Transformer is All You Need: Multimodal Multitask Learning with a Unified Transformer

날짜	@2021/03/15
연도	2021
학회	arXiv

0. Reference

- Paper Link
- Authors: Ronghang Hu (FAIR), Amanpreet Singh (FAIR)

1. Introduction

Problem Statement

- Transformers have shown great success in a wide range of ***domains***, including natural language, images, video and audio
- However, despite the success to ***specific domains***, there has not been much prior effort to *connect different tasks across domains* with transformers

> " *Is it possible to build a single, unified model that simultaneously handles tasks in a variety of domains ?* "

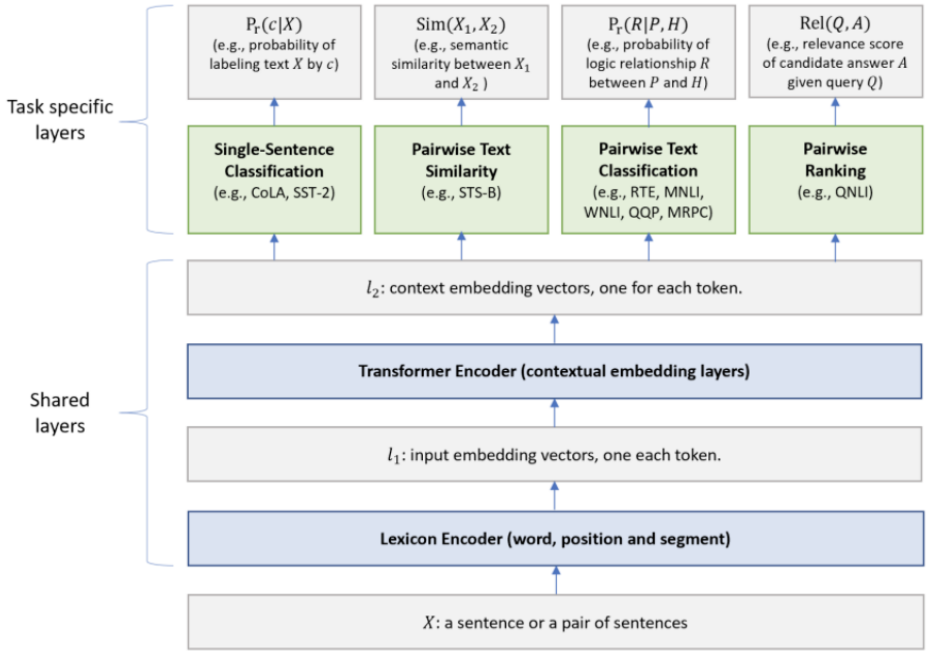
Previous Work

- Previous work tries to tackle some of the question but *only in limited scope:*

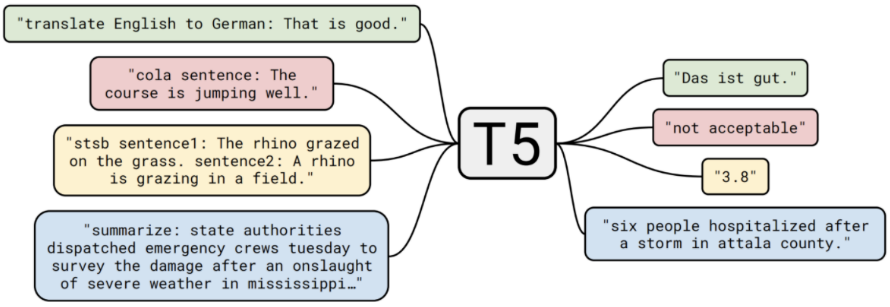
1. Work only tasks from a **single domain** or **specific multi-modals**
   - [ ] **Vit** and **DETR** focus on **vision-only** tasks
   - [ ] **BERT** and **RoBERTa** handle **language** tasks
   - [ ] **VisualBERT** and **VILBERT** work only on **specific multi-modal domain** of vision and language
2. Require *task-specific fine-tuning* for each task, **not leveraging any shared parameters** across tasks
   - [ ] Usually, end up with *N x parameters* for N tasks
3. Perform **multi-task** upon *related or similar tasks* only from a *single domain*
   - [ ] **MT-DNN** and **T5** work only on tasks in natural language

▼ Diagram

MT-DNN (Multi-Task Deep Neural Network)

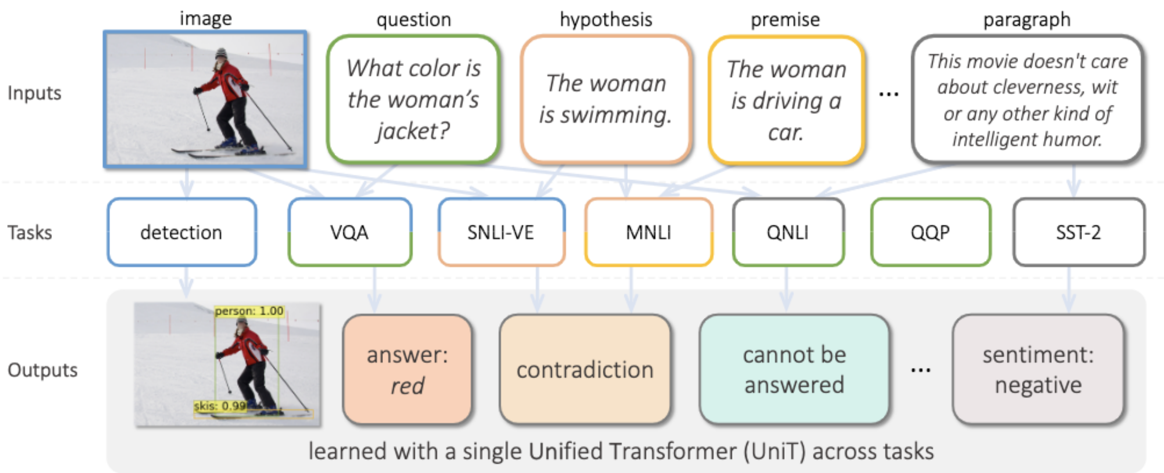


T5 (Text-to-Text Transfer Transformer)



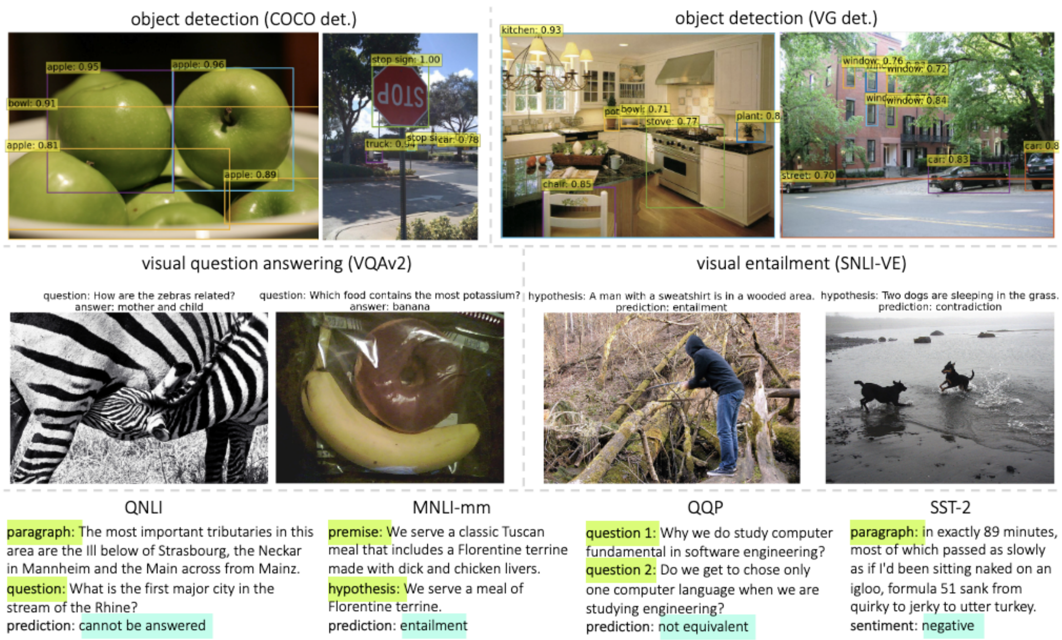
- [ ] **VILBERT-MT** works only on related vision-and-language tasks

Main Contribution



- Propose **UniT**, a **uni**fied **t**ransformer encoder-decoder architecture capable of learning *multiple tasks and domains* in a single model
- Jointly learn the most **prominent tasks** in visual and textual domains

▼ List of tasks

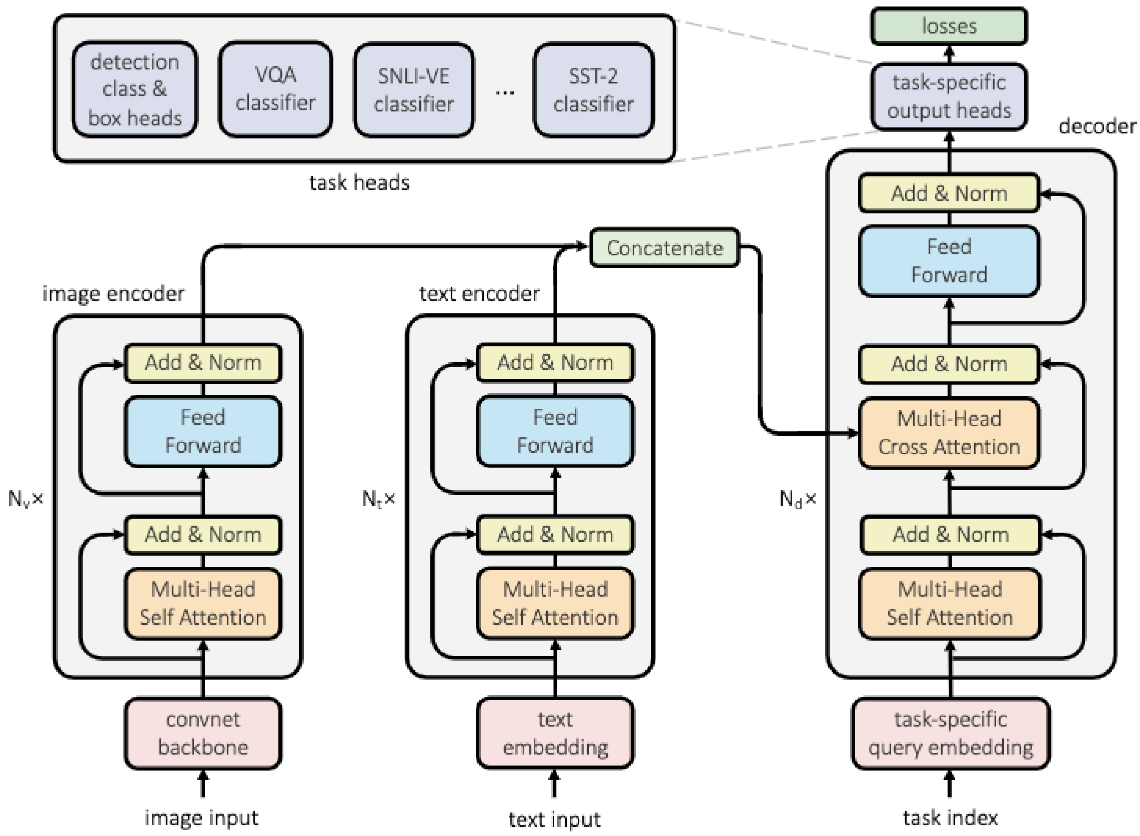


1. Object Dection **(COCO / Visual Genome)**
2. Visual Question Answering **(VQAv2)**
3. Visual Entailment **(SNLI-VE)**
4. Question-answering NLI, **QNLI** **(GLUE)**
5. Multi-Genre Natural Language Inference, **MNLI** **(GLUE)**
6. Quora Question Pairs, **QQP** **(GLUE)**
7. Stanford Sentimen t Treebank, **SST-2** **(GLUE)**

- Show that multi-modal tasks such as VQA and Visual Entailment benefit from multi-task training

2. Model Architecture

Overview



Encoder

- Two input modalities
    1. Image
        - [ ] First apply a CNN backbone to extract visual feature map
        - [ ] Then encoded by a Transformer encoder into a list of hidden states
        - ▼ Mathematical Expression

Our image encoding process is inspired by and similar to DETR [5]. First, a convolutional neural network backbone B is applied on the input image to extract a visual feature map $\mathbf{x}^v$ of size $H_v \times W_v \times d_v^b$ as

$$\mathbf{x}^v = B(I). \tag{1}$$

- B follows structure of ResNet-50 with dilation applied to its last C5 black, and is pre-trained on object detection in DETR

We apply a visual transformer encoder E_v with N_v layers and hidden size d_v^e on top of the feature map $\mathbf{x}^v$ to further encode it to visual hidden states $\mathbf{h}^v$ of size $L \times d_v^e$ (where $L = H_v \times W_v$ is the length of the encoded visual hidden states). In addition, given that different tasks (such as object detection and VQA) might require extracting different types of information, we also add a task embedding vector w_v^{task} into the transformer encoder to allow it to extract task-specific information in its output as follows.

$$\mathbf{h}^v = \{h_1^v, h_2^v, \cdots, h_L^v\} = E_v(P_{b\to e}(\mathbf{x}^v), w_v^{task}) \quad (2)$$

$P_{b\to e}$ is a linear projection from visual feature dimension d_v^b to encoder hidden size d_v^e. The structure of the visual transformer encoder E_v follows DETR [5], where positional encoding is added to the feature map. The task token w^{task} is a learned parameter of dimension d_v^e, which is concatenated to the beginning of the flattened visual feature list $P_{b\to e}(\mathbf{x}^v)$ and stripped from the output hidden states $\mathbf{h}^v$.

2. Text

- [ ] BERT is used to encode input words into a sequence of hidden states

▼ Mathematical Expression

size $S \times d_t^e$, where d_t^e is the BERT hidden size. Similar to the image encoder, in the text encoder, we also add a learned task embedding vector w_t^{task} as part of the BERT input by prefixing it at the beginning of the embedded token sequence, and later stripping it from the output text hidden states as follows.

$$\mathbf{h}^t = \{h_1^t, h_2^t, \cdots, h_S^t\} = \mathrm{BERT}(\{w_1, \cdots, w_S\}, w_t^{task}) \quad (3)$$

However, we find that it works nearly equally well in practice to keep only the hidden vector corresponding to [CLS] in $\mathbf{h}^t$ as input to the decoder, which saves computation.

Decoder

- Depending on the task, *single encoded modality* or the *both modalities* (concatenated) are provided to the decoder
- Explore either having *separate* (*i.e. task-specific*) or *shared decoders* among all tasks
- The representation from the decoder is passed to a *task-specific head* (*two-layer classifier*)

▼ Mathematical Expression

- Unlike encoders, decoder is built upon the same domain-agnostic transformer decoder across all tasks
- For vision-only tasks, $h^{enc} = h^v$
- For language-only tasks, $h^{enc} = h^v$
- For joint vision-and-language tasks, $h^{enc} = concat(h^v, h^t)$

The transformer decoder D takes the encoded input sequence $\mathbf{h}^{enc}$ and a task-specific query embedding sequence $\mathbf{q}^{task}$ of length q. It outputs a sequence of decoded hidden states $\mathbf{h}^{dec,l}$ for each of the l-th transformer decoder layer, which has the same length q as the query embedding $\mathbf{q}^{task}$.

$$\{\mathbf{h}^{dec,l}\} = D(\mathbf{h}^{enc}, \mathbf{q}^{task}) \tag{4}$$

- During experiment, use either
  1. A single shared decoder D^{all} for all tasks OR
  2. Separate decoder D_i^{task} for each specific task i

Task-specific Head

Apply a *task-specific head* for each task t for final prediction

▼ Object Detection

- Add a *class head* to produce a classification output

- Add a *box head* to produce a bounding box output
- For Visual Genome, also add an *attribute classification head*

processed into object bounding boxes. Following DETR, we apply these heads to all layers l in the decoder hidden states $\mathbf{h}^{dec,l}$ during training as

$$\begin{aligned} \mathbf{c}^l &= \text{class\_head}(\mathbf{h}^{dec,l}) && (5)\\ \mathbf{b}^l &= \text{box\_head}(\mathbf{h}^{dec,l}) && (6)\\ \mathbf{a}^l &= \text{attr\_head}(\mathbf{h}^{dec,l}, \mathbf{c}^l) && (7) \end{aligned}$$

where $\mathbf{c}^l$, $\mathbf{b}^l$, and $\mathbf{a}^l$ are class, box and attribute output sequences, all having the same length q as the query embedding $\mathbf{q}^{task}$ for detection.

- At test time, only take the prediction from the top decoder layer, h^{dec,N_d}

▼ All Other Tasks

- Visual QA, Visual Entailment and Natural Language Understanding
- All can be cast as a *classification task* among c_t classes for each task t
- For each classifier, use a *two-layer perceptron* with *GeLU* activation

$$\begin{aligned} \mathbf{p} &= \mathrm{W}_1 \cdot \mathrm{GeLU}(\mathrm{W}_2 \cdot \mathbf{h}_1^{dec,top} + \mathrm{b}_2) + \mathrm{b}_1 && (8)\\ \text{loss} &= \text{CrossEntropyLoss}(\mathbf{p}, \mathbf{t}) && (9) \end{aligned}$$

3. Experiment & Result

Sampling

- During training, manually specify a sampling probability for each task based on the dataset size and empirical evidence

Reshaping

- Apply scale and crop augmentation on image inputs during training for object detection
- However, no scale and crop for vision-and-language tasks

Preliminary Experiment

- First experiment with *objection detection* as a *vision-only* task and *VQA* as a *vision-and-language* task

#	Experiment setup	COCO det. mAP	VG det. mAP	VQAv2 accuracy
1	single-task	40.4 / –	4.02	66.25 / –
2	separate	40.7 / –	4.22	**68.36** / –
3	shared	38.5 / –	4.16	61.51 / –
4	shared (COCO init.)	**40.9** / 41.2	**4.56**	67.72 / 68.43

- Training with *separate* decoder outperforms *shared* decoder and *single-task* setting
- However *shared* decoder underperforms *single-task* model for COCO and VQA by a noticeable margin
    - This may be due to *relatively short training iterations* for *shared* decoder model
- Therefore, initialize the model from a model trained on COCO detection alone (**COCO init**)
    - In this case, joint model with *shared* decoders outperforms all *single-task* models

Main Result

#	decoder	COCO det.	VG det.	VQAv2	SNLI-VE	QNLI	MNLI-mm	QQP	SST-2
1	UniT – single-task training	40.4	4.02	66.25 / –	70.52 / –	91.62 / –	84.23 / –	91.18 / –	91.63 / –
2	UniT – separate	32.2	2.54	67.38 / –	74.31 / –	87.68 / –	81.76 / –	90.44 / –	89.40 / –
3	UniT – shared	33.8	2.69	67.36 / –	74.14 / –	87.99 / –	81.40 / –	90.62 / –	89.40 / –
4	UniT – separate (COCO init.)	38.9	3.22	67.58 / –	74.20 / –	87.99 / –	81.33 / –	90.61 / –	89.17 / –
5	UniT – shared (COCO init.)	39.0	3.29	66.97 / 67.03	73.16 / 73.16	87.95 / 88.0	80.91 / 79.8	90.64 / 88.4	89.29 / 91.5
6	DETR [5]	43.3	4.02	–	–	–	–	–	–
7	VisualBERT [30]	–	–	67.36 / 67.37	75.69 / 75.09	–	–	–	–
8	BERT [13] (bert-base-uncased)	–	–	–	–	91.25 / 90.4	83.90 / 83.4	90.54 / 88.9	92.43 / 93.7

Table 3: **Performance of our UniT model on 7 tasks across 8 datasets**, ranging from vision-only tasks (object detection on COCO and VG), vision-and-language reasoning (visual question answering on VQAv2 and visual entailment on SNLI-VE), and language-only tasks from the GLUE benchmark (QNLI, MNLI, QQP, and SST-2). For the line 5, 7 and 8, we also show results on VQAv2 test-dev, SNLI-VE test, and from GLUE evaluation server.

- Models trained on each *task separately* outperform all other variants *except multimodal tasks* *VQAv2* and *SNLI-VE*
    - This is **UNSURPRISING** as
        1. Unimodal tasks have low cross-modality overlap
        2. Each task is trained for full 500k iterations while less for UniT
        3. Vision-and-language tasks (*VQAv2* & *SNLI-VE*) consistently benefit from multi-task training together with vision-only and language-only tasks

- Despite a gap when comparing line 5 to lines 6,7,8 ; UniT achieves strong performance on each task with *a single generic model*

Ablations

#	Model configuration	COCO det. mAP	SNLI-VE accuracy	MNLI-mm accuracy
1	UniT (default, d_t^d=768, N_d=6)	38.79	69.27	81.41
2	decoder layer number, N_d=8	40.13	68.17	80.58
3	decoder layer number, N_d=12	39.02	68.82	81.15
4	decoder hidden size, d_t^d=256	36.32	69.68	81.09
5	using all hidden states from BERT instead of just [CLS]	38.24	69.76	81.31
6	losses on all decoder layers for SNLI-VE and MNLI-mm	39.46	69.06	81.67
7	no task embedding tokens	38.61	70.22	81.45
8	batch size = 32	35.03	68.57	79.62

Table 4: Ablation analyses of our UniT model with different model configurations on COCO det., SNLI-VE, and MNLI.

- Decoder layers and hidden size
  - Drop in **object detection** with a *smaller decoder hidden size* *(line4)*
  - Rise in **object detection** but drop in **SNLI-VE** and **MNLI** with a *deeper decoder layer number* *(line2)*
- Using *all BERT outputs* as input to the decoder has a relatively minor impact *(line5)*
- Losses on all decoder layers *(line6)*
  - Benefit for **object detection** but not for **SNLI-VE** and **MNLI**
  - Likely because these tasks require *outputting a single label*
- No task embedding tokens has minor impact *(line7)*
- Smaller batch size hurts *(line8)*

4. Conclusion

- Show that the **Transformer framework** can be applied over *a variety of domains*
- This leads to jointly handle **multiple tasks** within *a single unified model*

> *" Our model makes a step towards building general-purpose intelligence agents capable of handling a wide range of applications in different domains, including visual perception, language understanding, and reasoning over multiple modalities "*