', y') \in \mathcal{B}_i} \exp(h(x', y')/\tau)}$$

Distilling the Slow model into the Fast model.

- Similar distribution from the Fast student model is as follows

$$q(\mathcal{B}_i)(x, y) = \frac{\exp(f(x)^\top g(y)/\tau)}{\sum_{(x', y') \in \mathcal{B}_i} \exp(f(x')^\top g(y')/\tau)}$$

- Using both the distributions, the distillation loss was calculated as follows

$$\mathcal{L}_{\text{distill}} = \sum_{i=1}^{n} \mathcal{H}(p(\mathcal{B}_i), q(\mathcal{B}_i))$$

where H is the cross entropy between two probability distributions.

Distilling the Slow model into the Fast model.

- The final loss is the combination of standard contrastive loss and the distillation loss.

- Mathematically,

$$\min_{f,g} \mathcal{L}_{\text{distill}} + \alpha \mathcal{L}_{\text{DE}}$$

where $\alpha > 0$ determines the contribution of contrastive loss to the final loss.

Re-ranking the Fast results with the Slow model

- The authors see that only distillation cannot recover the full accuracy of the Slow model using the Fast model.

- They re-rank a few of the top retrieved candidates obtained using the Fast model using the approximate nearest neighbour search.

- Then the top K (e.g. 10 or 50) results are re-ranked by the Slow model.

- Mathematically,

$$\arg\max_{x \in \mathcal{X}_K} h(x, y) + \beta f(x)^\top g(y)$$

where $\beta > 0$ is a hyperparameter that weights the output scores of the two models.

Results

Model	Type	Train	F-R@1	F-R@5	C-R@1	C-R@5
Fast NCE BoW	DE	COCO	27.2	54.1	24.8	53.7
NCE BERT			24.4	48.0	24.2	52.0
PixelBERT	CA	COCO	30.0	55.1	25.1	52.5
VirTex Fwd only			33.4	58.1	31.8	61.2
VirTex			**38.1**	**62.8**	**35.1**	**64.6**
Fast NCE BoW	DE	CC	32.4	59.6	14.9	35.0
NCE BERT			25.8	50.7	12.2	29.8
PixelBERT	CA	CC	30.4	57.7	14.1	33.6
VirTex Fwd only			32.2	58.4	14.7	32.9
VirTex			**35.0**	**60.7**	**16.1**	**36.4**

Cross-attention models are better than Dual Encoders. Captioning models are surprisingly good for retrieval.

Benefits of our gradual upsampling architecture design.

Feature map	Size	F-R@1	F-R@5	C-R@1	C-R@5
Slow 96x96	384	**44.8**	**70.5**	**39.0**	**67.7**
Slow 56x56	224	42.2	66.8	38.5	65.2
Slow 28x28		40.4	66.3	37.4	66.8
Slow 14x14		39.2	63.8	36.8	64.9
VirTex `conv5` (7x7)	224	38.1	62.8	35.1	64.6
VirTex `conv4` (14x14)		38.9	64.4	34.9	63.5
VirTex `conv3` (28x28)		32.4	57.9	30.4	58.3
VirTex `conv2` (56x56)		20.6	41.1	18.3	43.0

Results

Model	Top K	Dist.	Train	F-R@1	F-R@5	C-R@1	C-R@5	F-Qt	C-Qt
Slow	✗	✗		44.8	70.4	39.0	67.7	4 s	19 s
	10	✗		44.0	63.0	38.6	61.5	**0.12 s**	**0.12 s**
	10	✓	COCO	47.2	70.1	40.5	67.8	**0.12 s**	**0.12 s**
Fast & Slow	50	✗		46.7	65.6	40.2	68.2	0.60 s	0.60 s
	50	✓		**47.6**	**73.2**	**40.9**	**70.0**	0.60 s	0.60 s
Slow	✗	✗		46.9	71.5	21.0	43.3	4 s	19 s
	10	✗		47.7	66.6	22.6	41.1	**0.12 s**	**0.12 s**
	10	✓	CC	48.4	67.4	22.7	43.4	**0.12 s**	**0.12 s**
Fast & Slow	50	✗		50.2	73.4	**23.8**	**46.9**	0.60 s	0.60 s
	50	✓		**50.5**	**73.6**	**23.8**	**46.9**	0.60 s	0.60 s

Combination of re-ranking and distillation provides better performance

Results

Method	Object Det.	Size	Train	Zero-shot	F-R@1	F-R@5	F-R@10
VILBERT [46]	✓	Full	CC	✓	31.9	61.1	72.8
Fast and *Slow* (K=100)	✗	384	CC	✓	**48.7**	**74.2**	**82.4**
VILBERT [46]	✓	Full	CC	✗	58.2	84.9	91.5
Fast and *Slow* (K=100)	✗	384	CC	✗	**68.2**	**89.7**	**93.9**
PixelBERT (R50) [29]	✗	800	COCO +VG	✗	59.8	85.5	**91.6**
Fast and *Slow* (R50, K=100)	✗	384	COCO	✗	**62.9**	**85.8**	91.3
Unicoder-VL [37]	✓	Full	CC + SBU	✗	71.5	90.9	94.9
UNITER [8]	✓	Full	COCO +CC +SBU +VG	✗	75.6	**94.1**	**96.8**
OSCAR [40]	✓	Full	COCO +CC +SBU +GQA	✗	**75.9**	93.3	96.6
Fast and *Slow* (K=100)	✗	384	COCO +CC	✗	72.1	91.5	95.2

Performance of proposed Fast and Slow model on Flickr30k images

K represents the retrieved images with DE encoders (distilled version)

Conclusion

- The authors introduced an accurate but Slow text-vision transformer based architecture with fine-grained cross attention for retrieval.

- They augment a fast scalable dual encoder through a combination of distillation and reranking.

- As a result, the combined approach achieves better results than the slow model and significantly reduces the inference time.

Thank You