CS197 Harvard: AI Research Experiences

Fall 2022: Lecture 3 – "Shoulders of Giants"
Reading AI Research Papers

Instructed by Pranav Rajpurkar. Website https://cs197.seas.harvard.edu/

Abstract

Maybe you're trying to get into AI research, maybe you are a graduate student that is considering joining a new research lab, maybe you're in industry trying to present the latest advances on an AI problem to your colleagues. Regardless, you'll be faced with the daunting task of understanding the state of progress on the problem topic, and the gaps that are left to fill. I go through this exercise continually in my career, and a structured approach to understanding the state and gaps in something can make the task less daunting. This lecture is set up as a real walkthrough of the steps you would take to learn about a new topic in AI. My hope is that by the end of the lecture, you will have a blueprint for the kind of workflow you can use while approaching the reading of AI research papers.

DALL-E Generation: A voracious reader surrounded by papers around them, digital art

Learning outcomes:

- Conduct a literature search to identify papers relevant to a topic of interest
- Read a machine learning research paper and summarize its contributions

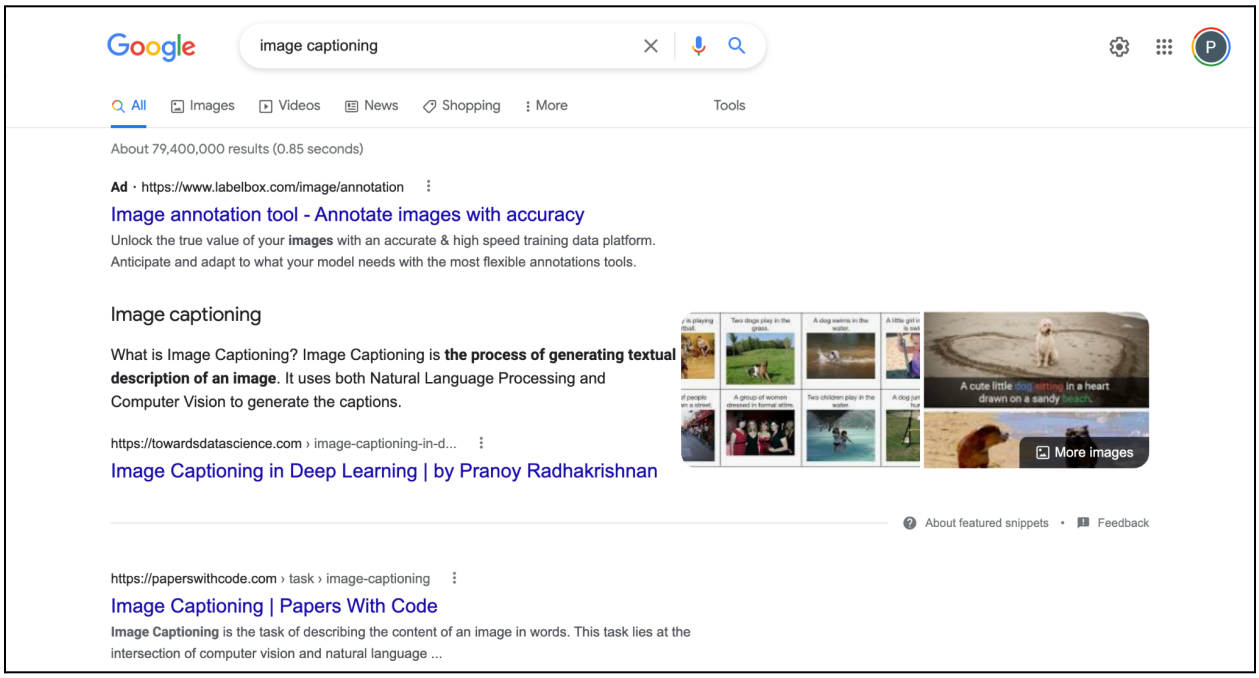
Approach

I'm going to break down the process of reading AI research papers into two pieces: reading *wide*, and reading *deep*. When you start learning about a new topic, you typically get more out of reading *wide*: this means navigating through literature reading small amounts of individual research papers. Our goal when reading wide is to build and improve our mental model of a research topic. Once you have identified key works that you want to understand well in the first step, you will want to read *deep*: here, you are trying to read individual papers in depth. Both reading *wide* and *deep* are necessary and complimentary, especially when you're getting started.

I am going to walk you through how you would approach reading wide and reading deep very concretely using the example of "deep learning for image captioning" as the hypothetical topic we want to break into. Let's get started.

Reading Wide

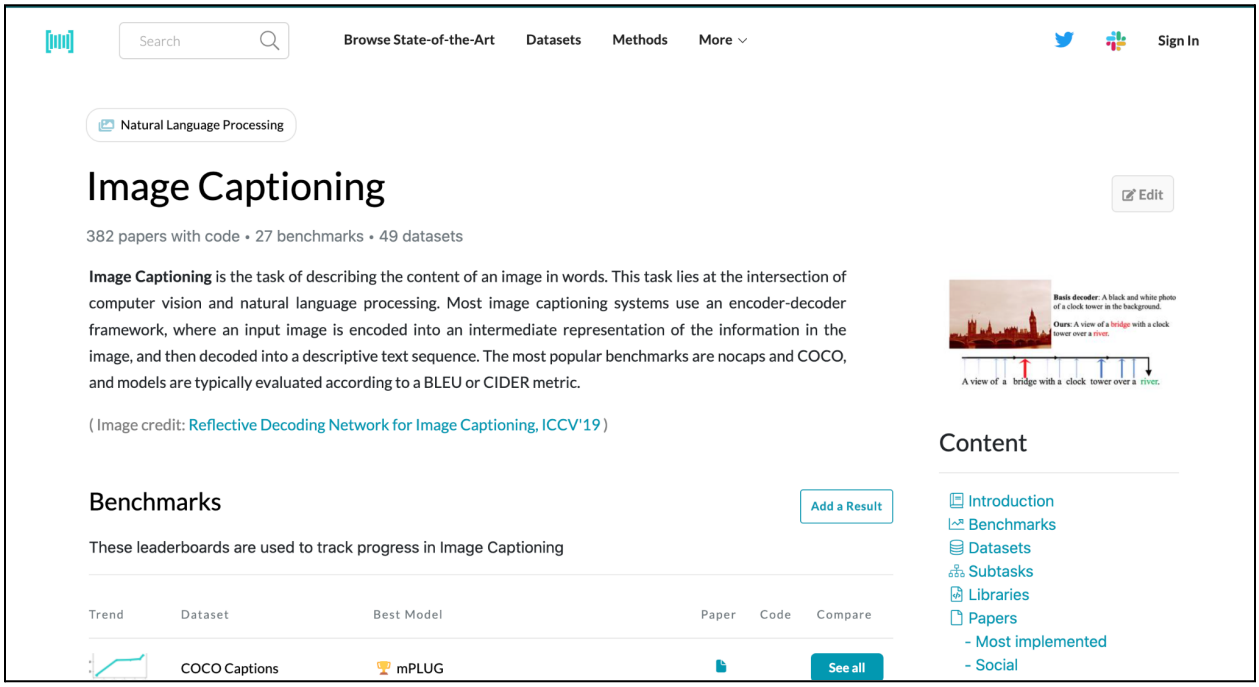
Let's start with a simple google search for "image captioning".



We get a definition of image captioning from the first result that defines the task for us. It's the second link that's interesting for us: Papers with Code

Papers with Code

Papers with Code is a community project with the mission to create a free and open resource with Machine Learning papers, code, datasets, methods and evaluation tables. I like it a lot and use it quite frequently.



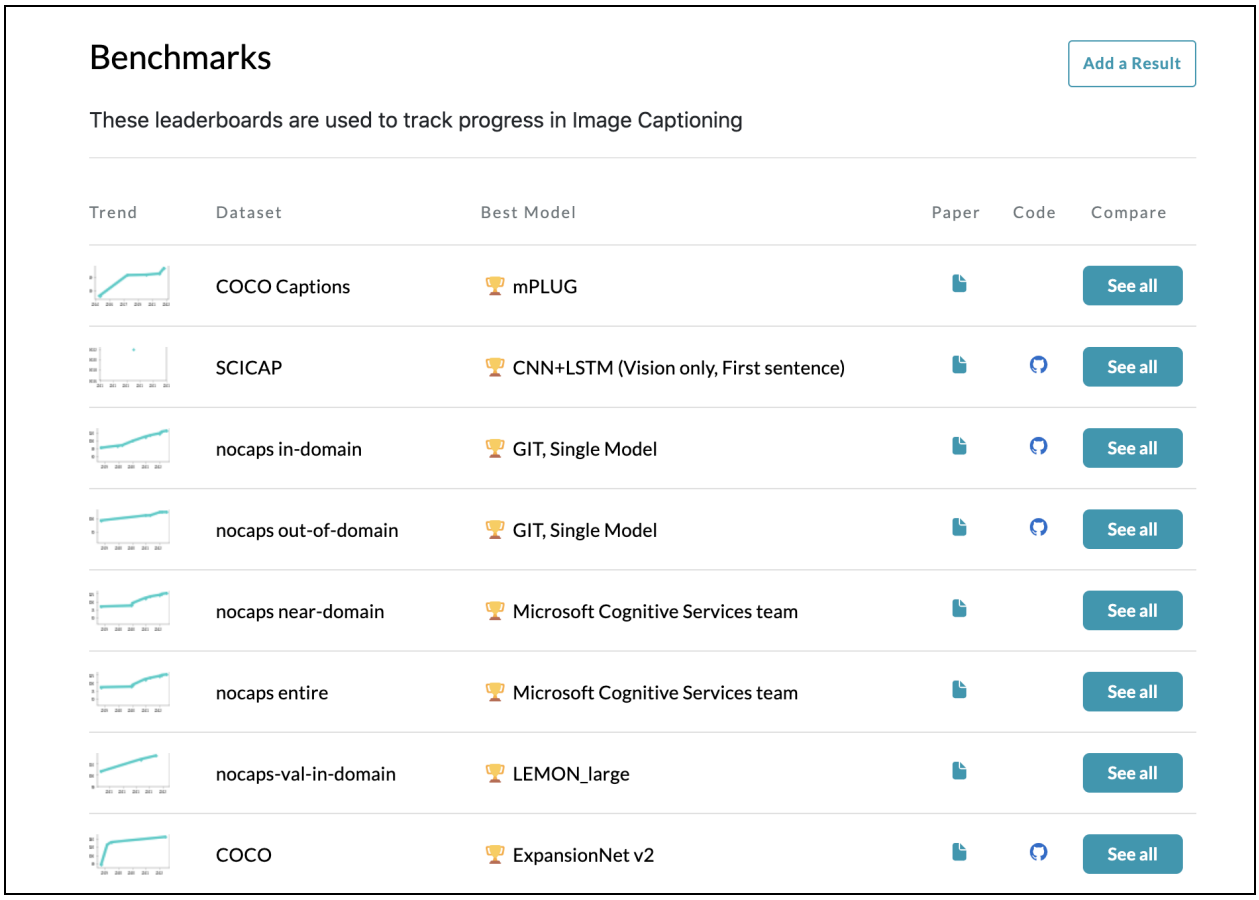
We're going to need to start making notes, so let's open up a Google Doc, and create an entry. (This is the final version of my notes).

Image Captioning Notes

https://paperswithcode.com/task/image-captioning

- Task: describing the content of an image in words.
- Methods: encoder-decoder framework – input image is encoded into an intermediate representation of the information in the image, and then decoded into a descriptive text sequence.
- Datasets: nocaps and COCO
- Evaluation: BLEU or CIDER metric.

Let's scroll to the benchmarks section on papers with code.



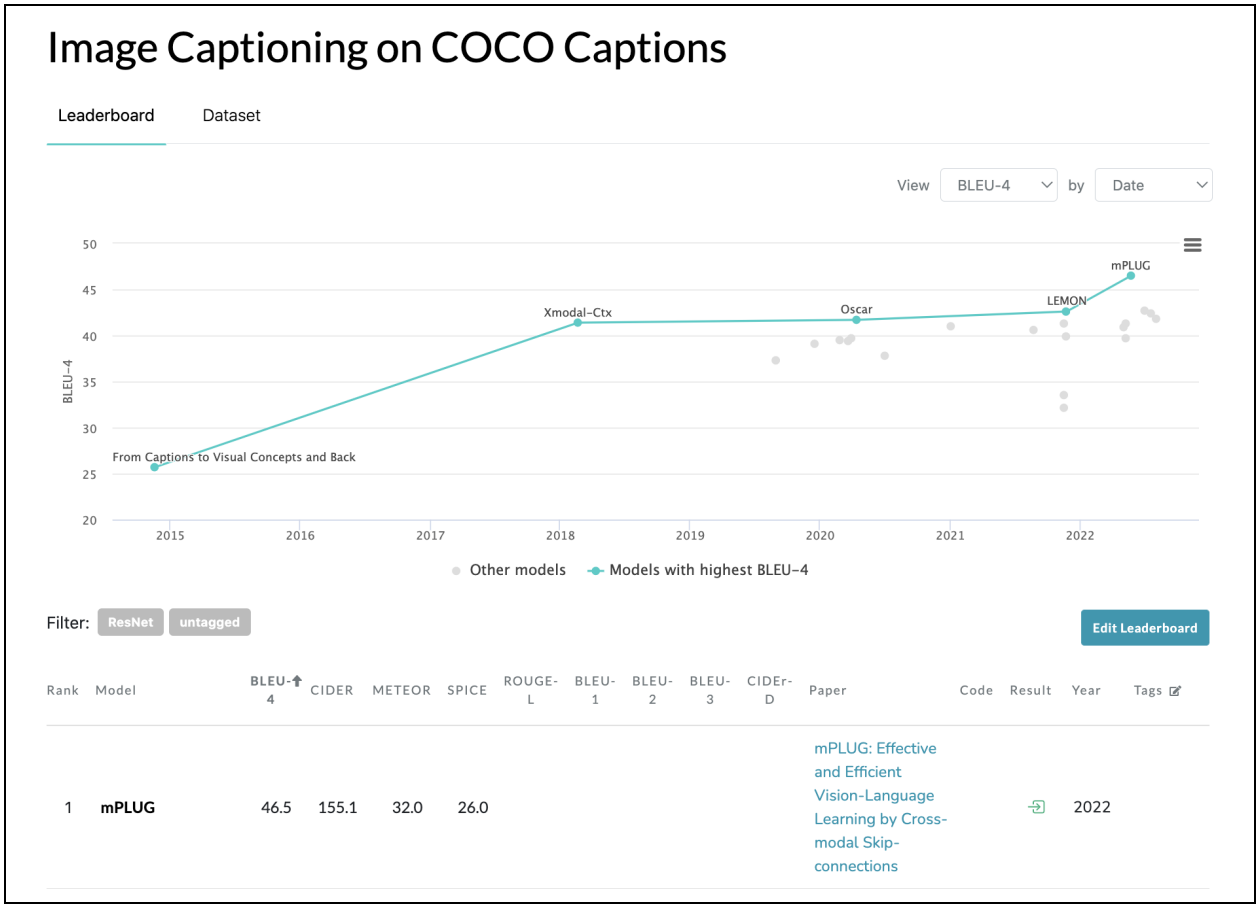
Benchmarks

Add a Result

These leaderboards are used to track progress in Image Captioning

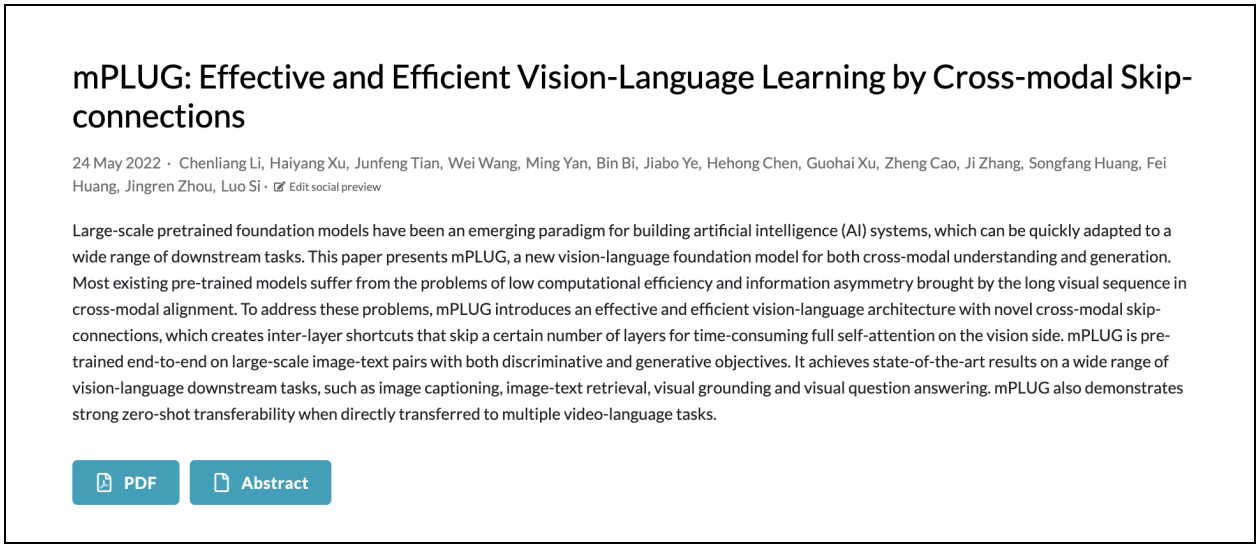
Trend	Dataset	Best Model	Paper	Code	Compare
	COCO Captions	mPLUG			See all
	SCICAP	CNN+LSTM (Vision only, First sentence)			See all
	nocaps in-domain	GIT, Single Model			See all
	nocaps out-of-domain	GIT, Single Model			See all
	nocaps near-domain	Microsoft Cognitive Services team			See all
	nocaps entire	Microsoft Cognitive Services team			See all
	nocaps-val-in-domain	LEMON_large			See all
	COCO	ExpansionNet v2			See all

We see a few benchmarks, with the words “COCO” and “nocaps” repeating. Let’s try to look at the top benchmark first: COCO captions.



This shows us that as recently as in mid 2022, there has been a state-of-the-art (you might hear SOTA (pronounced like soda with a 't') with a method called mPLUG. The leaderboard is quite useful: it shows us metrics including BLEU-4, CIDER, METEOR, SPICE, and similar variants. Soon, we'll need to understand what these are.

Let's click on the mPLUG paper link:



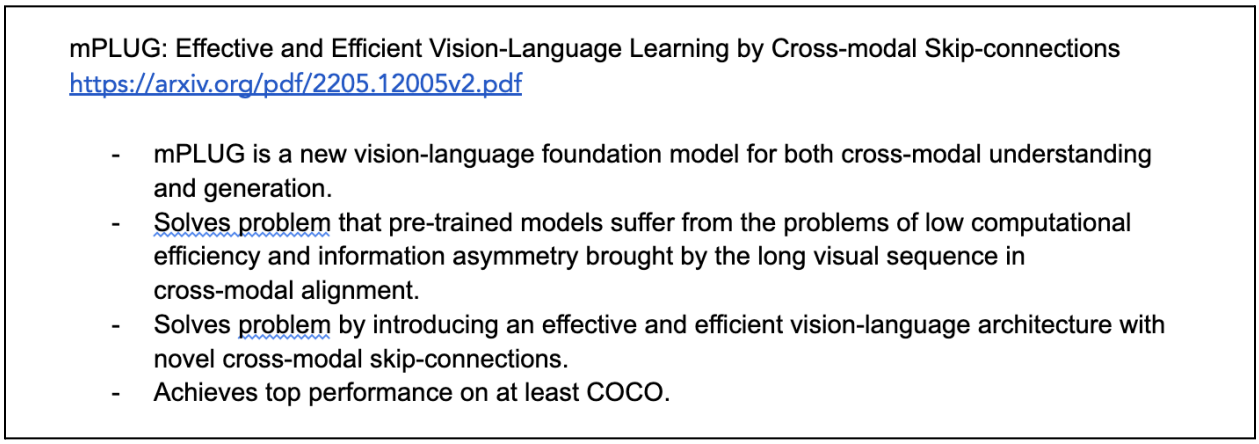
mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections

24 May 2022 · Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, Luo Si · Edit social preview

Large-scale pretrained foundation models have been an emerging paradigm for building artificial intelligence (AI) systems, which can be quickly adapted to a wide range of downstream tasks. This paper presents mPLUG, a new vision-language foundation model for both cross-modal understanding and generation. Most existing pre-trained models suffer from the problems of low computational efficiency and information asymmetry brought by the long visual sequence in cross-modal alignment. To address these problems, mPLUG introduces an effective and efficient vision-language architecture with novel cross-modal skip-connections, which creates inter-layer shortcuts that skip a certain number of layers for time-consuming full self-attention on the vision side. mPLUG is pre-trained end-to-end on large-scale image-text pairs with both discriminative and generative objectives. It achieves state-of-the-art results on a wide range of vision-language downstream tasks, such as image captioning, image-text retrieval, visual grounding and visual question answering. mPLUG also demonstrates strong zero-shot transferability when directly transferred to multiple video-language tasks.

PDF Abstract

Let's read the abstract. We are new to image captioning, so don't expect to understand most of the sentences. I am going to make note of key information here in our Google Doc.



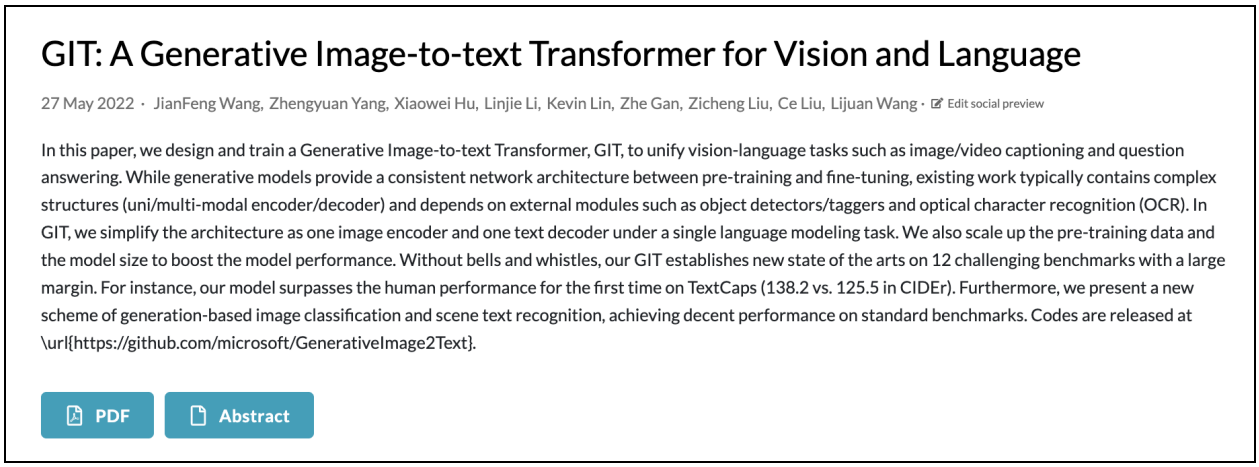
mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections
https://arxiv.org/pdf/2205.12005v2.pdf

- mPLUG is a new vision-language foundation model for both cross-modal understanding and generation.
- Solves problem that pre-trained models suffer from the problems of low computational efficiency and information asymmetry brought by the long visual sequence in cross-modal alignment.
- Solves problem by introducing an effective and efficient vision-language architecture with novel cross-modal skip-connections.
- Achieves top performance on at least COCO.

Again, it's okay that we don't understand what this all means just yet; this is simply an attempt to extract the key idea first.

I am going to go back to Papers with Code and repeat this exercise for the winner on 2 more benchmarks: I picked nocaps-in domain and nocaps near-domain because they both had very recent submissions that had high performance.

Thus we come across the GIT method.



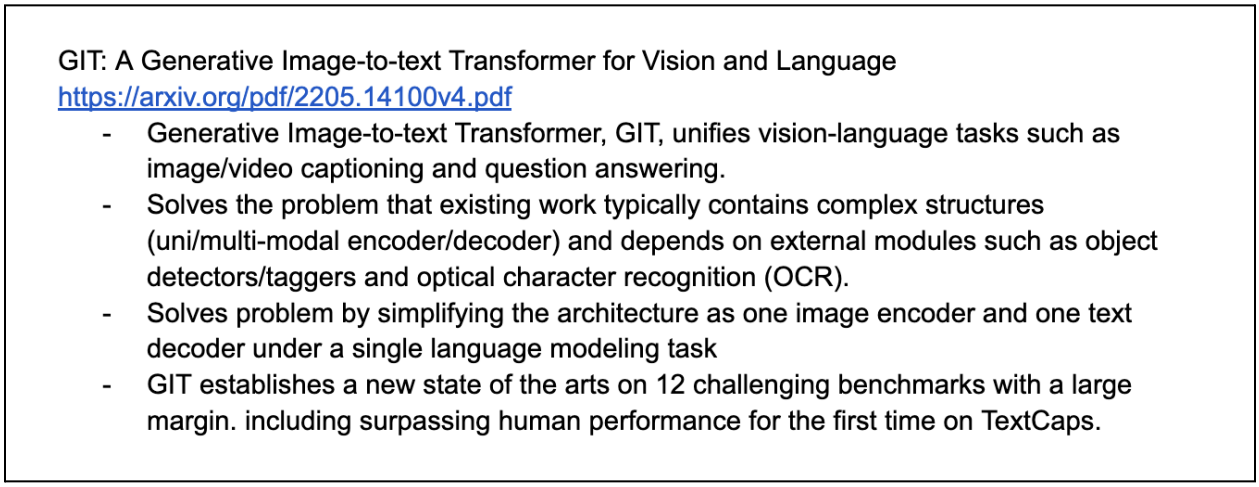
GIT: A Generative Image-to-text Transformer for Vision and Language

27 May 2022 · JianFeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, Lijuan Wang · Edit social preview

In this paper, we design and train a Generative Image-to-text Transformer, GIT, to unify vision-language tasks such as image/video captioning and question answering. While generative models provide a consistent network architecture between pre-training and fine-tuning, existing work typically contains complex structures (uni/multi-modal encoder/decoder) and depends on external modules such as object detectors/taggers and optical character recognition (OCR). In GIT, we simplify the architecture as one image encoder and one text decoder under a single language modeling task. We also scale up the pre-training data and the model size to boost the model performance. Without bells and whistles, our GIT establishes new state of the arts on 12 challenging benchmarks with a large margin. For instance, our model surpasses the human performance for the first time on TextCaps (138.2 vs. 125.5 in CIDEr). Furthermore, we present a new scheme of generation-based image classification and scene text recognition, achieving decent performance on standard benchmarks. Codes are released at \url{https://github.com/microsoft/GenerativeImage2Text}.

PDF Abstract

You can follow my above example to make similar notes using this abstract. Note that we haven't yet opened the papers: that's intentional. Here are the notes I made:



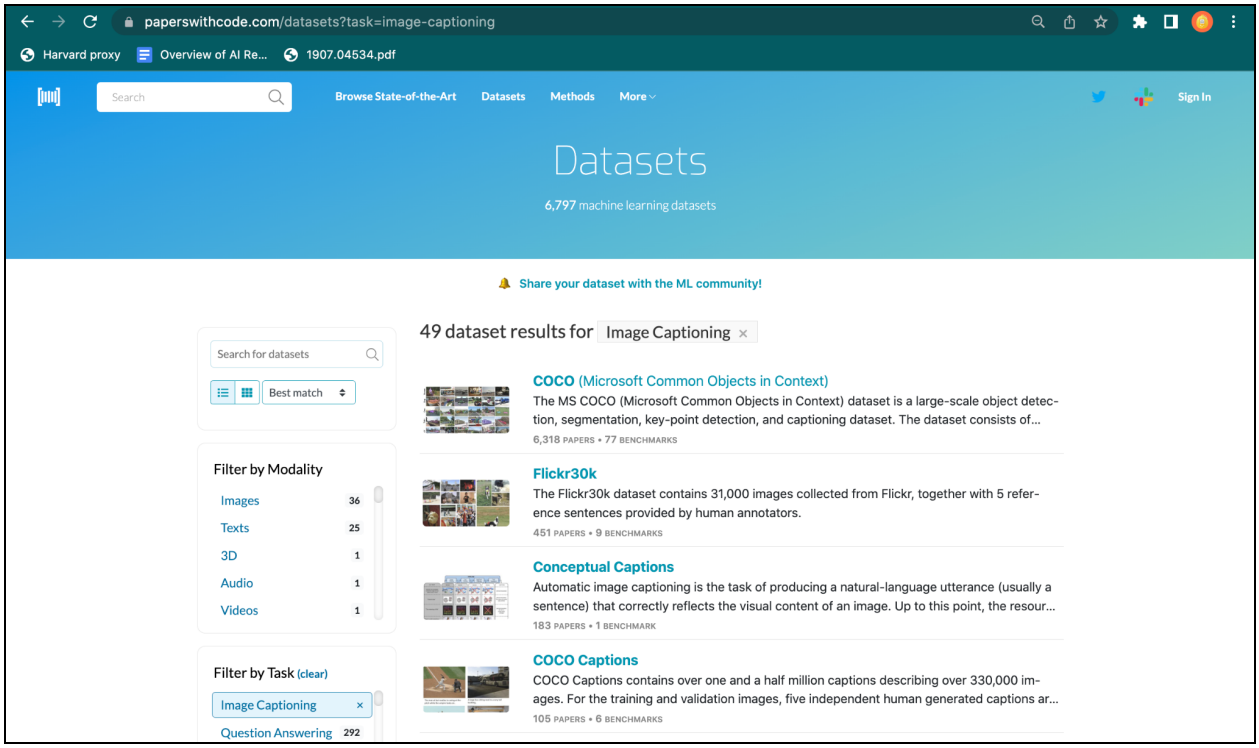
GIT: A Generative Image-to-text Transformer for Vision and Language
https://arxiv.org/pdf/2205.14100v4.pdf

- Generative Image-to-text Transformer, GIT, unifies vision-language tasks such as image/video captioning and question answering.
- Solves the problem that existing work typically contains complex structures (uni/multi-modal encoder/decoder) and depends on external modules such as object detectors/taggers and optical character recognition (OCR).
- Solves problem by simplifying the architecture as one image encoder and one text decoder under a single language modeling task
- GIT establishes a new state of the arts on 12 challenging benchmarks with a large margin. including surpassing human performance for the first time on TextCaps.

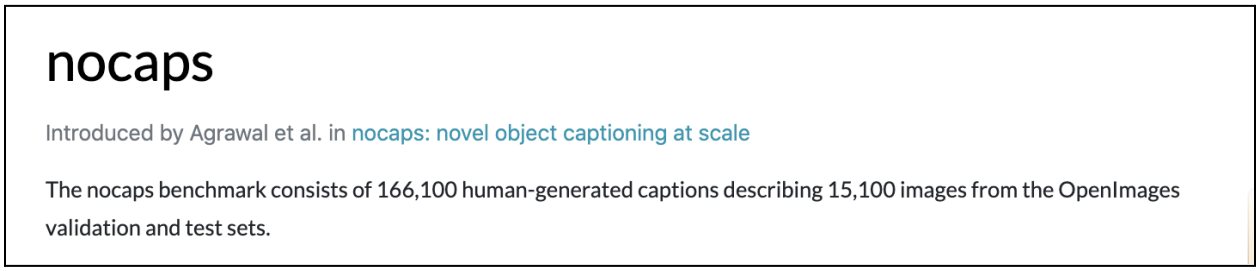
I haven't used any external knowledge here; it's simply organizing key information from the abstract.

Exercise: Perform a similar summarization of the abstract for another method using the benchmarks. Look for active image captioning benchmarks where you see recent submissions, especially if the recent submissions are setting new SOTA.

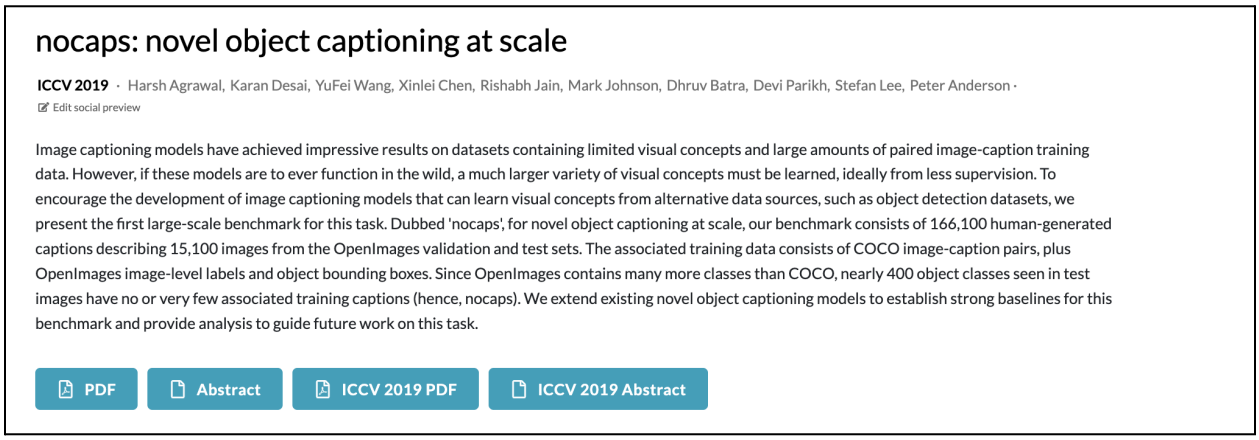
Now that we've got an understanding of a couple of methods, let's try to understand the datasets a little better. We can still stick to papers with code. Let's find our way to the image captioning datasets:



We'll look into the 2 datasets we have come across before: nocaps, and COCO captions. Let's start with nocaps.



We can click the link to get to the abstract of the paper introducing the dataset.



As we did for the models, we can similarly make notes for the datasets.

nocaps: novel object captioning at scale
https://arxiv.org/pdf/1812.08658v3.pdf

- Dubbed 'nocaps', for **n**ovel **o**bject **cap**tioning at **s**cale, benchmark consists of 166,100 human-generated captions describing 15,100 images from the OpenImages validation and test sets.
- Introduced to encourage the development of image captioning models that can learn visual concepts from alternative data sources, such as object detection datasets.
- The associated training data consists of COCO image-caption pairs, plus OpenImages image-level labels and object bounding boxes.
- Nearly 400 object classes seen in test images have no or very few associated training captions.
    - I came across *nocaps out-of-domain and nocaps in-domain earlier; maybe that's associated with above?*

Notice that I am leaving myself comments to come back to in my notes. Like for the methods, some words here might be new, but most words here are in typical English, so it's more easy to understand. We can do the same for the other dataset, COCO captions.

Microsoft COCO Captions: Data Collection and Evaluation Server

1 Apr 2015 · Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, C. Lawrence Zitnick · Edit social preview

In this paper we describe the Microsoft COCO Caption dataset and evaluation server. When completed, the dataset will contain over one and a half million captions describing over 330,000 images. For the training and validation images, five independent human generated captions will be provided. To ensure consistency in evaluation of automatic caption generation algorithms, an evaluation server is used. The evaluation server receives candidate captions and scores them using several popular metrics, including BLEU, METEOR, ROUGE and CIDEr. Instructions for using the evaluation server are provided.

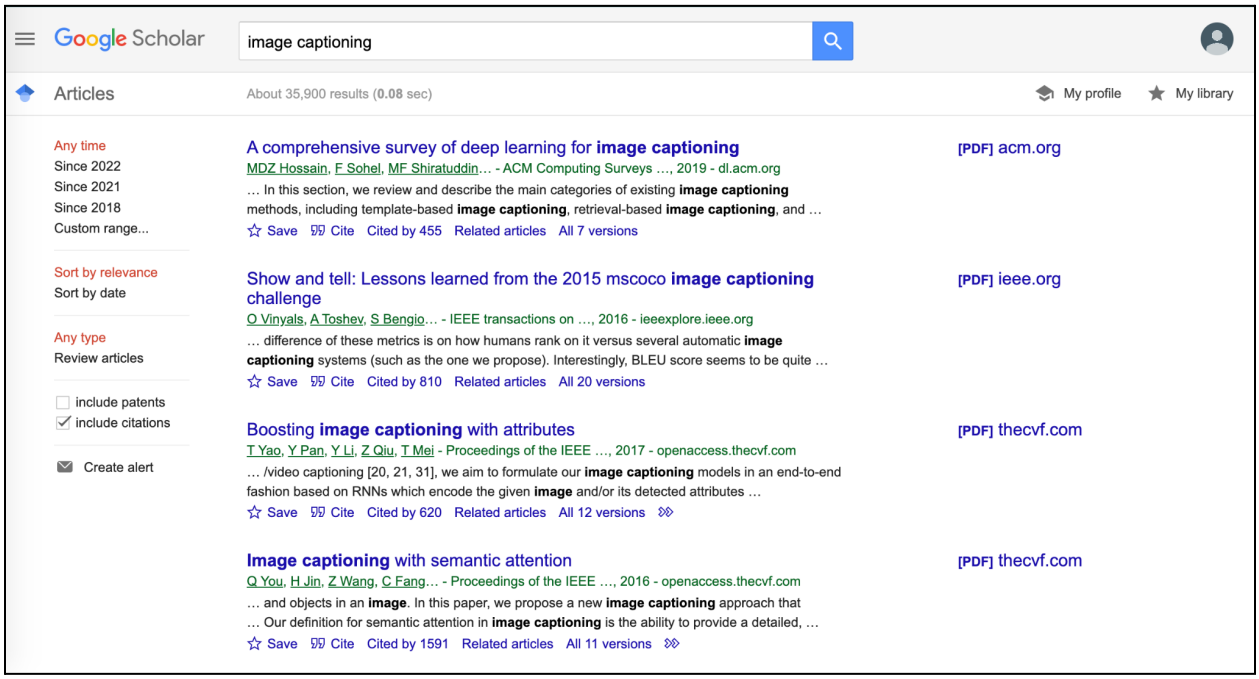
Microsoft COCO Captions: Data Collection and Evaluation Server
https://arxiv.org/pdf/1504.00325v2.pdf

- Dataset (hopefully now done) contains over one and a half million captions describing over 330,000 images.
- For the training and validation images, five independent human generated captions are provided.
- Uses an evaluation server – *not sure what that means?*
- Server scores captions using several popular metrics, including BLEU, METEOR, ROUGE and CIDEr.

At this point, we have a good collection of 4 papers: 2 recent SOTA methods and 2 datasets. Looking at SOTA methods might especially be useful if we're looking to improve on the methods, but if we're trying to understand a problem domain more broadly, we need to pick up some more mature and influential works here.

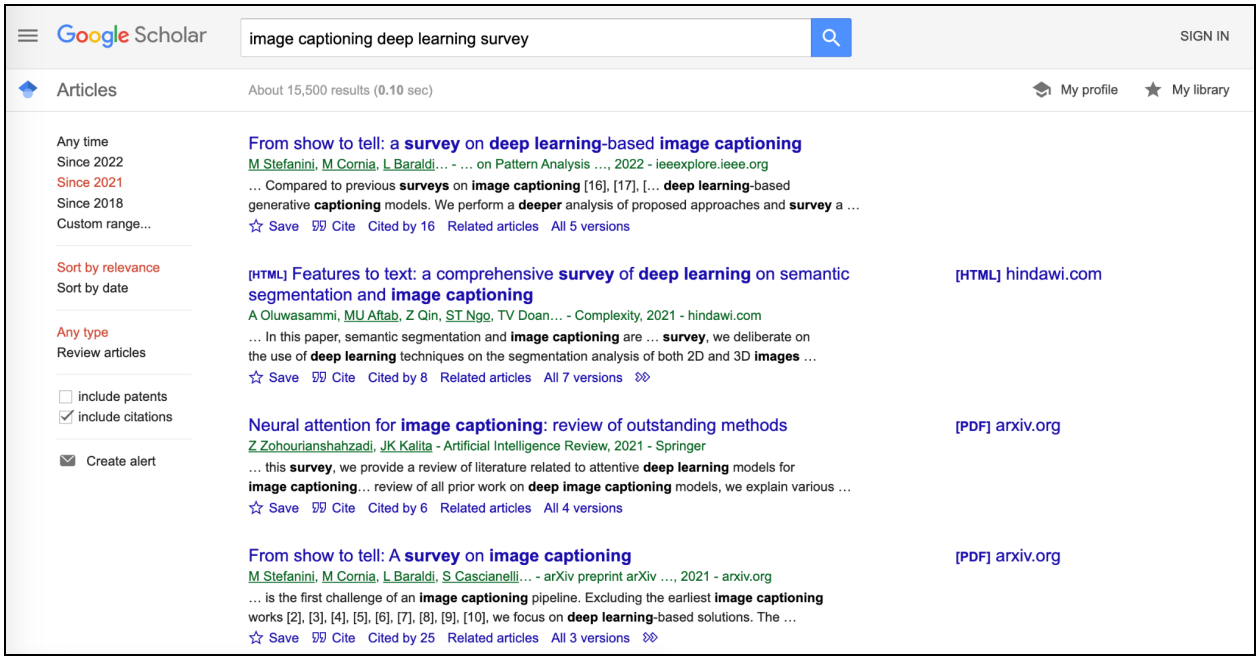
Google Scholar

We're going to turn next to Google Scholar. Let's start by entering our search term: image captioning:

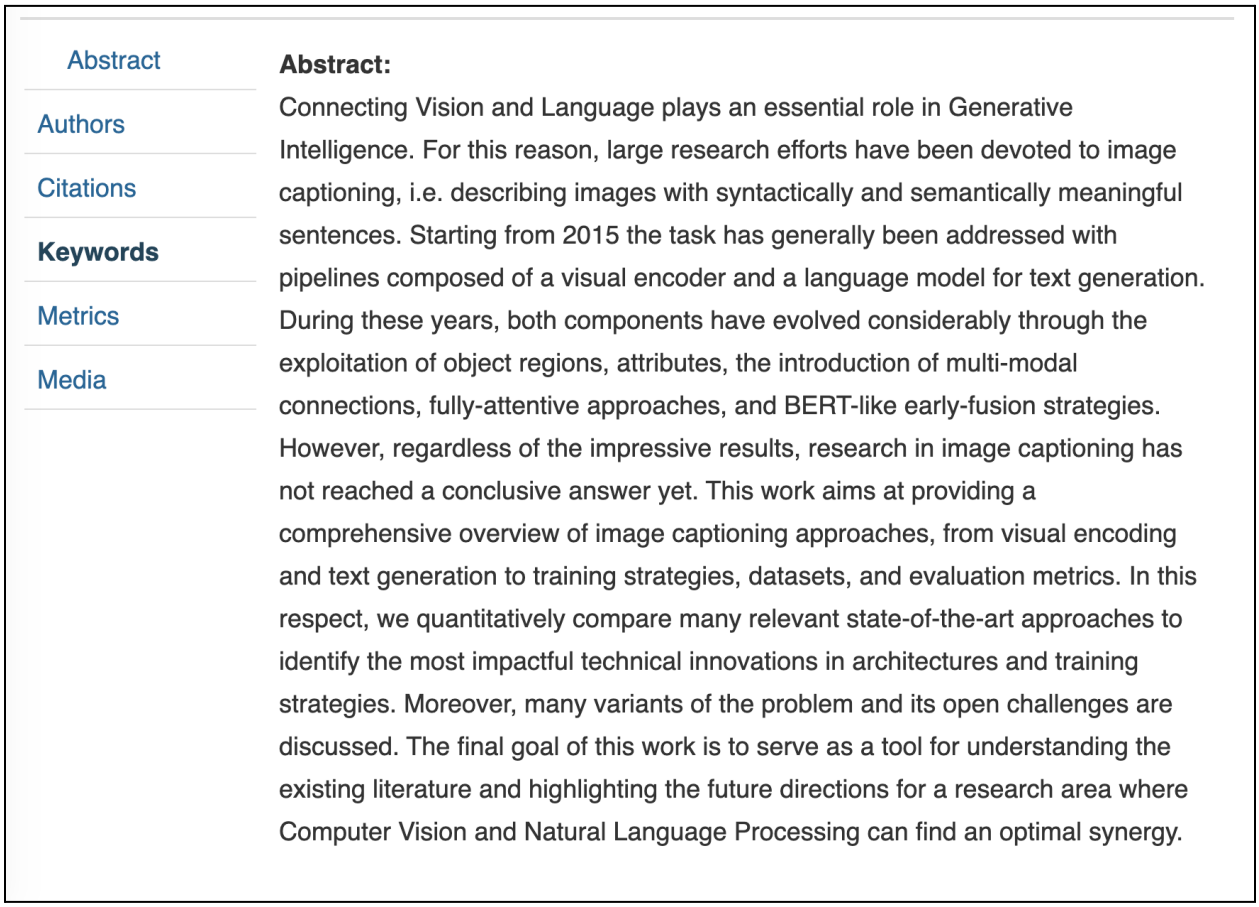


Google scholar sorts by relevance and includes a few useful details, including the number of citations that the paper has received. We're in luck – we have a survey paper at the top of our search results. A survey paper typically reviews and describes the state of a problem space, and often includes challenges and opportunities. Reviews/Surveys may not always be up-to-date, comprehensive, or completely accurate, but especially if we're new to a space, can get us up to speed.

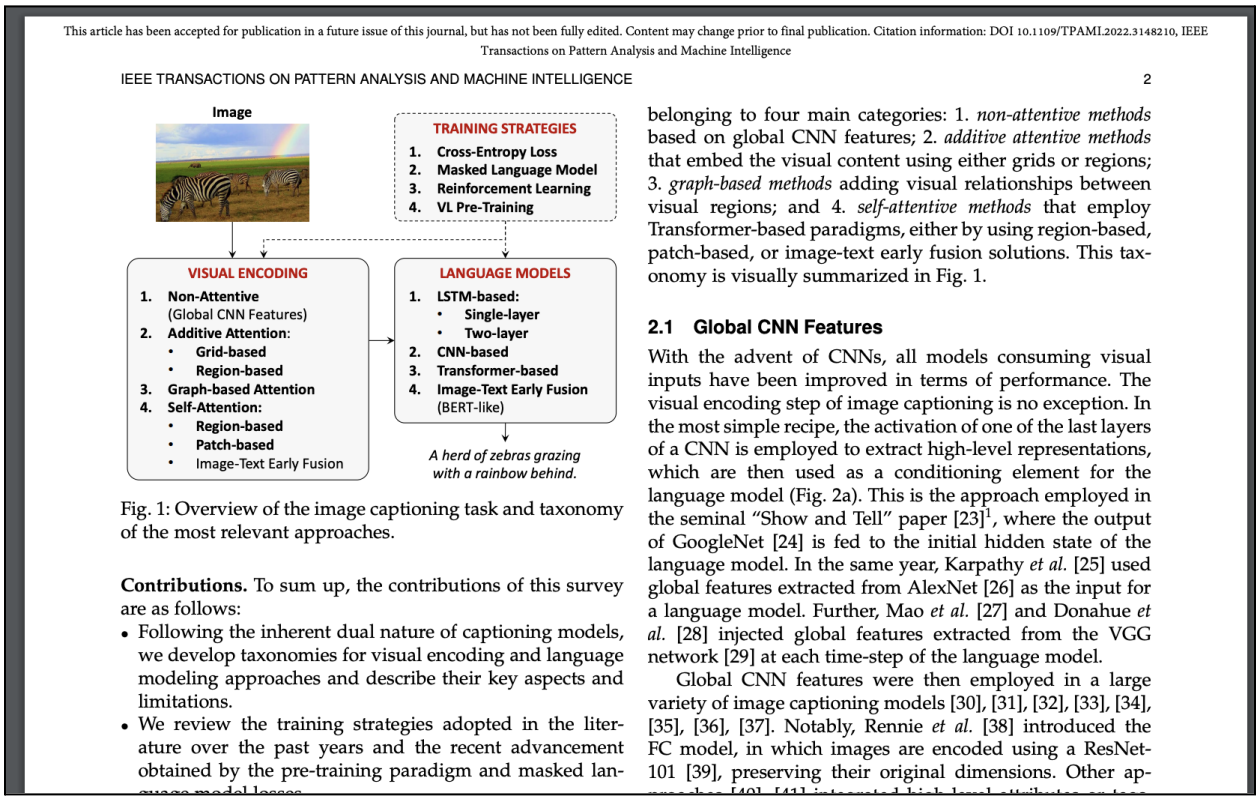
We might suspect that since we've seen some fairly recent papers achieve SOTA (2022), we might not want to look at a 2019 survey. Let's see if we can find a more recent one by explicitly searching for a survey and using the left timeline sidebar to filter results at least as recent as 2021.



Let's hit the first result.



You might find that for survey papers, the abstract does not typically provide the same level of specificity as a typical research article. Survey papers, however, are typically more accessible (at least in some parts) because they include more background about a topic. Let's open this one up.



This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3148210, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 1: Overview of the image captioning task and taxonomy of the most relevant approaches.

Contributions. To sum up, the contributions of this survey are as follows:

- Following the inherent dual nature of captioning models, we develop taxonomies for visual encoding and language modeling approaches and describe their key aspects and limitations.
- We review the training strategies adopted in the literature over the past years and the recent advancement obtained by the pre-training paradigm and masked language model losses

belonging to four main categories: 1. *non-attentive methods* based on global CNN features; 2. *additive attentive methods* that embed the visual content using either grids or regions; 3. *graph-based methods* adding visual relationships between visual regions; and 4. *self-attentive methods* that employ Transformer-based paradigms, either by using region-based, patch-based, or image-text early fusion solutions. This taxonomy is visually summarized in Fig. 1.

2.1 Global CNN Features

With the advent of CNNs, all models consuming visual inputs have been improved in terms of performance. The visual encoding step of image captioning is no exception. In the most simple recipe, the activation of one of the last layers of a CNN is employed to extract high-level representations, which are then used as a conditioning element for the language model (Fig. 2a). This is the approach employed in the seminal "Show and Tell" paper [23][1], where the output of GoogleNet [24] is fed to the initial hidden state of the language model. In the same year, Karpathy *et al.* [25] used global features extracted from AlexNet [26] as the input for a language model. Further, Mao *et al.* [27] and Donahue *et al.* [28] injected global features extracted from the VGG network [29] at each time-step of the language model.

Global CNN features were then employed in a large variety of image captioning models [30], [31], [32], [33], [34], [35], [36], [37]. Notably, Rennie *et al.* [38] introduced the FC model, in which images are encoded using a ResNet-101 [39], preserving their original dimensions. Other ap-

How should we read through a 15-page 2 column review article? In this stage, where we're reading *wide*, we will be very selective in what we read. For this paper, I read:

- Figure 1 on page 2 of the review. This typically gives a good visual organization of the key point of the review. Organization of section headings in this paper
- The "Contributions" on the second page. I didn't find this super useful in this paper, since it didn't make clear the takeaways. On other papers, I would try to find the takeaways.
- The "Conclusions and Future Directions" on the last page.

These rather short pieces of the paper should be sufficient for now. Take some time on your own to read through these sections and see if you can come up with 8-10 bullet points of notes.

Here are the notes I made using the above sections.

From Show to Tell: A Survey on Deep Learning-based Image Captioning
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9706348

- Divided into visual encoding, training strategies, language models. Attention shows up a lot in visual encoding.
- Variety of visual encoding strategies (pre-training from scratch, using detections, using features from multi-modal models) all appear to be on par in terms of performance.
- Training strategies have a recent advancement obtained by the pre-training paradigm and masked language model losses.
    - Growing size of pre-training models is a concern for equality in the community.
    - Growing dichotomy between early-fusion strategies and the encoder-decoder paradigm is an open issue.
- Specializing in particular domains and generating captions with different styles and aims is still among the main open challenges for image captioning.
    - variants such as novel objects captioning or controllable captioning, or subword-based tokenization techniques might handle issues.
- Fairness and bias need for designing specific evaluation metrics and focusing on the robustness to unwanted correlations.
- Development of scores that do not need reference captions for assessing the performance would be key for a shift towards unsupervised image captioning.

At this point in time, we already have 2 pages worth of notes!

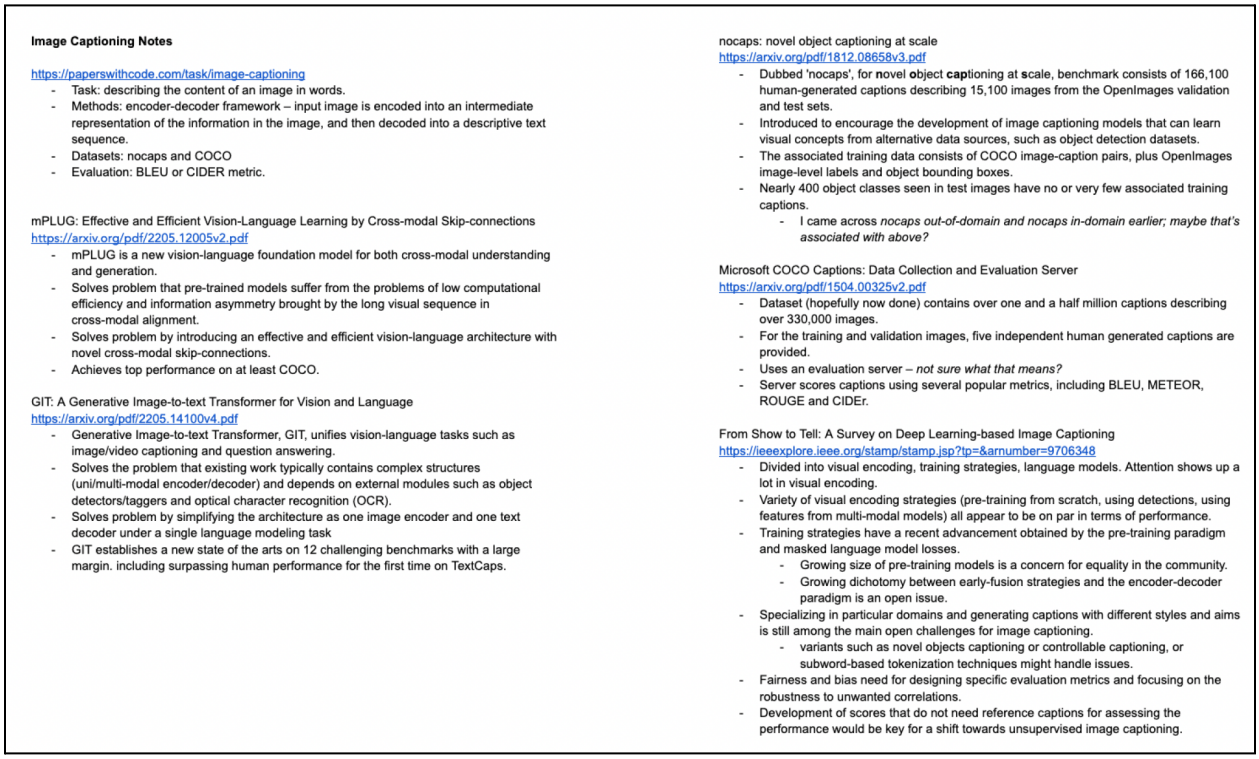


Image Captioning Notes

https://paperswithcode.com/task/image-captioning
- Task: describing the content of an image in words.
- Methods: encoder-decoder framework – input image is encoded into an intermediate representation of the information in the image, and then decoded into a descriptive text sequence.
- Datasets: nocaps and COCO
- Evaluation: BLEU or CIDER metric.

mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections
https://arxiv.org/pdf/2205.12005v2.pdf
- mPLUG is a new vision-language foundation model for both cross-modal understanding and generation.
- Solves problem that pre-trained models suffer from the problems of low computational efficiency and information asymmetry brought by the long visual sequence in cross-modal alignment.
- Solves problem by introducing an effective and efficient vision-language architecture with novel cross-modal skip-connections.
- Achieves top performance on at least COCO.

GIT: A Generative Image-to-text Transformer for Vision and Language
https://arxiv.org/pdf/2205.14100v4.pdf
- Generative Image-to-text Transformer, GIT, unifies vision-language tasks such as image/video captioning and question answering.
- Solves the problem that existing work typically contains complex structures (uni/multi-modal encoder/decoder) and depends on external modules such as object detectors/taggers and optical character recognition (OCR).
- Solves problem by simplifying the architecture as one image encoder and one text decoder under a single language modeling task
- GIT establishes a new state of the arts on 12 challenging benchmarks with a large margin. including surpassing human performance for the first time on TextCaps.

nocaps: novel object captioning at scale
https://arxiv.org/pdf/1812.08658v3.pdf
- Dubbed 'nocaps', for **no**vel **o**bject **cap**tioning at **s**cale, benchmark consists of 166,100 human-generated captions describing 15,100 images from the OpenImages validation and test sets.
- Introduced to encourage the development of image captioning models that can learn visual concepts from alternative data sources, such as object detection datasets.
- The associated training data consists of COCO image-caption pairs, plus OpenImages image-level labels and object bounding boxes.
- Nearly 400 object classes seen in test images have no or very few associated training captions.
    - I came across *nocaps out-of-domain and nocaps in-domain earlier; maybe that's associated with above?*

Microsoft COCO Captions: Data Collection and Evaluation Server
https://arxiv.org/pdf/1504.00325v2.pdf
- Dataset (hopefully now done) contains over one and a half million captions describing over 330,000 images.
- For the training and validation images, five independent human generated captions are provided.
- Uses an evaluation server – *not sure what that means?*
- Server scores captions using several popular metrics, including BLEU, METEOR, ROUGE and CIDEr.

From Show to Tell: A Survey on Deep Learning-based Image Captioning
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9706348
- Divided into visual encoding, training strategies, language models. Attention shows up a lot in visual encoding.
- Variety of visual encoding strategies (pre-training from scratch, using detections, using features from multi-modal models) all appear to be on par in terms of performance.
- Training strategies have a recent advancement obtained by the pre-training paradigm and masked language model losses.
    - Growing size of pre-training models is a concern for equality in the community.
    - Growing dichotomy between early-fusion strategies and the encoder-decoder paradigm is an open issue.
- Specializing in particular domains and generating captions with different styles and aims is still among the main open challenges for image captioning.
    - variants such as novel objects captioning or controllable captioning, or subword-based tokenization techniques might handle issues.
- Fairness and bias need for designing specific evaluation metrics and focusing on the robustness to unwanted correlations.
- Development of scores that do not need reference captions for assessing the performance would be key for a shift towards unsupervised image captioning.

That's great. Your notes, like mine, probably contain many terms you are encountering for the first time… We've seen terms like "encoder-decoder architecture," "language modeling task,"

"cross-modal skip connections," or "subword-based tokenization techniques" that might be beyond our reach of current understanding, but that's okay because when we're reading *wide*.

At this point, we can still put together a fairly neat summary of what we've learnt! Before you read mine, try to compile your learnings – from reading the 1 overview page, 2 methods paper abstracts, 2 datasets paper abstracts, and small sections from 1 survey paper – all into a summary paragraph.

Learnings

- The task of image captioning is to describe the content of an image in words. Popular datasets to tackle this question include nocaps and coco, which each contain hundreds of thousands of images. Recent SOTA methods for image captioning include mPLUG and GIT. mPLUG proposes an effective and efficient vision-language architecture and GIT simplifies typically complex structures into one image encoder and one text decoder under a single language modeling task. Specializing image captioning for particular domains and generating captions with different styles and aims is possibly a big open challenge.

This is great!

Between Wide and Deep

You may have noticed that your notes are likely already diverging from mine, especially towards the last paper! Even if we have read the same content, you are starting to develop your own mental model of the problem space, and your interest will be piqued by a different set of words. That's good – you're developing a taste for the kind of research questions you will find interesting.

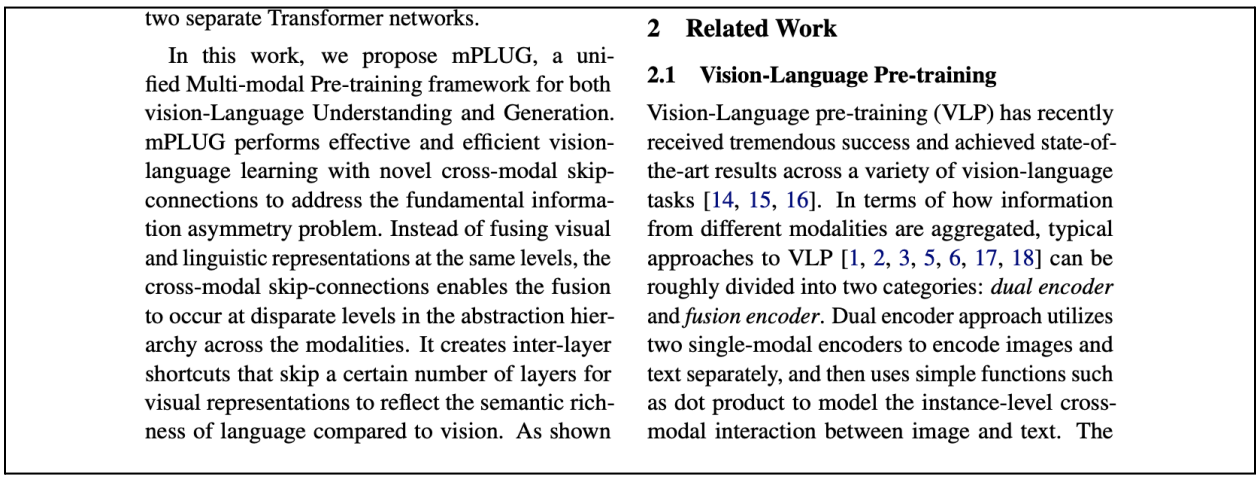
At this point, I find myself particularly intrigued by the SOTA methods: Why are they achieving high performance? According to the review paper, it looks like "training strategies using pre-training" have been an advance. Maybe that's worth keeping an eye out for!

Related Work

At this stage, I'll find it effective to read the related works sections of these papers: they often make it clear how researchers in the field have traditionally approached the problems and what the emerging trends are. It's important to pick papers that are recently published.

Let's dive in: both mPLUG (https://arxiv.org/pdf/2205.12005v2.pdf) and GIT (https://arxiv.org/pdf/2205.14100v4.pdf) are recently published methods that achieve SOTA.

The mPLUG related work starts towards the end of page 2. Let's read the related works subsection (I've only included one paragraph). I will focus particularly on the vision-language pre-training subsection.



two separate Transformer networks.

In this work, we propose mPLUG, a unified Multi-modal Pre-training framework for both vision-Language Understanding and Generation. mPLUG performs effective and efficient vision-language learning with novel cross-modal skip-connections to address the fundamental information asymmetry problem. Instead of fusing visual and linguistic representations at the same levels, the cross-modal skip-connections enables the fusion to occur at disparate levels in the abstraction hierarchy across the modalities. It creates inter-layer shortcuts that skip a certain number of layers for visual representations to reflect the semantic richness of language compared to vision. As shown

2 Related Work

2.1 Vision-Language Pre-training

Vision-Language pre-training (VLP) has recently received tremendous success and achieved state-of-the-art results across a variety of vision-language tasks [14, 15, 16]. In terms of how information from different modalities are aggregated, typical approaches to VLP [1, 2, 3, 5, 6, 17, 18] can be roughly divided into two categories: *dual encoder* and *fusion encoder*. Dual encoder approach utilizes two single-modal encoders to encode images and text separately, and then uses simple functions such as dot product to model the instance-level cross-modal interaction between image and text. The

Now, attempt to update your previous notes of mPLUG with your understanding of this related work section. Here are my own notes, unfiltered: I've used shorthand and left in spelling/grammar errors :)

mPLUG: Effective and Efficient Vision-Language Learning by Cross-modal Skip-connections
https://arxiv.org/pdf/2205.12005v2.pdf

- mPLUG is a new vision-language foundation model for both cross-modal understanding and generation.
- Solves problem that pre-trained models suffer from the problems of low computational efficiency and information asymmetry brought by the long visual sequence in cross-modal alignment.
- Solves problem by introducing an effective and efficient vision-language architecture with novel cross-modal skip-connections.
- Achieves top performance on at least COCO.
- Related work:
    - For vision language pre training, there are two approaches to how info from different modalities is aggregate: dual encoder, and fusion.
        - Dual encoder uses 2 encoders, one vision, one text, and simple functions to model interaction between image and text.
            - Plus: computationally efficient.
            - Minus: can't handle complicated vision-language tasks.
            - Examples: CLIP, ALIGN
        - Fusion encoders uses attention to model interaction between image and text.
            - Plus: better capture association between image and text for vision-language understanding tasks.
            - Minus: inference is slow, with exceptions e.x. PixelBERT.
            - Examples: Single-stream: UNITER, two-stream: LXMERT.
        - mPLUG enables the fusion to occur at disparate levels in the abstraction hierarchy across the modalities.

The examples we have collected will serve to populate our reading list when we start to read through individual papers.

Exercise: Repeat this process, this time for the GIT paper.

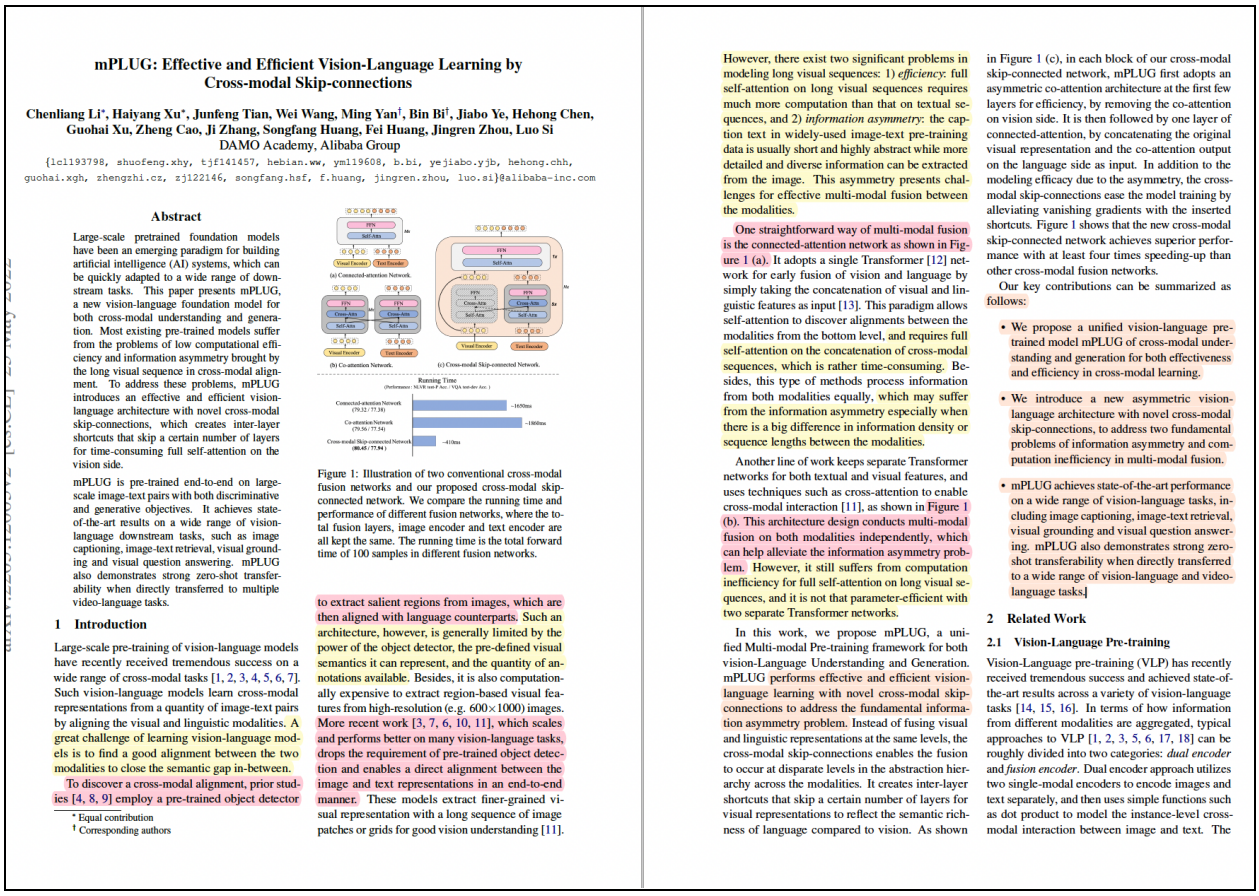
Reading Deep

We've identified a few key works for our topic in the first stage and gotten a mental model of the space of image captioning. We're now going to compliment wide reading with deep reading, covering the way to read an individual paper. Most papers are written for an audience that shares a common foundation: that's what allows for the papers to be relatively concise. Building that foundation takes time, in the span of months, if not years. Thus reading a first paper on a topic can easily take over 10 hours (some papers have definitely taken me 20 or 30 hours) and leave one feeling overwhelmed.

So I would like you to take an incremental approach here. Understand that, in your first pass, you will not understand more than 10% of the research paper. The paper may require us to read another more fundamental paper (which might require reading a third paper and so on; it could be turtles all the way down)! Then, in your second pass, you might understand 20% of the paper. Understanding 100% of a paper might require a significant leap, maybe because it's poorly written, insufficiently detailed, or simply too technically/mathematically advanced. We thus want to aim to build up to understanding as much of the paper as possible – I'll bet that 70-80% of the paper is a good target.

We will go through the mPLUG paper, and I'll walk you through my first read of it. It's a good idea to be able to highlight papers as you go through them: or make comments. You can use Adobe or Preview (on Mac) to highlight PDFs (or a web-based solution like https://hypothes.is/ for the annotation).

I'm going to read the Introduction section first. In a later lecture, I will share with you how to write good introductions. Introductions are a good way to start a paper because they are typically written for a more general audience than the rest of the paper.



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

Chenliang Li*, Haiyang Xu*, Junfeng Tian, Wei Wang, Ming Yan†, Bin Bi†, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, Luo Si
DAMO Academy, Alibaba Group
{lcl193798, shuofeng.xhy, tjf141457, hebian.ww, ym119608, b.bi, yejiabo.yjb, hehong.chh, guohai.xgh, zhengzhi.cz, zj122146, songfang.hsf, f.huang, jingren.zhou, luo.si}@alibaba-inc.com

Abstract

Large-scale pretrained foundation models have been an emerging paradigm for building artificial intelligence (AI) systems, which can be quickly adapted to a wide range of downstream tasks. This paper presents mPLUG, a new vision-language foundation model for both cross-modal understanding and generation. Most existing pre-trained models suffer from the problems of low computational efficiency and information asymmetry brought by the long visual sequence in cross-modal alignment. To address these problems, mPLUG introduces an effective and efficient vision-language architecture with novel cross-modal skip-connections, which creates inter-layer shortcuts that skip a certain number of layers for time-consuming full self-attention on the vision side.

mPLUG is pre-trained end-to-end on large-scale image-text pairs with both discriminative and generative objectives. It achieves state-of-the-art results on a wide range of vision-language downstream tasks, such as image captioning, image-text retrieval, visual grounding and visual question answering. mPLUG also demonstrates strong zero-shot transferability when directly transferred to multiple video-language tasks.

1 Introduction

Large-scale pre-training of vision-language models have recently received tremendous success on a wide range of cross-modal tasks [1, 2, 3, 4, 5, 6, 7]. Such vision-language models learn cross-modal representations from a quantity of image-text pairs by aligning the visual and linguistic modalities. A great challenge of learning vision-language models is to find a good alignment between the two modalities to close the semantic gap in-between.

To discover a cross-modal alignment, prior studies [4, 8, 9] employ a pre-trained object detector to extract salient regions from images, which are then aligned with language counterparts. Such an architecture, however, is generally limited by the power of the object detector, the pre-defined visual semantics it can represent, and the quantity of annotations available. Besides, it is also computationally expensive to extract region-based visual features from high-resolution (e.g. 600×1000) images. More recent work [3, 7, 6, 10, 11], which scales and performs better on many vision-language tasks, drops the requirement of pre-trained object detection and enables a direct alignment between the image and text representations in an end-to-end manner. These models extract finer-grained visual representation with a long sequence of image patches or grids for good vision understanding [11].

* Equal contribution
† Corresponding authors

Figure 1: Illustration of two conventional cross-modal fusion networks and our proposed cross-modal skip-connected network. We compare the running time and performance of different fusion networks, where the total fusion layers, image encoder and text encoder are all kept the same. The running time is the total forward time of 100 samples in different fusion networks.

However, there exist two significant problems in modeling long visual sequences: 1) *efficiency*: full self-attention on long visual sequences requires much more computation than that on textual sequences, and 2) *information asymmetry*: the caption text in widely-used image-text pre-training data is usually short and highly abstract while more detailed and diverse information can be extracted from the image. This asymmetry presents challenges for effective multi-modal fusion between the modalities.

One straightforward way of multi-modal fusion is the connected-attention network as shown in Figure 1 (a). It adopts a single Transformer [12] network for early fusion of vision and language by simply taking the concatenation of visual and linguistic features as input [13]. This paradigm allows self-attention to discover alignments between the modalities from the bottom level, and requires full self-attention on the concatenation of cross-modal sequences, which is rather time-consuming. Besides, this type of methods process information from both modalities equally, which may suffer from the information asymmetry especially when there is a big difference in information density or sequence lengths between the modalities.

Another line of work keeps separate Transformer networks for both textual and visual features, and uses techniques such as cross-attention to enable cross-modal interaction [11], as shown in Figure 1 (b). This architecture design conducts multi-modal fusion on both modalities independently, which can help alleviate the information asymmetry problem. However, it still suffers from computation inefficiency for full self-attention on long visual sequences, and it is not that parameter-efficient with two separate Transformer networks.

In this work, we propose mPLUG, a unified Multi-modal Pre-training framework for both vision-Language Understanding and Generation. mPLUG performs effective and efficient vision-language learning with novel cross-modal skip-connections to address the fundamental information asymmetry problem. Instead of fusing visual and linguistic representations at the same levels, the cross-modal skip-connections enables the fusion to occur at disparate levels in the abstraction hierarchy across the modalities. It creates inter-layer shortcuts that skip a certain number of layers for visual representations to reflect the semantic richness of language compared to vision. As shown in Figure 1 (c), in each block of our cross-modal skip-connected network, mPLUG first adopts an asymmetric co-attention architecture at the first few layers for efficiency, by removing the co-attention on vision side. It is then followed by one layer of connected-attention, by concatenating the original visual representation and the co-attention output on the language side as input. In addition to the modeling efficacy due to the asymmetry, the cross-modal skip-connections ease the model training by alleviating vanishing gradients with the inserted shortcuts. Figure 1 shows that the new cross-modal skip-connected network achieves superior performance with at least four times speeding-up than other cross-modal fusion networks.

Our key contributions can be summarized as follows:

- We propose a unified vision-language pre-trained model mPLUG of cross-modal understanding and generation for both effectiveness and efficiency in cross-modal learning.
- We introduce a new asymmetric vision-language architecture with novel cross-modal skip-connections, to address two fundamental problems of information asymmetry and computation inefficiency in multi-modal fusion.
- mPLUG achieves state-of-the-art performance on a wide range of vision-language tasks, including image captioning, image-text retrieval, visual grounding and visual question answering. mPLUG also demonstrates strong zero-shot transferability when directly transferred to a wide range of vision-language and video-language tasks.

2 Related Work

2.1 Vision-Language Pre-training

Vision-Language pre-training (VLP) has recently received tremendous success and achieved state-of-the-art results across a variety of vision-language tasks [14, 15, 16]. In terms of how information from different modalities are aggregated, typical approaches to VLP [1, 2, 3, 5, 6, 17, 18] can be roughly divided into two categories: *dual encoder* and *fusion encoder*. Dual encoder approach utilizes two single-modal encoders to encode images and text separately, and then uses simple functions such as dot product to model the instance-level cross-modal interaction between image and text. The

I have highlighted what I found to be important parts of the introduction. The yellow highlights are the problems/challenges, the pink highlights are the solutions to the challenges, and the orange highlights are the main contributions of the work we're reading.

Notice the alternating yellow/pink highlights. The paper is introducing a general problem, talking about a solution to that problem, then a problem with the solution, and another solution to that problem. Four levels deep, the paper specifies the problem it solves.

Notice then that the contribution of the paper is a specific solution for a specific problem of a more general solution for a more general problem of an even more general solution to an even more general problem etc. This is typical. We can summarize our understanding of this problem-solution chain:

- Introduction:
  - Problem 1: How to find alignment between image and text modalities?
  - Solution 1: Pre-trained object detectors to find salient regions from images.
  - Problem 2: Limited by power of object detector & available annotations.
  - Solution 2: Direct alignment without object detectors
  - Problem 2a: Efficiency because of lot of computation of self-attention on visual sequences
  - Problem 2b: information asymmetry because text is short compared to info in image.
  - Solution 2a: connected attention network, using single transformer for early fusion. Has problems 2a and 2b.
  - Solution 2b: cross attention network, does fusion on both modalities independently. No longer has problem 2b, but still has 2a.
  - Solution 3 (proposed solution): cross-modal skip connections. Solves problem 2a and 2b.

This is neat – we've built a mental model of how the different pieces fit in. A well written introduction should allow you to extract such a problem–solution chain, but not every paper will make this explicit. We can often refer to the figure 1 to understand the main idea of the paper better:

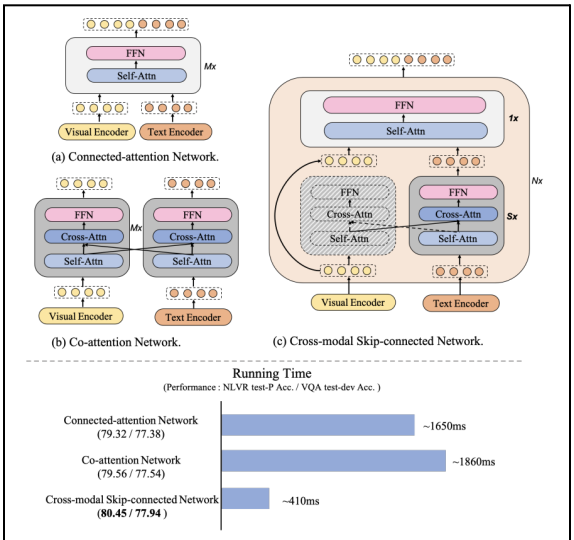


Figure 1: Illustration of two conventional cross-modal fusion networks and our proposed cross-modal skip-connected network. We compare the running time and performance of different fusion networks, where the total fusion layers, image encoder and text encoder are all kept the same. The running time is the total forward time of 100 samples in different fusion networks.

We can see the proposed solution (c), and the comparison to solutions in (a) and (b), which correspond to solutions 2a and 2b in our notes.

What's next? We've already read the abstract previously, we've read the introduction, we've seen Figure 1. On this paper, we've also had the opportunity to read the related work.

Now we read the methods, right?

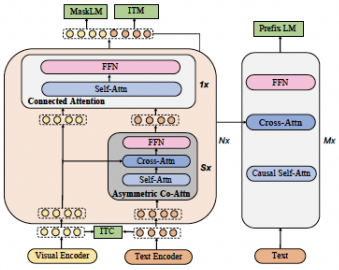
the transformed input. DenseNet [23] designs new architectures with concatenated skip-connections, allowing the subsequent layers to re-use all the middle representations of previous layers. Layer Normalization and recursive skip connection are further used in combination with plain skip connection for further stablizing model optimization and better incorporating the transformed input [12, 25]. In this work, mPLUG proposes a new cross-modal skip connection method to address cross-modal fusion problem, and combines the concatenated skip-connection and summed skip-connection for choosing whether to attend to all the concatenated representations of different modalities or just focus on the cross-modal interaction part at each layer.

3 mPLUG

In this section, we will first introduce our new model architecture with the key module of the cross-modal skip-connected network, and then give the details of the pre-training objectives and scalable training infrastructure.

3.1 Model Architecture

As shown in Figure 2, mPLUG consists of two unimodal encoders for image and text independently, a cross-modal skip-connected network and a decoder for text generation. To better model the inherent modality bias information, we first use two unimodal encoders to encode image and text separately. Following [11, 26], we use a visual transformer [27] directly on the image patches as the visual encoder, which is more computation-friendly than using pre-trained object detectors for visual feature extraction [8, 9]. The visual encoder divides an input image into patches and encodes them as a sequence of embeddings $\{v_{cls}, v_1, v_2, ..., v_M\}$ with an additional $[CLS]$ token. The input text is fed to the text encoder and represented as a sequence of embeddings $\{l_{cls}, l_1, l_2, ..., l_N\}$, where l_{cls} is the embedding of the $[CLS]$ token and used to summarize the input text. Then, the visual and linguistic representations are fed into a cross-modal skip-connected network, which consists of multiple skip-connected fusion blocks. In each skip-connected fusion block, we adopt connected cross-modal fusion to each of S *asymmetric co-attention* layers where S is a fixed stride value. The aim of this network is to take advantage of the effectiveness of the connected cross-modal fusion and the efficiency of the asymmetric co-attention for



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)
```

Figure 2: The model architecture and objectives of mPLUG, which consists of two unimodal encoders for images and text separately, a cross-modal skip-connected network and a decoder for text generation. An image-text contrastive loss is first applied to align the unimodal representations from the visual encoder and text encoder. Then, we use a novel cross-modal skip-connected network to fuse the visual and linguistic representations effectively and efficiently. We adopt connected cross-modal fusion to every S *asymmetric co-attention* layers, where S is a fixed stride value. Based on the connected representation of the image and prefix sub-sequence, the decoder is trained with a prefix language modeling (Prefix LM) loss by generating the remaining caption.

enhanced cross-modal fusion in a recursive manner. Finally, the output cross-modal representations are fed into a transformer decoder for sequence to sequence learning, which equips mPLUG with both understanding and generation capabilities.

3.2 Cross-modal Skip-connected Network

The cross-modal skip-connected network consists of N skip-connected fusion blocks. In each skip-connected fusion block, we adopt *connected-attention* layer to each of S *asymmetric co-attention* layers where S is a fixed stride value. We first pass the text feature and image feature from unimodal encoders through the S asymmetric co-attention layers, and then connect the output text feature and image feature to one connected-attention layer. We repeat the skip-connected fusion block N times for the final connected image and text representation.

Specifically, the asymmetric co-attention is composed of the self-attention (SA) layer, cross-attention (CA) layer and the feed-forward network (FFN). The input text feature l^{n-1} is first fed to the self-attention layer, and then the visual feature v^{n-1} is injected into the text feature l^n_{SA} by the cross-attention layer which gives l^n_{CA}. The output of self-attention l^n_{SA} and cross-attention l^n_{SA} are added up and fed to the FFN layer for the visual-aware text representation l^n:

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

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

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

where LN is short for layer normalization.

The connected-attention layer is composed of the self-attention (SA) layer and the feed-forward network (FFN). We connect the image feature v^{n-1} and input text feature l^{n-1}, where l^{n-1} is the output of S asymmetric co-attention layers. The connected image and text feature $[v^{n-1}; l^{n-1}]$ are fed to the self-attention layer and FFN layer:

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

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

Then $[v^n; l^n]$ is fed into the next cross-modal skip-connected network repeatedly to get the final connected image and text representation. Finally, the connected output is fed into a Transformer decoder for sequence to sequence learning.

As you can see, I have made no highlights on sections 3.1, and 3.2. We're missing the context of 10-20 papers that we will need to read to fill in the gaps. That's okay: this is the 20-30% of the paper we are not going to have an in-depth understanding of. We can still squeeze out some understanding from the methods section using Figure 2, which presents a simpler description of the architecture and objectives of mPlug. We also have python pseudocode that actually makes it easier to work through the operations and flow of the model. In general, well constructed figures and algorithm pseudocode can be of huge help to us!

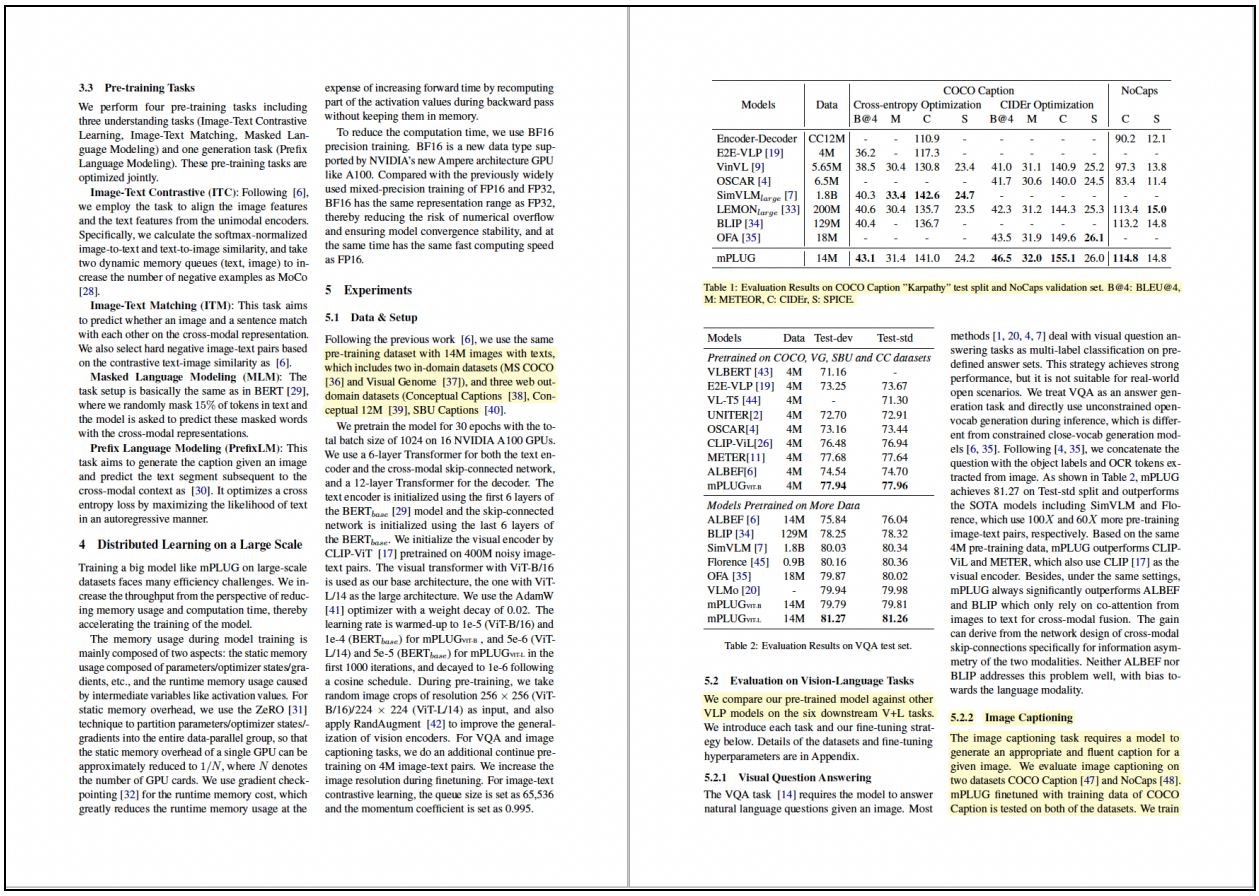
What we can do for the methods section is maintain a list of concepts that we haven't quite understood: if there's a link to the paper references, we'll copy it over. Here's what that might look like:

To Understand:

- From mPLUG:
    - Self-attention + Cross-attention?
        - Detail: Layer normalization?
    - Image-Text Contrastive (ITC): follows "Align before fuse: Vision and language representation learning with momentum distillation"
    - Prefix Language Modeling (PrefixLM): task follows Palm: Pre-training an autoencoding & autoregressive language model for context-conditioned generation.
    - From related work, "Vlmo: Unified vision-language pretraining with mixture-of-modality-experts" using dual encoder and fusion encoder modules.
    - Cross modal interaction example: "An empirical study of training end-to-end vision-and-language transformers."

You would have thus created a list of concepts you need to learn about, and the relevant paper for each, if the paper specifies any.

Let's continue reading, making our way through the methods sections, and the experiments section, highlighting the parts that we consider relevant to our understanding of the method as it relates to image captioning.



3.3 Pre-training Tasks

We perform four pre-training tasks including three understanding tasks (Image-Text Contrastive Learning, Image-Text Matching, Masked Language Modeling) and one generation task (Prefix Language Modeling). These pre-training tasks are optimized jointly.

Image-Text Contrastive (ITC): Following [6], we employ the task to align the image features and the text features from the unimodal encoders. Specifically, we calculate the softmax-normalized image-to-text and text-to-image similarity, and take two dynamic memory queues (text, image) to increase the number of negative examples as MoCo [28].

Image-Text Matching (ITM): This task aims to predict whether an image and a sentence match with each other on the cross-modal representation. We also select hard negative image-text pairs based on the contrastive text-image similarity as [6].

Masked Language Modeling (MLM): The task setup is basically the same as in BERT [29], where we randomly mask 15% of tokens in text and the model is asked to predict these masked words with the cross-modal representations.

Prefix Language Modeling (PrefixLM): This task aims to generate the caption given an image and predict the text segment subsequent to the cross-modal context as [30]. It optimizes a cross entropy loss by maximizing the likelihood of text in an autoregressive manner.

4 Distributed Learning on a Large Scale

Training a big model like mPLUG on large-scale datasets faces many efficiency challenges. We increase the throughput from the perspective of reducing memory usage and computation time, thereby accelerating the training of the model.

The memory usage during model training is mainly composed of two aspects: the static memory usage composed of parameters/optimizer states/gradients, etc., and the runtime memory usage caused by intermediate variables like activation values. For static memory overhead, we use the ZeRO [31] technique to partition parameters/optimizer states/gradients into the entire data-parallel group, so that the static memory overhead of a single GPU can be approximately reduced to $1/N$, where N denotes the number of GPU cards. We use gradient checkpointing [32] for the runtime memory cost, which greatly reduces the runtime memory usage at the expense of increasing forward time by recomputing part of the activation values during backward pass without keeping them in memory.

To reduce the computation time, we use BF16 precision training. BF16 is a new data type supported by NVIDIA's new Ampere architecture GPU like A100. Compared with the previously widely used mixed-precision training of FP16 and FP32, BF16 has the same representation range as FP32, thereby reducing the risk of numerical overflow and ensuring model convergence stability, and at the same time has the same fast computing speed as FP16.

5 Experiments

5.1 Data & Setup

Following the previous work [6], we use the same pre-training dataset with 14M images with texts, which includes two in-domain datasets (MS COCO [36] and Visual Genome [37]), and three web out-domain datasets (Conceptual Captions [38], Conceptual 12M [39], SBU Captions [40].

We pretrain the model for 30 epochs with the total batch size of 1024 on 16 NVIDIA A100 GPUs. We use a 6-layer Transformer for both the text encoder and the cross-modal skip-connected network, and a 12-layer Transformer for the decoder. The text encoder is initialized using the first 6 layers of the BERT_{base} [29] model and the skip-connected network is initialized using the last 6 layers of the BERT_{base}. We initialize the visual encoder by CLIP-ViT [17] pretrained on 400M noisy image-text pairs. The visual transformer with ViT-B/16 is used as our base architecture, the one with ViT-L/14 as the large architecture. We use the AdamW [41] optimizer with a weight decay of 0.02. The learning rate is warmed-up to 1e-5 (ViT-B/16) and 1e-4 (BERT_{base}) for $\text{mPLUG}_{\text{VIT-B}}$, and 5e-6 (ViT-L/14) and 5e-5 (BERT_{base}) for $\text{mPLUG}_{\text{VIT-L}}$ in the first 1000 iterations, and decayed to 1e-6 following a cosine schedule. During pre-training, we take random image crops of resolution 256×256 (ViT-B/16)/224×224 (ViT-L/14) as input, and also apply RandAugment [42] to improve the generalization of vision encoders. For VQA and image captioning tasks, we do an additional continue pre-training on 4M image-text pairs. We increase the image resolution during finetuning. For image-text contrastive learning, the queue size is set as 65,536 and the momentum coefficient is set as 0.995.

Models	Data	COCO Caption								NoCaps	
		Cross-entropy Optimization				CIDEr Optimization					
		B@4	M	C	S	B@4	M	C	S	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.

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**

Table 2: Evaluation Results on VQA test set.

5.2 Evaluation on Vision-Language Tasks

We compare our pre-trained model against other VLP models on the six downstream V+L tasks. We introduce each task and our fine-tuning strategy below. Details of the datasets and fine-tuning hyperparameters are in Appendix.

5.2.1 Visual Question Answering

The VQA task [14] requires the model to answer natural language questions given an image. Most methods [1, 20, 4, 7] deal with visual question answering tasks as multi-label classification on predefined answer sets. This strategy achieves strong performance, but it is not suitable for real-world open scenarios. We treat VQA as an answer generation task and directly use unconstrained open-vocab generation during inference, which is different from constrained close-vocab generation models [6, 35]. Following [4, 35], we concatenate the question with the object labels and OCR tokens extracted from image. As shown in Table 2, mPLUG achieves 81.27 on Test-std split and outperforms the SOTA models including SimVLM and Florence, which use 100.X and 60.X more pre-training image-text pairs, respectively. Based on the same 4M pre-training data, mPLUG outperforms CLIP-ViL and METER, which also use CLIP [17] as the visual encoder. Besides, under the same settings, mPLUG always significantly outperforms ALBEF and BLIP which only rely on co-attention from images to text for cross-modal fusion. The gain can derive from the network design of cross-modal skip-connections specifically for information asymmetry of the two modalities. Neither ALBEF nor BLIP addresses this problem well, with bias towards the language modality.

5.2.2 Image Captioning

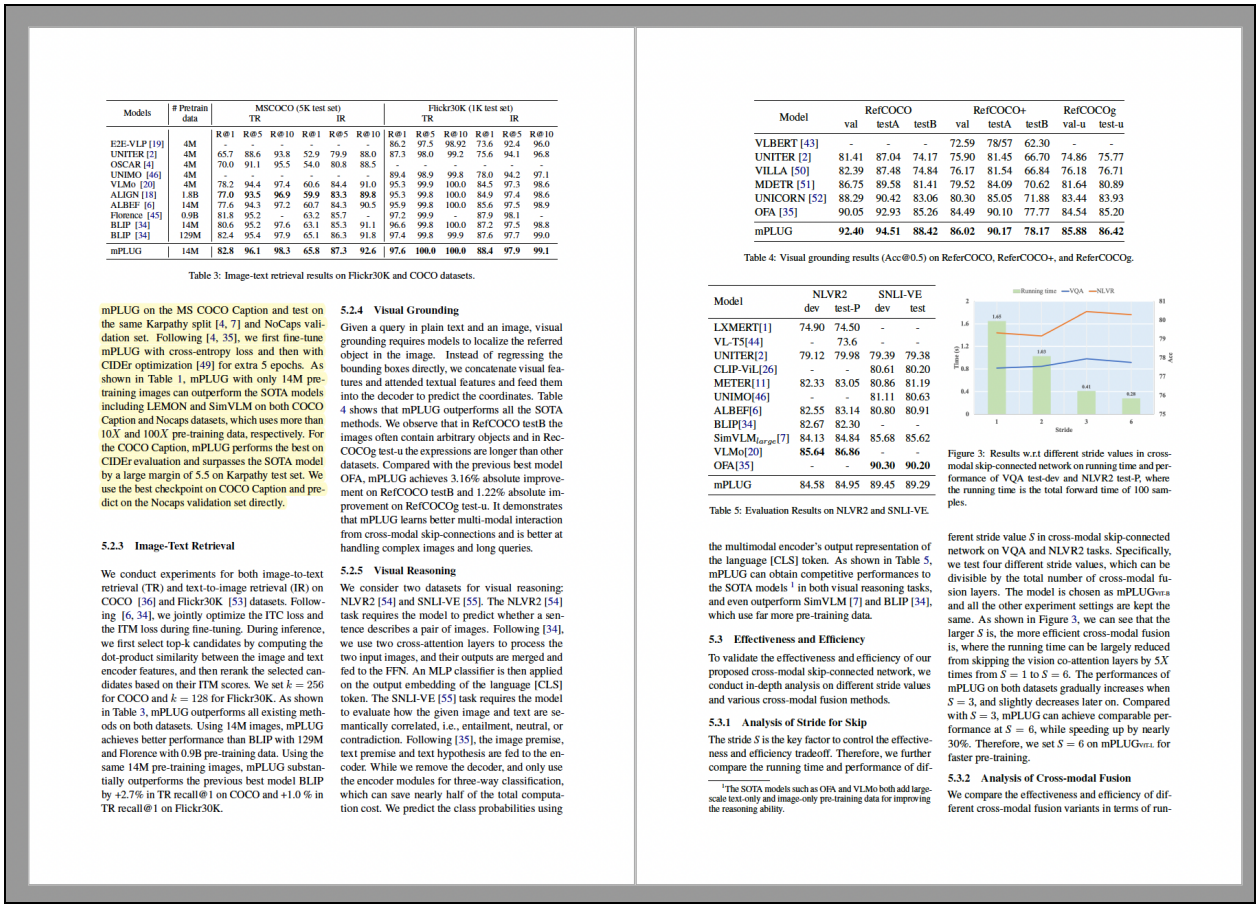
The image captioning task requires a model to generate an appropriate and fluent caption for a given image. We evaluate image captioning on two datasets COCO Caption [47] and NoCaps [48]. mPLUG finetuned with training data of COCO Caption is tested on both of the datasets. We train

Here, I have highlighted the data setup, the main table of results (Table 1) corresponding to image captioning, and the image captioning section. Table 1 includes both methods and evaluation metrics that may be unfamiliar to us, and we can make not of that in our "To Understand" list.

To Understand:

- From mPLUG:
  - Methods
    - Self-attention + Cross-attention?
      - Detail: Layer normalization?
    - Image-Text Contrastive (ITC): follows “Align before fuse: Vision and language representation learning with momentum distillation”
    - Prefix Language Modeling (PrefixLM): task follows Palm: Pre-training an autoencoding & autoregressive language model for context-conditioned generation.
    - From related work, “Vlmo: Unified vision-language pretraining with mixture-of-modality-experts” using dual encoder and fusion encoder modules.
    - Cross modal interaction example: “An empirical study of training end-to-end vision-and-language transformers.”
  - Experiments:
    - What are BLEU-4, METEOR, CIDEr, SPICE?
    - Method comparisons: SimVLM is a competitive method. Read “Simvlm: Simple visual language model pretraining with weak supervision”

We continue reading through the rest of the experiments section



Models	# Pretrain data	MSCOCO (5K test set) TR			IR			Flickr30K (1K test set) TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	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.

mPLUG on the MS COCO Caption and test on the same Karpathy split [4, 7] and NoCaps validation set. Following [4, 35], we first fine-tune mPLUG with cross-entropy loss and then with CIDEr optimization [49] for extra 5 epochs. As shown in Table 1, mPLUG with only 14M pre-training images can outperform the SOTA models including LEMON and SimVLM on both COCO Caption and Nocaps datasets, which uses more than $10X$ and $100X$ pre-training data, respectively. For the COCO Caption, mPLUG performs the best on CIDEr evaluation and surpasses the SOTA model by a large margin of 5.5 on Karpathy test set. We use the best checkpoint on COCO Caption and predict on the Nocaps validation set directly.

5.2.3 Image-Text Retrieval

We conduct experiments for both image-to-text retrieval (TR) and text-to-image retrieval (IR) on COCO [36] and Flickr30K [53] datasets. Following [6, 34], we jointly optimize the ITC loss and the ITM loss during fine-tuning. During inference, we first select top-k candidates by computing the dot-product similarity between the image and text encoder features, and then rerank the selected candidates based on their ITM scores. We set $k = 256$ for COCO and $k = 128$ for Flickr30K. As shown in Table 3, mPLUG outperforms all existing methods on both datasets. Using 14M images, mPLUG achieves better performance than BLIP with 129M and Florence with 0.9B pre-training data. Using the same 14M pre-training images, mPLUG substantially outperforms the previous best model BLIP by +2.7% in TR recall@1 on COCO and +1.0 % in TR recall@1 on Flickr30K.

5.2.4 Visual Grounding

Given a query in plain text and an image, visual grounding requires models to localize the referred object in the image. Instead of regressing the bounding boxes directly, we concatenate visual features and attended textual features and feed them into the decoder to predict the coordinates. Table 4 shows that mPLUG outperforms all the SOTA methods. We observe that in RefCOCO testB the images often contain arbitrary objects and in RecCOCOg test-u the expressions are longer than other datasets. Compared with the previous best model OFA, mPLUG achieves 3.16% absolute improvement on RefCOCO testB and 1.22% absolute improvement on RefCOCOg test-u. It demonstrates that mPLUG learns better multi-modal interaction from cross-modal skip-connections and is better at handling complex images and long queries.

5.2.5 Visual Reasoning

We consider two datasets for visual reasoning: NLVR2 [54] and SNLI-VE [55]. The NLVR2 [54] task requires the model to predict whether a sentence describes a pair of images. Following [34], we use two cross-attention layers to process the two input images, and their outputs are merged and fed to the FFN. An MLP classifier is then applied on the output embedding of the language [CLS] token. The SNLI-VE [55] task requires the model to evaluate how the given image and text are semantically correlated, i.e., entailment, neutral, or contradiction. Following [35], the image premise, text premise and text hypothesis are fed to the encoder. While we remove the decoder, and only use the encoder modules for three-way classification, which can save nearly half of the total computation cost. We predict the class probabilities using

Model	RefCOCO val	RefCOCO testA	RefCOCO testB	RefCOCO+ val	RefCOCO+ testA	RefCOCO+ testB	RefCOCOg val-u	RefCOCOg 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.

Model	NLVR2 dev	NLVR2 test-P	SNLI-VE dev	SNLI-VE test
LXMERT[1]	74.90	74.50	-	-
VL-T5[44]	-	73.6	-	-
UNITER[2]	79.12	79.98	79.39	79.38
CLIP-ViL[26]	-	-	80.61	80.20
METER[11]	82.33	83.05	80.86	81.19
UNIMO[46]	-	-	81.11	80.63
ALBEF[6]	82.55	83.14	80.80	80.91
BLIP[34]	82.67	82.30	-	-
SimVLM$_{large}$[7]	84.13	84.84	85.68	85.62
VLMo[20]	**85.64**	**86.86**	-	-
OFA[35]	-	-	**90.30**	**90.20**
mPLUG	84.58	84.95	89.45	89.29

Table 5: Evaluation Results on NLVR2 and SNLI-VE.

Figure 3: Results w.r.t different stride values in cross-modal skip-connected network on running time and performance of VQA test-dev and NLVR2 test-P, where the running time is the total forward time of 100 samples.

the multimodal encoder's output representation of the language [CLS] token. As shown in Table 5, mPLUG can obtain competitive performances to the SOTA models [1] in both visual reasoning tasks, and even outperform SimVLM [7] and BLIP [34], which use far more pre-training data.

5.3 Effectiveness and Efficiency

To validate the effectiveness and efficiency of our proposed cross-modal skip-connected network, we conduct in-depth analysis on different stride values and various cross-modal fusion methods.

5.3.1 Analysis of Stride for Skip

The stride S is the key factor to control the effectiveness and efficiency tradeoff. Therefore, we further compare the running time and performance of different stride value S in cross-modal skip-connected network on VQA and NLVR2 tasks. Specifically, we test four different stride values, which can be divisible by the total number of cross-modal fusion layers. The model is chosen as mPLUG$_{\text{VIT-B}}$ and all the other experiment settings are kept the same. As shown in Figure 3, we can see that the larger S is, the more efficient cross-modal fusion is, where the running time can be largely reduced from skipping the vision co-attention layers by $5X$ times from $S = 1$ to $S = 6$. The performances of mPLUG on both datasets gradually increases when $S = 3$, and slightly decreases later on. Compared with $S = 3$, mPLUG can achieve comparable performance at $S = 6$, while speeding up by nearly 30%. Therefore, we set $S = 6$ on mPLUG$_{\text{VIT-L}}$ for faster pre-training.

5.3.2 Analysis of Cross-modal Fusion

We compare the effectiveness and efficiency of different cross-modal fusion variants in terms of run-

[1]The SOTA models such as OFA and VLMo both add large-scale text-only and image-only pre-training data for improving the reasoning ability.

While we can read through the results on tasks other than "image captioning," I have ignored highlighting them. When reading papers, we are often being presented with a firehose of information, and it's useful to be selective in what we pay attention to. Let's continue reading.

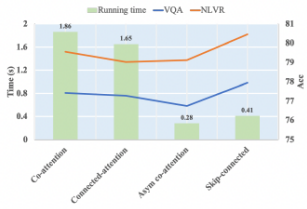


Figure 4: Results w.r.t different cross-modal fusions on running time and performance on VQA test-dev and NLVR2 test-P, where the running time is the total forward time of 100 samples.

Model	Throughput (Samples/S)
baseline	124.0
+ BFloat16	182.7
+ Gradient Checkpoint	238.2
+ ZeRO	**422.5**

Table 6: Training Throughput

ning time and performance on VQA and NLVR2 tasks. Specifically, we pre-train mPLUG with different cross-modal fusion network based on the same image encoder and text encoder. All the pre-training settings and the number of fusion layers are kept the same as in the original mPLUG pre-training. As shown in Figure 4, the fusion methods of co-attention and connected-attention both requires much more running time due to long visual sequence. Compared with the two fusion methods, our proposed skip-connected network is 4X faster and obtain better performance on both datasets. We also compare it with the asymmetric co-attention used in BLIP [6, 34] which only relies on the co-attention layers from images to text. Despite running slightly faster than the skip-connected network does, the asymmetric co-attention performs worse in accuracy on both datasets. The performance degradation is attributed to the information asymmetry and bias towards language, as shown in Section 5.2.1.

5.3.3 Large-scale Training

Combining the techniques introduced in Section 4 has dramatically increased the training throughput. With the utilization of memory saving and accelerated training techniques, the throughput of mPLUG improves 3X more from 124 samples per second to 422 samples per second, as shown in Table 6.

Model	In	Near	Out	Overall
SimVLM$_{base}$[7]	83.2	84.1	82.5	83.5
SimVLM$_{huge}$[7]	101.2	100.4	102.3	101.4
Oscar†[4]	85.4	84.0	80.3	83.4
VinVL†[9]	103.7	95.6	83.8	94.3
SimVLM$_{huge}$†[7]	113.7	110.9	115.2	112.2
mPLUG	86.34	81.5	90.49	84.02
mPLUG†	**116.7**	**113.75**	**117.0**	**114.8**

Table 7: Image captioning results on NoCaps validation split (zero-shot and finetuned), and {In, Near, Out} refer to in-domain, near-domain and out-of-domain respectively. † denotes the models finetuned on COCO Caption dataset.

Model	TR		IR	
	R@1	R@5	R@1	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.

5.4 Zero-shot Transferability

In this section, we examine the generalization of mPLUG and compare the zero-shot result on two Vision-Language and three Video-Language tasks.

5.4.1 Zero-shot Vision-Language Tasks

The pretraining of mPLUG adopts image-text contrastive and prefix language modeling tasks on large-scale image-text pairs. Thus, mPLUG has zero-shot generalization ability in image-text retrieval and image captioning. **Image Caption**: First, we take the pretrained mPLUG model and directly decode on NoCaps validation set without further finetuning. Following[7, 34], we feed a prefix prompt *"A picture of"* into the text encoder to improve the quality of decoded captions.

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].

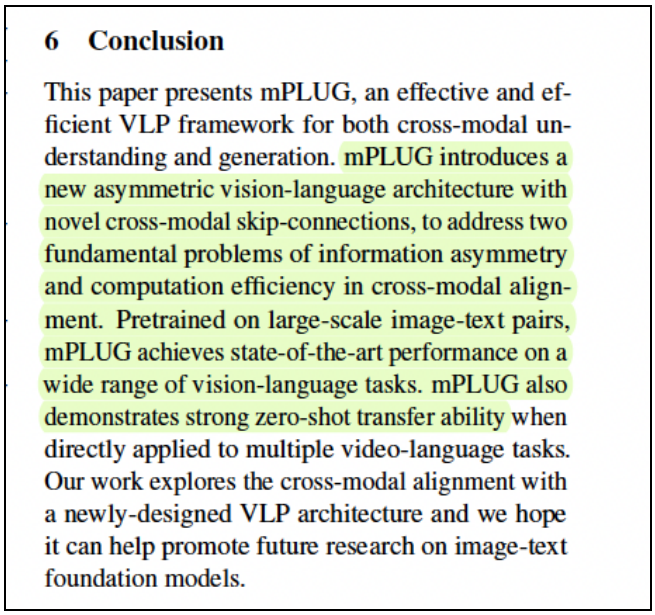
As shown in Table 7, the zero-shot performance of mPLUG is competitive with fully supervised baselines such like Oscar and VinVL. With further finetuning on MSCOCO dataset, mPLUG outperforms the SimVLM$_{huge}$, which use more pre-training image-text pairs and has larger model parameters. **Image-text Retrieval**: We perform zero-shot retrieval on Flickr30K. The result is shown in Table 8, where zero-shot mPLUG outperforms models (CLIP, ALIGN, Florence) pretrained with more image-text pairs. Following [34], we also evaluate zero-shot retrieval by the model finetuned on MSCOCO dataset. Table 8 shows that mPLUG achieves better performance than the previous SOTA models.

5.4.2 Zero-shot Transfer to Video-Language Tasks

To evaluate the generalization ability of mPLUG to Video-Language Tasks, we conduct zero-shot experiments on Video-text Retrieval, Video Caption and Video Question Answering. Following [34], we uniformly sample n frames for each video ($n = 8$ for Retrieval, $n = 16$ for QA, $n = 8$ for Caption), and concatenate the frame features into a single sequence. **Video-text Retrieval**: We evaluate the mPLUG models pretrained and further finetuned on the COCO-retrieval image-text dataset

You can now see that we have found another section of experiments. Zero-shot transferability may not be a concept we're familiar with, so we likely will have to add that to our "to understand" list.

Finally, let's read the conclusion of the paper.



6 Conclusion

This paper presents mPLUG, an effective and efficient VLP framework for both cross-modal understanding and generation. mPLUG introduces a new asymmetric vision-language architecture with novel cross-modal skip-connections, to address two fundamental problems of information asymmetry and computation efficiency in cross-modal alignment. Pretrained on large-scale image-text pairs, mPLUG achieves state-of-the-art performance on a wide range of vision-language tasks. mPLUG also demonstrates strong zero-shot transfer ability when directly applied to multiple video-language tasks. Our work explores the cross-modal alignment with a newly-designed VLP architecture and we hope it can help promote future research on image-text foundation models.

You might notice that although the conclusion was similar to the abstract and to the introduction, we already have a much better understanding of these words than we did towards the start of reading this paper. As I mentioned before, reading the paper the first time, we expect to really only understand ~10% of it, especially if we don't have the background of the papers which are being built on. With this same approach, we can start making our way through our "To Understand" list.

Conclusion

I hope this walkthrough of a first read of an AI paper in a new topic gives you the confidence to dive into a new problem topic. Both the process of reading wide and reading deep are iterative: we often need to re-search and re-read, and the act of making notes as you go can significantly help in building and cross-checking your mental model.