AN ANALYSIS OF LOW RECALL TEXT-IMAGE RETRIEVAL WITH LXMERT AND CLIP

An Honors Thesis Presented

By

GEORGE ZHIHONG WEI

Approved as to style and content by:

**** Mohit Nagaraja Iyyer 12/16/21 15:26 ****
Chair

**** Madalina Fiterau Brostean 12/16/21 15:58 ****
Committee Member

**** Philip Sebastian Thomas 12/17/21 12:20 ****
Honors Program Director

ABSTRACT

Previously, deep neural networks have found widespread success in computer vision due primarily to convolutional neural networks and the expressive feature extraction capabilities they have. More recently, deep learning has had a great impact on natural language processing with the introduction of attention and as a direct result Transformers and finally BERT. One such task that belongs in the intersection of these two subfields is the task of text-image retrieval, which is composed of text-based image retrieval (given a query text, retrieve the most relevant images) and image-based text retrieval (given a query image, retrieve the most relevant text/caption). In the past, models would be trained from scratch on this task. The recent paradigm shift to pretrain-finetune in deep learning has given rise to vision-language pretrained models which learn image and text associations during pretraining and get fine-tuned for vision-language downstream tasks. LXMERT, one such model, relies on a Faster R-CNN, a pretrained object detector model, to generate object detection features from the images and learns the joint associations of the modalities through some cross-modal Transformer blocks. In contrast, CLIP utilizes two separate but similar encoders for the two modalities -- a Vision Transformer (ViT) for the images and a regular Transformer for the text. In this paper, we analyze the performance of these two VLP models on the MSCOCO dataset by looking into the the patterns low recall queries and when CLIP outperforms LXMERT.

1 Introduction

The main tasks we are studying in this thesis are text-based image retrieval and image-based text retrieval on the MSCOCO dataset using deep learning. Specifically we explore the simple binary classification formulation of these tasks, where the objective is to identify if the image-text pair is a ground-truth pair or not, and the in-batch contrastive formulation, which focuses on increasing the inner product of the ground-truth pair and minimizing every unaligned pair. Although these formulations are pretty simple in principle, they have respectable performance at inference time. Section 1.1 through 1.3 will give an overview of the models selected, the dataset used, and the significance of this work. Section 2 presents related prior work in vision language pretraining. Section 3 discusses the methodology and experimental designs. Section 4 presents the results and observations from the experiments introduced in section 3. Section 5 concludes the project and discusses possible next steps.

1.1 Models

In this paper, the performance of two VLP models on low recall text-image retrieval was analyzed. The first is Learning Cross-Modality Encoder Representations from Transformers (LXMERT) and the second is Contrastive Language-Image Pre-training (CLIP). In the following sections, we give an overview of the models and why they were analyzed.

1.1.1 LXMERT

LXMERT (Tan and Bansal, 2019), a VLP model built as an extension to BERT (Devlin et al., 2019), a well known pre-trained natural language processing (NLP) model, was built to address visual question answering tasks such as VQA (Goyal et al., 2017) and GQA (Hudson and Manning, 2019) during pretraining and was fine-tuned for Natural Language for Visual Reasoning for Real (NLVR^2) (Suhr et al., 2019). The main claim to fame for this model is that this model was the only one to rank in the top-3 for both VQA and GQA. LXMERT relies on a Faster R-CNN (Ren et al., 2016) object detector for encoding images as well as five pre-training tasks: masked cross-modality language modeling, masked object prediction using RoI-feature regression, masked object detection using detected-label classification, cross-modality matching, and image question answering (Tan and Bansal, 2019). The reason why we chose this model was because it was the only existing VLP model with a HuggingFace Transformers (Wolf et al., 2020) implementation at the beginning of writing.

LXMERT has three different types of encoders – object-relationship encoders, language

encoders, and cross-modality encoders. Specifically, the model used had 9 language encoders, five object-relationship encoders, and five cross-modality encoders. The hidden activation function used was Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2020). The outputs are the learned image features, cross-modality matching output, and the language output. The cross-modality matching output is just the output of a pooling layer on the `[CLS]` token.

1.1.2 CLIP

OpenAI CLIP (Radford et al., 2021) is a VLP model that jointly learns separate image and text encoders from a comparatively simple pretraining objective of maximizing the correct image, text pairings within a batch. Specifically, a symmetric cross entropy loss is used as the objective function during pre-training. More importantly, CLIP is pretrained on the order of 400 million publicly available image-text pairs, which inherently regularizes the model and prevents overfitting to the dataset. This model was also chosen because of its high zero-shot transfer performance on various vision based benchmarks, such as ImageNet (Deng et al., 2009) and StanfordCars (Krause et al., 2013). It also has a HuggingFace Transformers implementation, which was released during early 2021.

For the image encoder in the publicly available pre-trained implementation, a Vision Transformer (ViT) (Dosovitskiy et al., 2021) base patch 32 was used. On the textual side, a Transformer (Vaswani et al., 2017) with the modifications described in (Radford et al., 2019) and base size of 63M-parameter 12-layer 512-wide model with 8 attention heads was used. A log parameterized learned temperature was used to control the distribution of the matching scores used within a batch, which is calculated as the dot product of an image feature vector and textual feature vector within the batch scaled by the learned temperature.

1.2 MSCOCO Dataset

The data used for this research is obtained from Microsoft's Common Objects in Context (MSCOCO) (Lin et al., 2015). MSCOCO was first made publicly available 2014 to push the state of the art in object recognition within the broader context of scene understanding. The dataset was a result of joint collaboration of academic and industry computer vision researchers. MSCOCO is composed of 164,062 images with five textual captions for each image. These images are further divided into 82,783 training images, 40,504 validation images, and 40,775 test images.

In image-text retrieval, the MSCOCO dataset is used for benchmarking performance of the models. However, since the captions for the test images are not publicly accessible, most

researchers use the Karpathy and Fei-Fei (2015) split, which take 5,000 images each for the validation and test splits from the original validation images and provides the remaining 30,504 images to the training split. So, there are 113,287 training images, and 5,000 images each for validation and testing.

1.3 Significance

The main contributions of this work is the analysis of what medium and large scale VLP models fail to retrieve given textual or visual queries. Since LXMERT and CLIP were not explicitly trained with text-image retrieval in mind, what do they learn to rank between candidate documents? In the case of LXMERT, we look at the performance on the task by simply adding a two linear layer binary classifying head like was done in (Li et al., 2020). In the case of CLIP, we look at its zero-shot transfer performance. Because of this, we also look at when CLIP outperforms the fine-tuned LXMERT to examine the image-text associations it learned during pre-training.

2 Background

ViLBERT (Lu et al., 2019) had two different "streams" for the two input modalities. On the visual side, a Faster R-CNN (Ren et al., 2016) pretrained on Visual Genome (Krishna et al., 2016) was used to encode and featurize the images. On the linguistic side, BERT base pre-trained on BookCorpus (Zhu et al., 2015) and English Wikipedia was used to get the textual embeddings. For an L layer ViLBERT, the textual data was sent through $L - k$ layers of transformer encoders for self-attention. For the other k layers, the separate streams of processing came together in "co-attentional transformer layers" followed by a traditional transformer encoder. Essentially, after the intermediate visual and linguistic hidden states were computed, this module would swap the key and value matrices with that of the other modality.

In like manner to masked language modeling, ViLBERT has two additional pretraining tasks. One is masked image modeling, where a section of the image is masked and the model predicts the distribution over the semantic classes of the image region in an effort to minimize the KL divergence with respect to the output distribution of the Faster R-CNN. The other is multi-modal alignment, which presents the model with a pairing of an image and some text and asks it to predict whether they go along with each other. ViLBERT was pretrained on the tasks using the Conceptual Captions (Sharma et al., 2018) dataset, which consists of around 3.3 million images with weakly associated captions scraped from alt-text

enabled images.

After the pre-training stage, ViLBERT was fine-tuned for Visual Question Answering (Krishna et al., 2016), Visual Commonsense Reasoning (Zellers et al., 2019), Grounding Referring Expressions (Zhang et al., 2018), and Caption-Based Image Retrieval. For most of the fine-tuning, it usually involved learning an additional linear layer or two. Specifically in caption-based image retrieval, the authors computed alignment scores in a 4-way multiple choice fashion by either substituting random captions, images, or a hard negative, softmaxing, and trained on cross-entropy loss. At prediction time, each caption-image pair in the test split was scored and then sorted.

A slightly different approach was shown in Unicoder-VL (Universal Encoder for Vision and Language) (Li et al., 2019), which has a single stream model that processes the images and text together. In similar fashion to ViLBERT, Unicoder-VL uses Faster R-CNN (Ren et al., 2016) for picking image regions and the same additional pre-training tasks. However, instead of predicting a distribution over semantic classes for the masked regions, Unicoder-VL opts for classification. The authors chose to pre-train the model on the combination of Conceptual Captions and SBU captions (Ordonez et al., 2011) (which consists of 1 million image-caption pairs scraped from the web) with the model initialized with BERT base weights (Devlin et al., 2019).

Unicoder-VL was fine-tuned on two downstream tasks: image-text retrieval and visual commonsense reasoning. On image-text retrieval, Unicoder-VL trained from scratch still outperformed the existing state of the art and benefited from deeper models. They formulated image-text retrieval as a ranking problem using the hardest negative triplet loss, maximizing the margin on negative samples.

In 2020, a key addition to VLP was introduced–adding bounding box ground truth classes or predicted classes (Li et al., 2020). Specifically, their proposed model Oscar (Object-Semantics Aligned Pre-training) was pre-trained using Word-Tag-Image $(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$ triples. The main motivation for doing so was the fact that the most salient objects in an image are often described in the accompanying text. Both Masked Token Loss $\mathcal{L}_{\text{MTL}}$ and a contrastive loss $\mathcal{L}_{\text{C}}$ is used, which measures the alignment of the tag and the image.

Oscar was ultimately pre-trained on a combination of MSCOCO (Lin et al., 2015), Conceptual Captions (Sharma et al., 2018), SBU Captions (Ordonez et al., 2011), Flickr30k (Young et al., 2014), and GQA (Hudson and Manning, 2019), totaling 4.1 million unique images and 6.5 million text-tag-image triples. For the downstream tasks, Oscar was fine-tuned on the image-text retrieval, image captioning, VQA, GQA, and NLVR^2 tasks. Unlike the previous two VLP models, they managed to make Oscar caption images by fine-tuning on the seq2seq objective, further masked language modeling, allowing caption tokens to attend

to the tokens before its position while allowing full attention on image regions and object tags, and incorporating beam search. At the time of publication, Oscar achieved the new state of the art in all (6) but one task, that task being GQA.

For ablation analysis, the importance of the tags were scrutinized. Ultimately, they found that fine-tuning with object tags led to faster convergence and gave Oscar an advantage over other VLP methods on all downstream tasks. Additionally, the authors analyzed the performance of object detectors trained on Visual Genome (Krishna et al., 2016) versus Open Images (Krasin et al., 2016), finding that VG tags perform slightly better than that of the other, postulating that it is due to the larger variety in the set of objects there. However, the object detector had higher precision when trained on OI.

Earlier this year, a group of Korean researchers noticed the particularly large amount of computation involved in processing the image modality in existing VLP models (Kim et al., 2021). On the textual side, language encoders are much shallower than that of the image size, which usually have some underlying deep CNN. As such, much of the computation time is due to the visual encoding step, something that these researchers wanted to address. In order to reduce the amount of computation dedicated to the visual modality, their model encodes images with a simply learned linear projection. Specifically, the linear projection used is patch projection, introduce by (Dosovitskiy et al., 2021) in ViT. Also, instead of performing layer normalization after multi-headed self attention like in BERT (Devlin et al., 2019), the authors did the other way around. During the finetuning stage, image augmentation with RandAugment (Cubuk et al., 2019) was used.

Pre-trained on MSCOCO, Visual Genome, SBU Captions, and Conceptual Captions, ViLT was fine-tuned on VQA, VCR, and image-text retrieval. Out of all of the models they studied, ViLT is the most parameter and time efficient. Although this model did not achieve new state of the art results, the results were fairly competitive with the models that achieved these results while running several orders of magnitude faster. From their ablation analysis, performance monotonically increases with more pre-training steps, masking whole words instead of subtokens in masked language modeling as well as finetuning with augmentation. However, the proposed addition of masked patch prediction, the adapted objective of masked region modeling compatible with patch projections, contributed little to the performance of the model.

Even more recently, a group of researchers proposed an end-to-end visual language pre-trained model which learns its own object detector (Kamath et al., 2021) rather than relying a pre-trained counterpart that outputs static representations of each image like was used in ViLBERT, Unicoder-VL, and Oscar. A direct benefit of this approach is that the model can then be used to fine-tune on text-conditioned object detection, unlike previous approaches.

As such, a few additional pre-training objectives were introduced such as soft token prediction and contrastive alignment.

The former is used to predict which tokens from the text refers to the given detected object. Whenever a detected object is not matched with referring text, the model is trained to predict $\varnothing$, or "no object." The latter is used to enforce alignment of the embeddings of the detected objects at the output of the decoder and the referring text at the output of the cross encoder. This is to learn encodings that are closer in feature space when the object is directly linked to the text.

3 Methodology

3.1 Language, Frameworks, and APIs

The fine-tuning/zero-shot transfer of the deep learning models was written primarily in Python (Van Rossum and Drake Jr, 1995). I used PyTorch (Paszke et al., 2019) as the deep learning framework of choice since it is heavily preferred in the deep learning research community, simplifies the multi-dimensional tensor operations, and has automatic differentiation built in. PyTorch is a free, open-source deep learning framework primarily developed by Facebook's AI Research lab (FAIR). PyTorch's automatic differentiation engine is very powerful since it records the structure of mathematical operations that have been performed and automatically computes the gradients for use during optimization. This frees up researchers from manually calculating gradients.

Since this project involves some natural language processing, I used HuggingFace's Transformers (Wolf et al., 2020) package. Over recent years, Huggingface has had an increasing influence in the deep natural language processing research area by having a large assortment of pre-trained transformer based NLP models in their Transformers package. Of particular interest to me in this package/API is the inclusion of pre-trained multi-modal models, like LXMERT and CLIP. Since HuggingFace already has a pre-trained LXMERT, I will be taking their model for fine-tuning on the image-text retrieval tasks. A pre-trained CLIP is also included in the framework, but is too large to fine-tune. As a result, its zero-shot transfer capabilities are going to be tested. Additionally, all of the HuggingFace's models have a PyTorch implementation, so there are not going to be any issues with the deep learning framework I am using.

3.2 COCO Preprocessing

Oscar provided a PyTorch archive file of all of the captions for each split and a TSV of the image keys for the 1k test split, so no additional preprocessing was necessary for the captions.

Since LXMERT uses Faster R-CNN to encode the images for the model, I used an existing pre-trained Faster R-CNN[1]. Specifically, this repository contains a pre-trained Faster R-CNN on Visual Genome. As called in the LXMERT paper, ResNet-101 (He et al., 2015) was used as the backbone CNN to generate the 36 region and spatial features per image. For computational efficiency, I preprocessed all the images through this Faster R-CNN and dumped them into a HDF5 (The HDF Group, 1997) file, a common file format used in computer vision to compress images and avoid loading all images into RAM.

In contrast, CLIP trains an image encoder from scratch, so the Faster R-CNN preprocessed images were not used. Instead, all images were first renamed to their COCO image ids, then read into NumPy (Harris et al., 2020) arrays, and finally dumped into a separate HDF5 file for the minival, val, and test splits.

3.3 LXMERT Fine-Tuning

Inspired by Oscar (Li et al., 2020), we fine-tuned LXMERT on text-image retrieval with a binary classification formulation. So, given a possibly unaligned image-text pair, the model is tasked with identifying if the pair is a ground-truth pair or not. For an aligned image-text pair, we unalign the pair with 30% probability. Of that probability, half of the time a random negative image is sampled and the other half of the time a random negative caption is sampled. When a text-image pair is fed to the model, we take the final representation of the `[CLS]` token and train it to learn when the image and text are aligned. At prediction time, the candidates are sorted in descending order by the output matching score.

The hyperparameters that we used for fine-tuning was a batch size of 32, learning rate of $1e^{-5}$, and a weight decay of $1e^{-4}$. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with default betas and epsilon as defined by the Trainer API (Wolf et al., 2020). We fine-tune for 10 epochs and do minival evaluation after each epoch. At the end of training, the best checkpoint on the recall sum metric is used to evaluate the model.

[1] https://github.com/shilrley6/Faster-R-CNN-with-model-pretrained-on-Visual-Genome

3.4 CLIP Zero-Shot Transfer

Since CLIP is such a large model, fine-tuning is not feasible. Instead, we feed it a batch of image-text pairs and get the diagonal entries of the square similarity matrix it produces to calculate the loss as the matching score. Since the PyTorch DataLoader (Paszke et al., 2019) takes care of getting every possible text-image pair in the test split, this is all we need to evaluate. Since we don't fine-tune on this task in specific and presumably was not trained on the images in MSCOCO, CLIP will be evaluated/analyzed on its zero-shot transfer capabilities

3.5 Evaluation

For image-text retrieval, given either each test caption or test image as input, the model needs to retrieve the most relevant test images or captions respectively. Here, recall@k is the main evaluation metric, which is the proportion of queries that had a ground-truth caption retrieved in the top k retrieved results.

Usually, both the 1k and full test split of MSCOCO is used to gauge the performance of the model. For perspective, $1000 \times 5000 = 5 \times 10^6$ image-text pairs need to be analyzed in order to calculate these metrics. The full test split contains $5000 \times 25000 = 1.25 \times 10^8$ image-text pairs, which is infeasible to do evaluation on the full val split during training.

3.6 Compute Resources

To address computational requirements, all experiments, including training and inference, were performed on the Gypsum GPU cluster, a cluster of GPU compute nodes located at Massachusetts Green High-Performance Computing Cluster (MGHPCC). This cluster was accessed through SSH, a secure command line utility to remotely connect to another computer. All experiments were scheduled through the SLURM workload manager.

Since modern language models are generally pretty large, most of the recorded experiments were ran on either Nvidia GeForce GTX 1080TIs or RTX 2080TIs on the long partition, with 11 Gigabytes of dedicated memory each. We ran all experiments on a single node with 8 GPUs, using PyTorch's Distributed Data Parallel shorten training/inference time. Training loops for LXMERT generally took between 24 and 28 hours to finish.

Method	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
	1K Test Set						5K Test Set					
LXMERT (Tan and Bansal, 2019)	63.0	90.9	95.8	55.4	87.8	94.9	36.5	67.6	79.4	31.3	62.0	74.7
CLIP (Radford et al., 2021)	30.5	56.6	67.9	24.8	51.9	64.2	13.7	30.0	39.9	10.9	26.1	35.3

Table 1: Results on Image-Text Retrieval for LXMERT and CLIP

4 Results

Table 1 contains the image-text retrieval results of the fine-tuned LXMERT and zero-shot transfer of CLIP on MSCOCO. As can readily be seen, the fine-tuned LXMERT outperformed CLIP on all metrics of interest. This is expected since LXMERT has seen and was trained on the dataset that inference was performed on. Despite never having seen the images or the captions before (presumably), CLIP managed to generally understand the pairings of text and images that beat random ranking.

Even though these extrinsic metrics can give us a relative sense of the performance between these two different models, what queries do the models characterize as "hard"? When does CLIP perform better than LXMERT? We address these questions in 4.1.

4.1 Error Analysis

For simplicity, only the 1K test split of MSCOCO was analyzed. We define a "hard" query as a query where the model in question does not retrieve a ground-truth document in the top 10 documents. This cutoff is fairly arbitrary, but most search engine users expect a relevant document retrieved within the first couple of documents, so this is an approximation of human preferences.

4.1.1 Text Retrieval

Across both models, three key observations stood out on the "hard" image queries.

1. Specific details in the images are either not mentioned or wrong in the retrieved captions.

2. The salient subject/objects/actions in the picture get mistaken.

3. Some retrieved captions are plausible for the query image.

Examples of these image queries and their retrieved results are shown in table 2. As can be seen in the first example, many incorrect words are present in the retrieved captions, such

Image Query	Ground-Truth Captions	LXMERT Top 10 Retrieved Captions	CLIP Top 10 Retrieved Captions
	a bus pulling up to the bus stop in a city A bus is parked near a bus stop on a street. White and blue buses parked on the side of the city road to let passengers in. A buss stopped at a bus stop on a city street Two blue and white buses stopped at a bus stop.	**Blue, white,** and green **passenger bus parked at a stop.** A red and white striped **bus** traveling past a store. A red, double decked **bus** is on a city street as small cars pass it on the street. A red and black double decker **bus** travelling on a street. A very big city **bus** on a big street. A two-floor **bus** riding along a busy road. A red double stacked **bus** traveling down the road. A **bus** with stripes is **on the street outside of some buildings.** A double decker **bus** is driving down a street. A red two level city **bus on a street** with cars behind it and in the other lane.	A striped **bus** travels down a city street at night. A **street is displayed** at night with time lapse photography. An electric **bus** is driving down a **city street.** A long **passenger bus is stationary in town** near several multiple story buildings **along a sidewalk.** The **city bus** is turning the corner of the street. A double decker **bus** is driving down a street. Cars drive past a **bus on a street.** A very big city **bus** on a big street. A red, double decked **bus is on a** city **street** as small cars pass it on the street. A two-floor **bus** riding along a busy road.
	A cow attempting to get water from a leaking pipe. There is a cow under a tree branch looking up. A brown humped cow is drinking from a pipe. A brown cow has its head under a piece of bark. a brown cow reaching for a tree limb	Of two dogs, only one may be the victor. a close up of a dog wearing a tie A close up view of a very cute scruffy dog. THERE IS A DOG WITH A PACK ON HIS BACK Two horses have feathers on their head. A dog wearing a tie poses for the camera. A brown and white dog wearing a tie on carpet. A brown dog with a stick in its mouth. A brown and white dog wearing a neck tie. A small dog with a collar in a room.	Lady enjoying bathing a dog in a small blue basin in a bathroom or kitchen. A fire hydrant pouring water from it with a neon blue glow to it. A picture of a giraffe drinking some water. a stuffed bear with goggled on wearing snow skies A dog faces a light up cow ornament. A woman is giving her dog a bath. a polar bear standing in the snow with its reflection in the water A young girl is lathered up with toothpaste. a giraffe bending down to drink from a body of water A large brown dog laying next to a blue pool.
	A desk with a small white computer set up on it. A desk with a laptop, alarm clock, and office supplies. A desk with a laptop and various other items on top of it. A laptop and keyboard sit on a desk. Computer and other office supplies on a neatly organized work space.	**a desk with a cup and a keyboard** A cup of coffee sitting next to a computer keyboard. **A desk with a laptop and various other items on top of it.** A desk with a computer, office items, and CDs on it. **A laptop sitting on a desk with other items.** **A computer on a small wooden desk cluttered with many items.** a coffee cup is next to a white keyboard **a desk that has a keyboard and a cup on it** **A desk with a computer and other items.** **A laptop and keyboard sit on a desk.**	A keyboard, a remote control, and two circuit boards with wires, all on top of a map. We see a low angle view of a cluttered desk top. A remote control, various electronic componets, and a computer keyboard sitting on a map. A crowded desktop is shown in very dim lighting. an image of a two large monitors in a computer center **A desk with a computer and other items.** Several computer monitors and keyboards sitting on the same desk. A full view of a working office with computers. An open book in front of a keyboard and monitor A desktop computer with a grassy field background

Table 2: Examples of the four different types of hard image queries: 1) similar semantic/salient objects, 2) mistaken subject/object/activity, and 3) plausible caption retrieved.

as that the picture is during nighttime or that the bus is a double-decker. The bus being stopped at a bus stop is not explicitly mentioned. Details in the pictures were what both models particularly struggled with.

With the third example, the subject is clearly a cow. However, LXMERT in this case insisted that the image is of a dog or horses. This can be justified given that the majority of the animal is not in the frame of photo. LXMERT seemed to mix up dogs with cats and a jetski for a surfboard. CLIP did not seem to learn what baseball looked like so kept insisting that those images were of skateboarding, skiing, or tennis.

For the final example, the fifth caption retrieved seems to be plausible with the given image query of a laptop on a small desk with various other office supplies. This seemed to occur more than expected, so it begs the question of whether the ground truth captions are the only plausible relevant captions for a given image.

4.1.2 Image Retrieval

For "hard" text queries, there seem to be three commonalities between LXMERT and CLIP.

1. Typos in caption lead to nonsensical retrievals.
2. Long modifiers/abstract concepts are generally forgotten or ignored.
3. Underspecified/general captions lead to more relevant image retrievals.

Textual Query	Ground-Truth Image	Top 10 Retrieved Images
People driving and observing giarffs in a natural environment.		
CLIP Retrievals		
A view of a bathroom that needs to be fixed up.		
CLIP Retrievals		
The group of people are gathering together in the yard.		
CLIP Retrievals		

Table 3: Examples of the three different types of hard textual queries: 1) typos in caption, 2) long range modifiers/abstract concepts, and 3) underspecified queries that have plausible images retrieved. White rows are LXMERT retrievals and gray ones are those for CLIP.

For the first example giraffes being misspelled as "giarffs" confused the model and therefore made the retrieved images pretty irrelevant. Another case when this negatively affected the performance of the models is when cat was misspelled as "car," leading to many retrieved images containing cars and only one picture of a cat being retrieved. These typos need to be addressed for retrieval metrics on text-based image retrieval to be correct.

The second example contains a modifier that severely alters the meaning of the sentence, namely "that needs to be fixed up." Without completely understanding/remembering this long range modifier, all of the retrieved images were just normal bathroom pictures rather than bathrooms that are damaged. This heavily impacted LXMERT in other queries with phrases that meant the subject was preparing for something (i.e. "Getting geared up to do a little snow boarding." and "Two boys are ready to go play in a baseball game."). With CLIP, abstract concepts/objects that are not as tangible were not understood that well (i.e. "Someone is taking notes or an open book test on the computer.").

Like in text retrieval, some textual queries were vague/open enough for there to have relevant non-ground-truth images. Consider the top two images of the ten images retrieved. Those images are plausible images for the given caption, meaning that more research needs to be done in automatically identifying relevant documents that are not the ground-truth.

4.1.3 When does CLIP perform better than LXMERT?

Considering that we fine-tuned LXMERT on the training split of MSCOCO, it is pretty obvious that its performance during inference would be better than that of CLIP. However what isn't clear is if there are specific sorts of queries when CLIP outperforms the fine-tuned LXMERT. So, we are going to be analyzing the queries where CLIP retrieves a ground truth document at a lower $k < 10$ than that for LXMERT and the difference in ranks is at least 10.

Image Queries The number of image queries where the above condition holds is 13. Of those 13, 3 of them were bathroom images (where CLIP retrieved a ground-truth caption at 2 while LXMERT would need at least 14. It is unclear why LXMERT did not perform better since bathroom pictures are not infrequent in different image datasets.

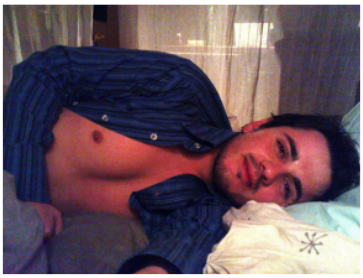
For either ambiguous or low quality images, CLIP performed significantly better in text retrieval as compared to LXMERT. This is almost surely due to the much larger amount of pre-training data CLIP saw, which made it learn to identify objects from uncommon viewpoints and be more robust to low quality.

Finally, LXMERT performed much worse in comparison to CLIP when the image query had people in uncommon poses for pictures. In similar manner to the last category of

images, the larger image pool CLIP has seen probably led to better reasoning on these types of images.

(a) Bathroom images

(b) Ambiguous/unclear/low quality images



(c) People in uncommon poses for pictures

Figure 1: The categories of image queries that CLIP retrieves captions better than LXMERT.

Textual Queries The number of textual queries that satisfy the condition established in 4.1.3 is 51. In a similar vein to the first example in the section above, LXMERT has a couple of visual topics that it seemed to struggle with. The captions of the blurry black-and-white image of the vase, bathrooms, the white and pink umbrela, and boys getting baseball bats gave the model a hard time, but not for CLIP.

Of particular note are skiing and living room captions. From inspection, it also seems that uncommon sentences (where it is hard to synthesize all of the representations of each object/subject mentioned) gave LXMERT a particularly hard time. These examples are shown in table 4. CLIP must have seen more examples of skiing, living rooms, and possibly more general and composable image features to be so dominant in the categories as compared to LXMERT.

Category	Textual Queries
Skiing	A man riding skis on top of a snow covered slope. a person riding skis on a snowy surface
Living Room	A living room has a couch and a rustic chest for a coffee table. A PHOTO OF A LIVING ROOM WITH COUCHES AND A TABLE A white couch sitting in a living room next to a wooden table.
Uncommon sentences	A small old fashioned TV with rabbit ears on a small stand. A pub sign for a small home bar has a crooked letter. A man approaches a team of two for advice.

Table 4: Captions that CLIP retrieved relevant images better than LXMERT. For the last category, the average ground-truth retrieval is ∼80.

5 Conclusions

In this paper, we identified which queries the models found difficult in retrieving a relevant document of the other modality. We were able to incorporate multi-gpu distributed training/inference with automatic mixed precision to speed up experiments. We fine-tuned LXMERT on the MSCOCO dataset which contains around 113,000 training images with 5 accompanying captions for each image. Additionally, we evaluated the fine-tuned LXMERT and performed zero-shot transfer with CLIP and got reasonable retrieval performance. From there, we sampled 40 "hard" queries for both image-based text retrieval and text-based image retrieval and pinpointed several key categories that led to worse retrieval. Also, we analyzed what types of queries that zero-shot CLIP outperformed LXMERT on.

5.1 Limitations

This thesis predominantly focused on two VLP models available within HuggingFace, namely LXMERT and CLIP. As a result, the results are inherently limited as they cannot be generalized to independent open-sourced repositories. Another key limitation is performing analysis on only the 1K test split of MSCOCO instead of the full 5K version. Perhaps the greater volume of queries could lead to more noticeable patterns in the difficult queries. Even if the full 5K test split was not used for analysis, all of the low recall queries need to be analyzed instead of just sampling 40 of them. Also, fine-tuning CLIP for the same number of epochs as that for LXMERT would have been a more fair comparison.

5.2 Future Work

Because of the limitations stated, an interesting line of work is seeing if these results apply to more VLP models such as ViLBERT, MDETR, etc. Another easily implemented change would be to perform the same analysis on the full 5K test split. Since the Flickr30k dataset has a similar format to MSCOCO, it would be interesting to augment the training and testing data of MSCOCO with this dataset. As seen in 4, the calculation of the evaluation metrics lead to some plausible retrieved documents being marked as irrelevant even though the seem relevant in hand inspection. A useful and good line of work is to improve the labeling of truly relevant captions/images. Finally, fine-tuning CLIP would also be worthwhile to pursue.

References

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2019). Randaugment: Practical automated data augmentation with a reduced search space.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Hendrycks, D. and Gimpel, K. (2020). Gaussian error linear units (gelus).

Hudson, D. A. and Manning, C. D. (2019). Gqa: A new dataset for real-world visual reasoning and compositional question answering.

Kamath, A., Singh, M., LeCun, Y., Misra, I., Synnaeve, G., and Carion, N. (2021). MDETR - modulated detection for end-to-end multi-modal understanding. *CoRR*, abs/2104.12763.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions.

Kim, W., Son, B., and Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision.

Krasin, I., Duerig, T., Alldrin, N., Veit, A., Abu-El-Haija, S., Belongie, S., Cai, D., Feng, Z., Ferrari, V., Gomes, V., Gupta, A., Narayanan, D., Sun, C., Chechik, G., and Murphy, K. (2016). Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., Bernstein, M., and Fei-Fei, L. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Li, G., Duan, N., Fang, Y., Gong, M., Jiang, D., and Zhou, M. (2019). Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.

Ordonez, V., Kulkarni, G., and Berg, T. L. (2011). Im2text: Describing images using 1 million captioned photographs. In *Neural Information Processing Systems (NIPS)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Suhr, A., Zhou, S., Zhang, A., Zhang, I., Bai, H., and Artzi, Y. (2019). A corpus for reasoning about natural language grounded in photographs.

Tan, H. and Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers.

The HDF Group (1997). Hierarchical Data Format, version 5. https://www.hdfgroup.org/HDF5/.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Zellers, R., Bisk, Y., Farhadi, A., and Choi, Y. (2019). From recognition to cognition: Visual commonsense reasoning.

Zhang, H., Niu, Y., and Chang, S.-F. (2018). Grounding referring expressions in images by variational context.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.