Mask it right for V+L: Adjusting MLM for vision[ROY: please capitalize all non-stopwords in titles (paper and section titles, but not paragraphs]

Anonymous EMNLP submission

Abstract

Vision-language pre-training (VLP) have significantly improved performance in a variety of Vision-and-Language (V+L) tasks. One of the primary pre-training objectives of VLP is the masked language modeling (MLM) pre-training[MICHAEL: Not needed:, which is found to be essential]. [MICHAEL: Motivated by replaced single-modal] As in the text-only MLM task in BERT, input word tokens are randomly masked, and the model predicts the masked tokens [MICHAEL: but in VLP, the model inspects both] the text context and the image. We argue that the current MLM pre-training method is sub-optimal for VLP, as it does not sufficiently leverage the visual aspect of the VLP setting. We [MICHAEL: replaced: establish] observe the [MICHAEL: following] disadvantages of the current MLM pre-training method: (1) [MICHAEL: >50%] more than half the masked tokens are stop-words and punctuation; (2) in a third of the sentences, no token is being masked, therefore, [MICHAEL: MLM is not active] MLM is not data efficient; (3) we show that it does not require the model to look on the image enough. [MICHAEL: Intro needed before this sentence] We investigate a range of masking strategies specific to the VLP setting that address these shortcomings and aim at encouraging a pre-trained model to pay more attention to images and lead to better fusion of text and image in the learned representation. [MICHAEL: We learn to identify which tokens in the text are to be masked in order to require the pre-trained model to actively rely on the image.] We learn what is important to mask from the image perspective,[ROY: be more concrete. what do you mean by "learn what is important?] [MICHAEL: Our approach presents] a hierarchy of semantic classes ranked by the necessity of the image . [MICHAEL: moved above: Using this hierarchy, we propose alternative masking strategies for MLM that require more information exchange with the image.] We show that after pre-training with our alternative masking strategies,[ROY: mention the LXMERT model, numerical gains][YONATAN: I will when 100% will be available, for now we have 10%, 20% which may be too long for the abstract] downstream models achieve improved results on 3 downstream benchmarks V+L datasets (GQA, VQA, NLVR2).[1]

1 Introduction + Related work[ROY: if this is a long paper, have a different related work section. If it is short, you can merge intro and related work, but still call it Introduction]

Figure 1: Illustration of our approach: [MICHAEL: during pre-training,] we mask words that require the image in order to be predicted, instead of random tokens. The model wouldn't be able to [MICHAEL: complete: guess] the masked words without the image, and it will be forced to look at the image in order to succeed. As a result, the model will perform better in the downstream question answering task.

[1] Our code, pre-trained, and fine-tuned models are published at anonymous.

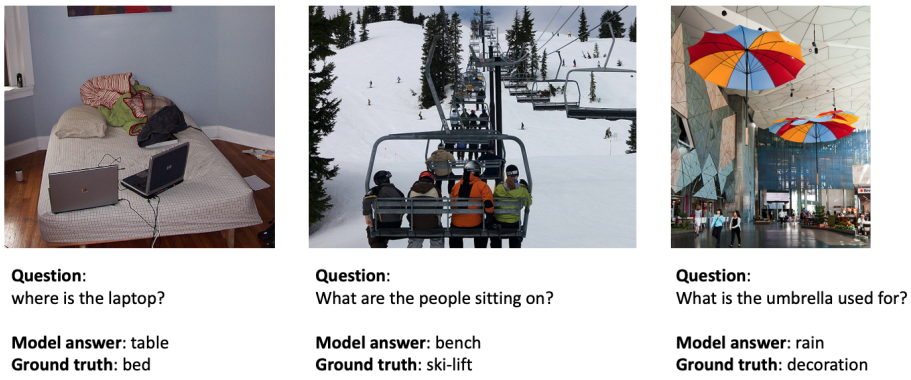

Figure 2: LXMERT mistakes observed on examples from GQA and VQA. The tendency of V+L models is to predict something that is correlated with the text, or common answers. In many cases, the prediction is not an item that even appears in the image.

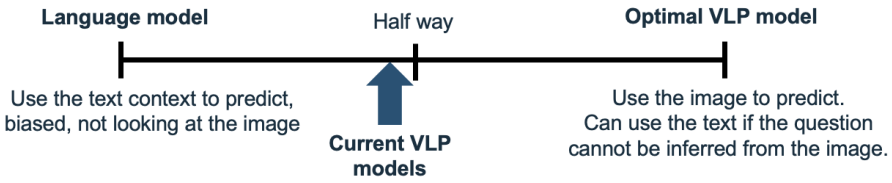

Figure 3: On one side, we have text-only MLM model.[ROY: what is this line measuring? what does it mean to be half-way here?][YONATAN: Measuring the use of text-context vs image-context. We are Half-way in using the image-context, because in >50% of the cases it is not using the image.] On the other side, we have the desired optimal VLP model. Because >50% of the tokens are stopwords or punctuation, the VLP model does not need the image in order to succeed in the MLM task in >50% of the cases (Elaborated in Section 3.1).

Challenges in VQA generalization (lack visual understanding): [MICHAEL: Not needed: Language and vision tasks have gained popularity in recent years.] Those tasks demand deep understanding of both the text and the image. Many works show that models can succeed on VQA tasks using a strong language priors, without deep understanding of the image (Zhang et al., 2016; Jabri et al., 2016; Goyal et al., 2017). Results are sometimes improved in superficial levels, but there are still challenges to overcome for tasks with more compositional structure (Bitton et al., 2021; Agarwal et al., 2020; **?**).[ROY: something is weird in the third citation][YONATAN: new paper, I will update the citation when its available] Research is trying to address this problems in multiple ways. More challanging datasets such VQA 2.0 and GQA (Goyal et al., 2017; Hudson and Manning, 2019) have been presented. Novel models with richer visual representations (Zhang et al., 2021) are presented, and some works try to "force"[ROY: use "this format" for parentheses][YONATAN: At the end I will fix all parentheses and citations format] the model to look at the "correct" image regions (Liu et al., 2021; Yang et al., 2020).

Vision Language Pretraining (VLP) : Recently, pre-trained vision-language (V+L) models such as ViLBERT (Lu et al., 2019), LXMERT (Tan and Bansal, 2019) and UNITER (Chen et al., 2019) have pushed the state-of-the-art research to a new level. Those models are pre-training on a large corpus of visual and language data. The pre-training objectives in many cases are: (1) Masked language modeling (MLM), where a model tries to predict masked tokens given the sentence and the image. Using the image is a[ROY: something is wrong here][YONATAN: Maybe now better] key-difference from BERT (Devlin et al., 2018):[ROY: You might want to talk more boradly about text-only models, **such as BERT** 3. this feels like a footnote rather than part of the main text] the model can predict masked words using the vision modality. (2) Masked region modeling (MRM), where the model tries to predict masked visual objects features, (3) Sentence-image matching, where the model predicts whether the sentence belongs to the image. Some models (Tan and Bansal, 2019; Li et al., 2021), also perform the downstream task

(question-answering), during the pre-training phase. [ROY: try to rephrase the next paragraph as a story rather than a list of findings]Previous works have found that: (1) MLM objective is important (Chen et al., 2019; Huang et al., 2020; Hendricks et al., 2021). (2) MRM objective is not always important (Su et al., 2019; Hendricks et al., 2021). (3) Sentence-image prediction is not always important (Hendricks et al., 2021; Li et al., 2019). For this reason, we focus on the MLM objective and we leave the other pre-trained objectives untouched. To the best of our knowledge, most VLP perform MLM in the same way that we claim to be sub-optimal~~.~~ :[ROY: we haven't claimed anything yet about sub-optimiality] Shin et al. (2021) recently reviewed how transformer architecture has been incorporated into visuo-linguistic cross-modal tasks. It includes 19 architectures that are all performing the masking in similar ways to the original BERT MLM which is random token masked in probability of 15%.[ROY: the BERT masking strategy feels like something you should have introduced earlier, not here on the fly]

The problem with language models in V+L pre-training (VLP): Language models learn spurious correlations in training data, sometimes leading to bias (Hendricks et al., 2018; Agarwal et al., 2020; Jaunet et al., 2021). Rabbits eat carrots. Pilots are usually men. Oranges are orange.[ROY: are these examples of biases? this is not a way to write an intro. Please make it part of the text.] V+L generalization is sometimes damaged from the same reason, as seen in Fig. 2. The bias is caused by the way the multi-modal transformers are trained (Yang et al., 2021; Hendricks et al., 2018). If the model sees many text-image pairs of rabbits eating carrots in the training data, then given a test image with a tiger eating carrot, the model will be biased to predict "rabbit" when asked who is eating the carrot. If we want to "de-bias" the V+L models, we can try alternative masking strategies that increase the necessity of the image during pre-training.[ROY: to discuss tomorrow: I disagree with this argument. You say that models are trained on multiple examples of rabbits eating carrots, which make them over-confident in the answer "rabbit" to something eating a carrot. But your proposal doesn't change that: given that most of the examples have rabbit, it wouldn't matter if you mask all the rabbits, the model would still learn these correlations. You might want to carefully pick the non-rabbit examples and mask them, but this is not what we're doing][YONATAN: The image helps to de-bias. Instead of focusing more on the text, we will now focus more on the image. Instead of completing text that "make-sense" as language-model does, the model will **learn** to look at the image in order to predict. It will lean on the image-context more than it does now. A model that is similar to a language-model, completes text that "make-sense" with the text context. Carrot? Rabbit! A VLP model that is looking at the image, completes text that **from the image modality**. Carrot? Let's look at the image and see what is appears with carrot!] This way we will shift from Language models toward better VLP models that are using more of the image and less training data bias, as exemplified in Fig. 3. In other words, in the trade-off between the text-context and the image-context, we hypothesis that VLP model will benefit more from the image-context.

Thought experiment: text-only MLM vs V+L MLM: Given the following example text-only MLM: "[MASK] have muscled hind legs that allow for maximum force, maneuverability, and acceleration that is divided into three main parts". There are several possible answers: kangaroos, horses, wolves, rabbits, etc. The ground truth is rabbits [2].[ROY: footnotes come after the punctuation, without white-spaces] A perfect model (or even person) has *low* probability to succeed in the task. In V+L MLM, we are given the following example: "A [MASK] is eating the carrot" and an image of a single animal eating carrot. If the animal seen in the image is rabbit or tiger, a perfect model will predict rabbit or tiger, respectively. A perfect model (or person) has *high* probability to succeed in this task. The image solves the ambiguity of the text-only MLM where multiple answers are possible.

Our work: In this work, we first establish the disadvantages of the current masking method: (1) >50% of the masked tokens are stopwords and punctuation; (3) in a third of the sentences no token is being masked therefore the MLM is not active; (3) We show that it does not require the model to look on the image enough.

We learn what is important to mask from the image perspective,[ROY: be more concrete] presenting a hierarchy of semantic classes. Using this hierarchy, we propose alternative masking strate-

[2] https://en.wikipedia.org/wiki/Rabbit

gies for MLM that require more information exchange with the image. For example, instead of masking random 15%, masking physical objects in the image will give the model a good incentive to look at the image. By that, we adjust the MLM to vision. For example, in Fig. 1, the original masking strategy will mask stopword (e.g., "is", "the", or "A") in >50% of the times. Our *Objects* model will mask an object ("tiger", or "carrot"), and our *Objects*, *Attributes*, *Relationships* will mask both an object: "tiger", an attribute: "orange", and a relationship: "eating". In a results, the model will be incented to look at the image in order to succeed. We show that our method improve results on 3 benchmarks V+L datasets (GQA, VQA, NLVR2).

In summary, our main contributions are: (1) We show that the current MLM pretraining method is sub-optimal for VLP, and it does not leverages enough of the visual aspect of the VLP task. (2) We learn what is important to mask from the image perspective, presenting a hierarchy of semantic classes. (3) Using this hierarchy, we propose alternative masking strategies for MLM that requires more information exchange with the image. We show that our method improve results on 3 benchmarks V+L datasets (GQA, VQA,NLVR2). [ROY: high level: 1. you never mention our most important finding explicitly: (50% stopwords, zero masked tokens) 2. you don't discuss our findings regarding the ranking of classes. how do we rank them? what do we find? what are the conclusions? 3. you don't present our strategies, our experiments and our results. Intro need to be significantly shorter, and contain much more of what we did].[YONATAN: Agreed, except that using the image is our most important finding. the 50% stopwords and zero masked tokens causes the image to be under-used. The SWs causes the model to be closer to a language-model and biased towards the text, where our point is to make it less biased towards the text by leaning more on the image. This is the key point of the paper.]

2 Related work

TODO: Needs to merge with the introduction

3 MLM for vision - current approach

Models and datasets selection In this paper we use the LXMERT (Tan and Bansal, 2019) model with the public implementation and analyzed its training datasets [3]. We use it because it uses the three main pre-train objectives, and it is implemented to fine-tune and test on the datasets we want to evaluate (GQA, VQA, NLVR2).

BERT text sequence length is much longer than V+L pre-train data The input sequence length in BERT is 512 tokens. Masking 15% in a 512 document results in masking 76.8 tokens in average. In LXMERT and other V+L models, the sequences are much shorter. Maximum sequence length in LXMERT is 20 tokens. Masking 15% of 20 results in 3 masked tokens. However, we measured the average lengthes of LXMERT pre-train data. The average lengths are 6.86, 5.73. Masking 15% of those lenghts results in $\sim$ 1 masked token. However, there can be cases where no token will be masked, and cases where multiple tokens will be masked.

Third of the sentences don't have masked tokens We start with observing how many tokens are masked in each sentence, using original 15% strategy, at Table 1.

# Sentences	9,181,126
# Sentences, 0 tokens were masked	3,307,857
# Sentences, 1 token was masked	3,404,331
# Sentences, > 1 token was masked	2,468,938

Table 1: Masked tokens for sentence

These statistics suggests that in a third of the cases, the model does not utilize the MLM objective for the given sentence. In addition, we are interested how the results are changed when there is more than 1 masked word, and we study it in the paper (Section 6.1).

> 50% of the masked words are stop-words or punctuation We measured the masked words in the current LXMERT pre-training procudure, and observe that 53.8% of the masked tokens are stop-words or punctuation [4].

3.1 The V+L don't need the image to succeed in the MLM pretrain task

Objective: We want to establish the claim that current V+L models are not required to look at the

[3] https://github.com/airsplay/lxmert

[4] Stop-words: NLTK and gensim postags, punctuation: string.punctuation

image enough.
Model: The published pre-trained LXMERT model from the official repository.
Task: Masked language modeling (MLM), with and without the image, using different masking strategies.
Data: LXMERT pretrain validation data (∼214K sentences)
Metrics: Exponent of the cross-entropy loss for the MLM evaluation task, and Accuracy @ 5, which is whether the label is among the top 5 most confident predictions of the model.

We evalute the published pre-trained LXMERT model on the masked language modeling task (MLM), with and without the image, while masking the tokens in different strategies:

- Original masking, where a token is masked in a probability of 15%.
- Stop-words and punctuation list
- Content words - Not stop-words nor punctuation.

With the image: The model has an access to both the text and the image (the original implementation). The model receives a masked token to predict, and it can use the text context *and* the image, in order to predict it.
Without the image: We block access to the image and use the model as single-stream model, without the co-attention layers from the image to the text. The model receives only the text and needs to complete the masked tokens.

The results are presented at Table 2.

Analysis: With the image, we can see that the model succeed in the MLM task in the Normal case (Validation loss 3.2, Acc @ 5: 89%). However, if we split it to stopwords vs. content words, we see that the high presence of stop-words & punctuation, where it achieve (Validation loss 1.5, Acc @ 5: 98%), reduces the it's task difficulty. However, in the content words masking strategy, the task is more difficult with the image (Validation loss of 9.4, Acc @ 5: 76%). Without the image, the content words masking strategy task is much harder (Validation loss of 38.7, Acc @ 5: 56%). Meaning it is much harder for the model to guess a correct word and succeed without proper information from the image. In the stop-words & punctuation case, the model can succeed very good (Validation loss of 2.9, Acc @ 5: 96%) without the image.

Conclusions: Current pre-training method does not require the model to look on the image enough. Different masking strategies such as Content words will be more challenging to the model, and the model will not be able to succeed in this task without the image. It means that the model will be incented to look at the image in order to succeed in the MLM task. We desire this property for our V+L models: We want them to be required to look at the image, and not just completing text that "makes-sense".

3.2 How good is current pre-training?

In the following section, we want to asses the gain of the current pre-training. The experiment with pre-training on 10% of the data. Results are presented at Table 3

We can see that:

- Pretrain on 10% of the data is effective on the GQA and NLVR2 datasets, but it is not effective on the VQA dataset. (R1 vs. R4)
- The MLM loss is indeed critical. Without it results descends significantly (R2).
- Pretraining is very effective in the NLVR2 case. The original LXMERT paper repoted 22% absolute gain on this dataset. With pre-training on 10% of the data, we achieve 7.48% absolute gain (R1 vs. R4).
- Pretraining on stop-words or punctuations lead to significantly lower gain than pretraining in the normal strategy. In the GQA & VQA cases, it is lower than no pretraining at all (R3 vs. R4).
- When pre-training on all of the data, pretraining is effective on all cases (R5).

4 What is important?

We begin by extracting Δ Validation loss for each word in the validation data in Section 4.1. We continue by clustering words into different groups, in order to see the hierarchy of each group's Δ Validation loss, and to understand which group will be beneficial to mask from the image perspective. We create a partition of the content words to *Objects*, *Attributes*, and *Relationships* in Section 4.2. We continue by aggregating *concreteness* measures for

	Image				Image significance	
Masking strategy	**With**		**Without**			
Metric	**Validation loss (exp)**	**Accuracy @ 5**	**Validation loss (exp)**	**Accuracy @ 5**	**Δ Validation loss (exp)**	**Δ Accuracy @ 5**
Normal	3.2	89%	8.9	79%	5.7	10%
Content words	9.4	76%	38.7	56%	29.3	20%
Stopwords & punctuation	1.5	98%	2.9	96%	1.4	2%

Table 2: Performance for the MLM task with and without the image in different masking strategies

Dataset	**GQA**	**VQA**	**NLVR2**
No pretrain	53.24	65.10	51.07
Pretraining 10% Without MLM loss	18.22	25.30	51.07
Pretraining 10% Stop words & punctuations	53.03	63.69	53.11
Pretraining 10% Current "Normal" strategy	54.40	65.06	58.55
Pretraining all data Reported LXMERT GitHub results	59.80	69.90	74.95

Table 3: Downstream task performance for limited pretraining methods

the words in the dataset in Section 4.3. We finish by presenting the full hierarchy in Section 4.4.

4.1 Extracting Δ Validation loss for each word in the validation data

In order to analyze which words will benefit from the image, we use the same Δ Validation loss metric defined in 3.1. For each sentence, we iterate each word, mask it, and predict it with the image and without the image, receiving Δ Validation loss for each word in the validation data.

4.2 *Objects*, *Attributes*, and *Relationships*

Definitions: We use the definitions of *objects*, *attributes*, and *relationships* as they were described at the Visual Genome (Krishna et al., 2017) paper.

Objects: physical objects in the image: Man, wall, child, shoe, etc.

Attributes: Attributes of physical objects. Attributes can be color (e.g. yellow), states (e.g. standing), etc.

Relationships: Relationships connect two objects together. These relationships can be actions (e.g. jumping over), spatial (e.g. is behind), descriptive verbs (e.g. wear), prepositions (e.g. with), comparative (e.g. taller than), or prepositional phrases (e.g. drive on).

In order to mask the tokens that belong to those semantic types, we first need to identify it in the sentence. It is possible to do it using scene-graph annotations as ground-truth, or by detecting it, which is approximated method. We now elaborate on the two methods.

Using the annotated scene-graph as ground truth A simple way to detect *objects*, *attributes*, and *relationships* in captions, is to obtain it if the image has scene-graph from Visual-Genome or GQA. In LXMERT pretraining data, 83% of the sentences have scene-graph annotations for it's image. For example, given the sentence, image pair: "The rabbit is eating the orange carrot", and an image, the ground truth by the scene-graph will include: *Objects*: ["rabbit", "carrot"], *Attributes*: ["orange"] *Relationships*: ["eating"]. When obtained from the scene-graph, we call it "Grounded" (Grounded objects, grounded attributes, and grounded relationships).

Predicting *objects*, *attributes*, and *relationships* in each caption: For more general and scalable method when scene-graph is not available, we can use matching heuristics. We use the Part-of-speech tagging (POS), and we aggregate lists of Objects, Attribute and Relationships from Visual Genome dataset annotations[5]. Those are our heuristics[6]:

- *Objects* are words with POS = "NOUN" and in Visual Genome objects list.
- *Attributes* are words with POS = "ADJ" and in Visual Genome attributes list.
- *Relationships* are words with POS = "ADP" or "VERB", and in Visual Genome relationships list.

Those simple rules are our predictions for detecting *Objects*, *Attributes*, and *Relationships* in a sentence.

[5] http://visualgenome.org/api/v0/api_home.html

[6] Our full code, including code to detect the semantic type tokens will be published

	# items	Accuracy	Recall
Objects	7,484,940	89.89	97.39
Attributes	3,240,096	92.91	79.91
Relationships	3,195,345	86.42	96.88

Table 4: Detection performance of *Objects*, *Attributes*, and *Relationships*

Validation of the *objects attributes* and *relationships* task: We can now evaluate the predicted *objects attributes* and *relationships* with the ground-truth obtained from the scene-graph. The grounding method (matching between the caption and the scene-graph) we use is simple - exact match between the word in the scene-graph and the caption. Using a more complex grounding algorithm will not change our predictions, but it can only improve our results (For example, if the caption has "women" that was predicted as *Object*, and the scene-graph has "woman", it is currently counted as "False-Positive" because it's not exact match). Results are presented at Table 4.

4.3 Concreteness

We hypothesis that there is a correlation between "masked words that need the image in order to be predicted" to concrete words. We believe concrete concepts will be better to mask than abstract concepts. For example, we believe that the image will have bigger benefit for predicting the word "cat" instead of the word "happy", because "cat" is more concrete word. We use a dataset of concreteness rating presented in (Brysbaert et al., 2014). This dataset covers more than 90% of the words in the pretraining dataset, where each word was annotated for concreteness, by 20-30 annotators. The rating scale is 1-5, when 1 is abstract, and 5 is concrete.

This is how they define concrete: "A concrete word comes with a higher rating and refers to something that exists in reality ; you can have immediate experience of it through your senses (smelling, tasting, touching, hearing, seeing) and the actions you do. The easiest way to explain a word is by pointing to it or by demonstrating it."

This is how they define abstract: "An abstract word comes with a lower rating and refers to something you cannot experience directly through your senses or actions. Its meaning depends on language. The easiest way to explain it is by using other words".

4.4 Hierarchy

We have the groups of Stopwords & punctuation, Content words, *objects*, *attributes*, *relationships*, grounded concepts, and concrete concepts. Given those groups, we can aggregate the Δ validation loss for each group. We present it in Fig 4. The groups are not mutually exclusive. Primary lessons we learn from this plot (with respect to the importance of the image):

- Stopwords & punctuation are the least important
- Grounded objects (objects that appear in the scene-graph) are more important than not grounded objects
- Concrete concepts are more important than abstract concepts
- Objects are important

5 What is challenging, but yet feasible?

After we learned what is important to mask in Section 4, we can ask what will be challenging the the model, but yet feasible.

Given the hierarchy, we can probe the model with different masking strategies in evaluation.

We want masking strategies that:

- Involve several semantic classes
- High Δ validation loss
- Feasible with the image
- Nearly impossible without the image

Results are presented in 5. Differently from the analysis in previous section, now we can probe the model with masking strategies that involve several masked words. For example with the sentence "The tiger is eating the orange carrot": in the "Mask all objects" strategy, we mask all of the objects in the sentence and try to predict all of them concurrently, (e.g., "The [MASK] is eating the [MASK]"). In the "Mask 1 object, 1 relationship, and 1 attribute", we choose 3 words: "The [MASK] is [MASK] the [MASK] carrot". For pretraining we chose 4 masking strategies, elaborated in next section.

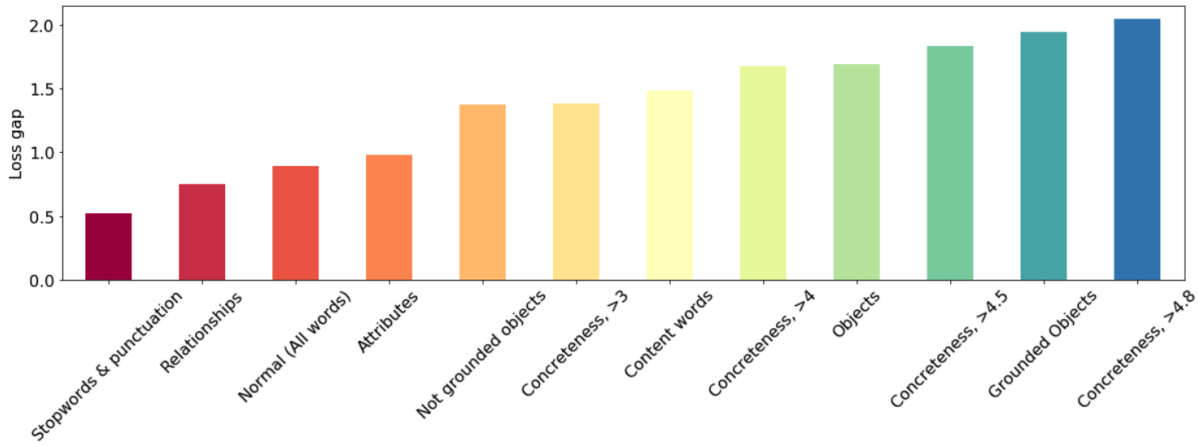

Figure 4: Hierarchy of semantic classes and it's importance by the Δ Validation loss metric (Loss without image - Loss with image).

	Image				Image significance	
Masking strategy	**With**		**Without**			
Metric	**Perplexity**	**Accuracy @ 5**	**Perplexity**	**Accuracy @ 5**	**Perplexity**	**Accuracy @ 5**
Normal	3.2	89%	8.9	78%	5.7	10%
Objects	8.9	76%	42.8	53%	33.9	23%
Stopwords & punctuation	1.5	98%	2.9	96%	1.4	2%
Content words	9.4	76%	38.7	56%	29.3	20%
Attribute	19.4	66%	63.9	47%	44.6	19%
Relationships	2.8	92%	6.6	84%	3.8	8%
Mask all objects	42.8	53%	79.8	44%	37.0	9%
Mask 1 object, 1 relationship, and 1 attribute	11.9	73%	43.8	55%	31.9	17%
Most concrete (=4.806)	8.1	77%	47.7	50%	39.6	27%

Table 5: Δ Validation loss for different masking strategies

6 Alternative masking strategies

6.1 How to change the 15% masking amount?

In Section 3 we discussed that 15% with short captions (∼6.86) causes that with third of the cases no token is masked, in another third 1 token is masked, and in the last third, multiple tokens are masked.

We isolate those factors by conducting 3 experiments:

- Not allowing 0 masked (if 0 tokens were masked, sampling 1 token to mask).
- Not allowing multiple masked (if multiple tokens were masked, sample 1 token from them to mask)
- Masking only 1 word. In this experiment, we differentiate between words and tokens: tokens are the BERT word-pieces. For example for "girrafe", the tokens are "gi", "raf", "fe". In this experiment, if we chose the word "girrafe", we later mask all of it's tokens, and only them. The motivation for this experiment, is that hiding only one word from the sentence, will give the model a "fair option" to look at the image and try to complete it. It is not too much - only 1 thing it needs to infer from the image. And it's not too easy - The model always need to infer something.

	GQA	VQA	NLVR2
Normal	54.4	65.06	58.55
Don't allow 0 masked	54.98	65.4	59.45
Don't allow multiple masked	54.46	65	58.82
Mask 1 word	55.07	65.26	61.25

Table 6: Changing 15% masking amount

Results are presented at Table 6.
We can see that not allowing multiple masked

tokens helps a bit. Not allowing 0 masked tokens helps more. And masking 1 word is the better overall strategy.

6.2 Chosen masking strategies

We chose the following masking strategies for pretraining experiments:

- *Original*: Original masking strategy as defined in the paper, 15% random token masking
- *Objects*: Choosing 1 object word to mask
- *Content words high, stop words low*: Choosing 1 word in each sentence. Instead of almost 50-50 partition between masking stopwords and content words, increase the probability to mask content word to 80% (and stopword will decrease to 20%)
- *Objects, attributes, relationship*: For the sentnce, "The tiger is eating the orange carrot", we choose object, 1 relationship, and 1 attribute, but this time instead of using [MASK], we use new masked words: [OBJ]/[ATT]/[REL] for objects/attributes/relationships (respectively). After masking, it will be: "The [OBJ] is [REL] the [ATT] carrot". Example in Fig 1. The reason for the separation is the hint the model what is the semantic class of the word to predict.

6.3 Pretraining experiments

Experimental setup We experimented with increasing amounts of pretraing data with different number of epochs and batch sizes. Full configuration details and run-times available in Table 7 in Appendix A). We pretrained the LXMERT model with each of our 4 masking strategies. Then, we finetuned the pretrained models on the downstream datasets.

Results Results are presented in Figures 5.

7 Conclusions

We established that the current MLM pretraining method is sub-optimal for Visual Language Pretraining (VLP), and is not leveraging enough of the visual aspect of the VLP task. We learned what is important to mask using the Δ loss metric on the validation dataset. We proposed alternative masking strategies for MLM that requires more information exchange with the image. We show that our pretraining method is effective and it creates models that are better suited for V+L tasks. We hope that future works with VLP will not use the traditional random 15% masking strategies that is suitable for text-only language-model training. We believe all VLP models will benefit from using alternative masking strategies. We call the community to suggest better ways to mask words, and progress towards V+L models that are less biased and better understand the visual aspect of the multimodal tasks.

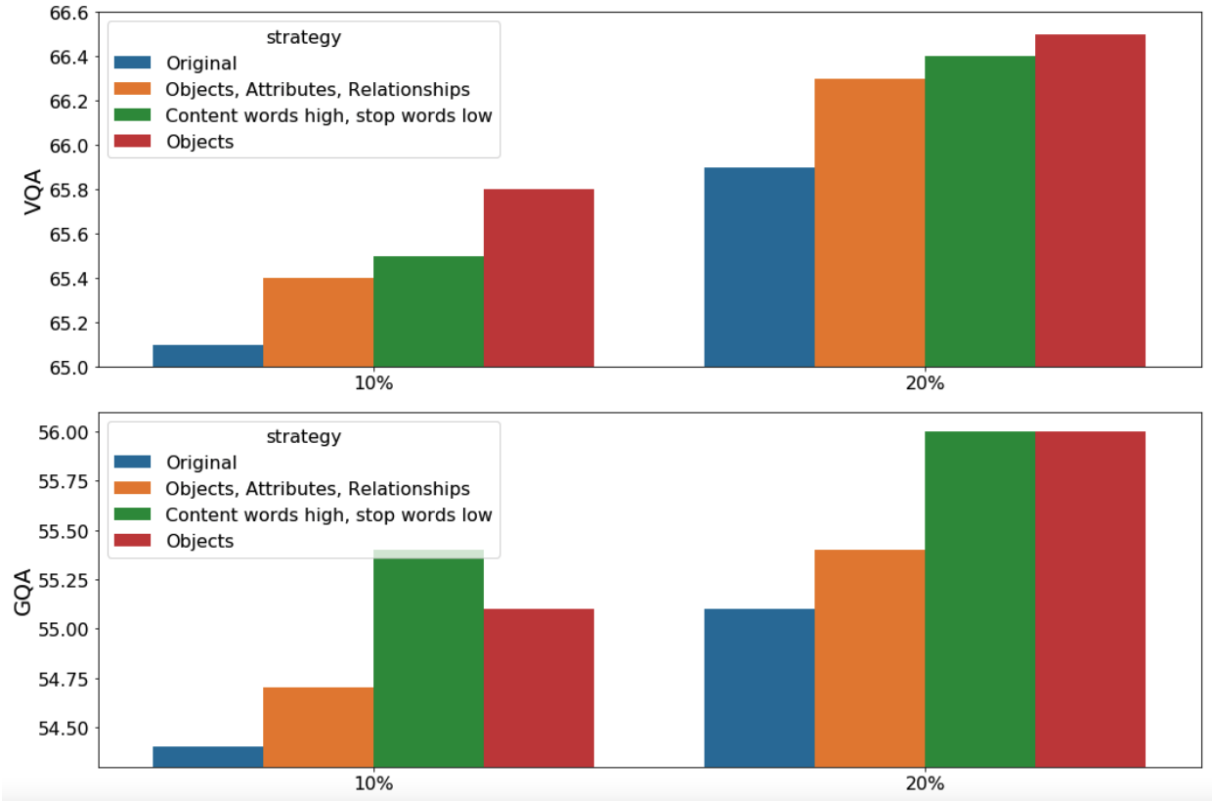

Figure 5: VQA and GQA downstream results

References

Vedika Agarwal, Rakshith Shetty, and Mario Fritz. 2020. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9690–9698.

Yonatan Bitton, Gabriel Stanovsky, Roy Schwartz, and Michael Elhadad. 2021. Automatic generation of contrast sets from scene graphs: Probing the compositional consistency of gqa. *arXiv preprint arXiv:2103.09591*.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.

Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. 2018. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 771–787.

Lisa Anne Hendricks, John Mellor, Rosalia Schneider, Jean-Baptiste Alayrac, and Aida Nematzadeh. 2021. Decoupling the role of data, attention, and losses in multimodal transformers. *arXiv preprint arXiv:2102.00529*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709.

Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. 2016. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer.

Theo Jaunet, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf. 2021. Visqa: X-raying vision and language reasoning in transformers. *arXiv preprint arXiv:2104.00926*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Chenliang Li, Ming Yan, Haiyang Xu, Fuli Luo, Wei Wang, Bin Bi, and Songfang Huang. 2021. Semvlp: Vision-language pre-training by aligning semantics at multiple levels. *arXiv preprint arXiv:2103.07829*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Yibing Liu, Yangyang Guo, Jianhua Yin, Xuemeng Song, Weifeng Liu, and Liqiang Nie. 2021. Answer questions with right image regions: A visual attention regularization approach. *arXiv preprint arXiv:2102.01916*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*.

Andrew Shin, Masato Ishii, and Takuya Narihira. 2021. Perspectives and prospects on transformer architecture for cross-modal tasks with language and vision. *arXiv preprint arXiv:2103.04037*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Jianwei Yang, Jiayuan Mao, Jiajun Wu, Devi Parikh, David D Cox, Joshua B Tenenbaum, and Chuang Gan. 2020. Object-centric diagnosis of visual reasoning. *arXiv preprint arXiv:2012.11587*.

Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. 2021. Causal attention for vision-language tasks. *arXiv preprint arXiv:2103.03493*.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5014–5022.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Making visual representations matter in vision-language models. *arXiv preprint arXiv:2101.00529*.

A Appendix

Reproducibility The experiments have been performed with the public implementations of LXMERT (Tan and Bansal, 2019) with the public implementation [7]. The experiments were performed with a NVIDIA RTX2080 GPUs.

Pretraining data	**10%**	**20%**	**50%**	**100%**
Number of epochs	7	7	13	14
Batch size	64	64	100	256
# GPUs	1	1	3	4
Time to complete	2 days	4 days	4 days	7 days

Table 7: Pretraining experiments configurations for each different masking strategy

[7] https://github.com/airsplay/lxmert