



How to Adapt Pre-trained Vision-and-Language Models to a Text-only Input?

Downloaded from: <https://research.chalmers.se>, 2024-04-19 17:31 UTC

Citation for the original published paper (version of record):

Hagström, L., Johansson, R. (2022). How to Adapt Pre-trained Vision-and-Language Models to a Text-only Input?. International Conference on Computational Linguistics . Proceedings, 29(1): 5582-5596

N.B. When citing this work, cite the original published paper.

How to Adapt Pre-trained Vision-and-Language Models to a Text-only Input?

Lovisa Hagström¹ Richard Johansson^{1,2}

¹Chalmers University of Technology, ²University of Gothenburg
{lovhag, richajo}@chalmers.se

Abstract

Current language models have been criticised for learning language from text alone without connection between words and their meaning. Consequently, multimodal training has been proposed as a way for creating models with better language understanding by providing the lacking connection. We focus on pre-trained multimodal vision-and-language (VL) models for which there already are some results on their language understanding capabilities. An unresolved issue with evaluating the linguistic skills of these models, however, is that there is no established method for adapting them to text-only input without out-of-distribution uncertainty. To find the best approach, we investigate and compare seven possible methods for adapting three different pre-trained VL models to text-only input. Our evaluations on both GLUE and Visual Property Norms (VPN) show that care should be put into adapting VL models to zero-shot text-only tasks, while the models are less sensitive to how we adapt them to non-zero-shot tasks. We also find that the adaptation methods perform differently for different models and that unimodal model counterparts perform on par with the VL models regardless of adaptation, indicating that current VL models do not necessarily gain better language understanding from their multimodal training.

1 Introduction

Having models learn language from text alone has been criticised based on several aspects, from fundamental arguments about how language works (Bender and Koller, 2020) to findings on lack of certain information in text (Gordon and Van Durme, 2013; Paik et al., 2021). To train language models on more sources than text is therefore a proposed direction for creating language models with better language understanding (Bisk et al., 2020). These models would then become multimodal, with the capability to process both text and information from other modalities.

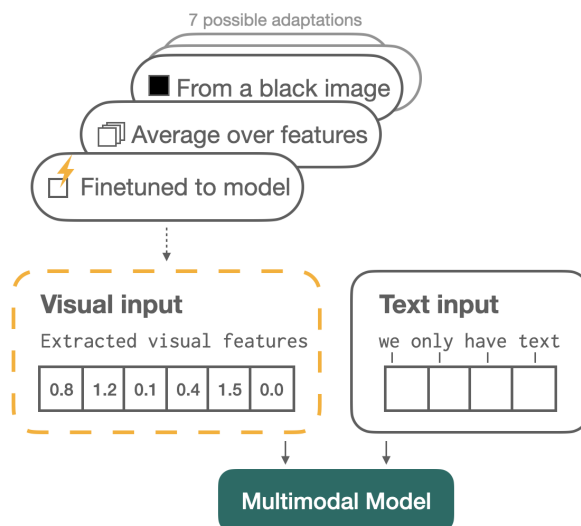


Figure 1: An overview of different ways to adapt a multimodal model to text-only input. It showcases three of the seven adaptations evaluated in this work.

The multimodal models of interest in this work are vision-and-language (VL) models that have been trained on images and their corresponding captions or visual questions (Lu et al., 2019; Tan and Bansal, 2019; Su et al., 2020; Li et al., 2019; Chen et al., 2020). These models are performant on several image-text tasks such as image captioning and VQA, while there also is an increased interest for evaluating how their natural language understanding is influenced by their multimodal training (Iki and Aizawa, 2021; Yun et al., 2021).

It is however tricky to investigate the pure natural language understanding of the aforementioned VL models, since their language processing is conditioned on visual features. For certain investigations, we may simply wish to evaluate the models on text-only domains, while these models have not been developed for this purpose. If we do not attend to the issue of accurately adapting VL models to text-only domains we risk evaluating them out-of-distribution and fail to accurately measure their natural language understanding capabilities.

Different methods for adapting VL models to a text-only input have already been tried and we have some results on the natural language understanding capabilities of these models (Iki and Aizawa, 2021; Yun et al., 2021). However, no systematic search for the best way to adapt VL models to a text-only input has been performed and it is unclear how well the VL models work with the previously proposed adaptations. If we wish to continue the search for better natural language understanding in multimodal models, we should ensure that we evaluate them in the best way possible. In this work, we search for the best method for adapting existing VL models to a text-only input, as illustrated in Figure 1.¹

With the adaptations in place, we can then compare the VL models to their unimodal text-only counterparts. This will complement already existing results on the natural language understanding capabilities of VL models and the effect of multimodal training.

The contributions of our work are as follows:

- We investigate and compare seven methods for adapting LXMERT (Tan and Bansal, 2019), VisualBERT (Li et al., 2019) and CLIP-BERT (Norlund et al., 2021) to a text-only input (Section 3). Two of these adaptations have already been used in previous investigations of the linguistic capabilities of VL models (Frank et al., 2021; Iki and Aizawa, 2021).
- We evaluate these adaptations on the GLUE benchmark (Wang et al., 2018) (Section 4.1). This gives us results on how well the adaptations work for tasks that aim to evaluate general natural language understanding.
- We also evaluate the adaptations on the Visual Property Norms (VPN) (Hagström and Johansson, 2022) (Section 4.2). This gives us results on how well the adaptations work for zero-shot tasks that aim to evaluate visual conceptual knowledge in the models.
- We compare the adapted VL models to their unimodal BERT-base (Devlin et al., 2019) counterparts on the aforementioned evaluation tasks. The ensuing results should provide additional clarity on the natural language understanding of VL models.
- We also compare the adapted VL models to the multimodal FLAVA model (Singh et al., 2022) that requires no adaptation to text-only tasks.

2 Models

We investigate adaptations to text-only input for the three multimodal models CLIP-BERT, LXMERT and VisualBERT. We also compare their results with those of a baseline BERT-base model and FLAVA. The models are further described below and an overview of them can be found in Table 1.

For each of the multimodal models, we also describe how to make the model function without visual input. This is later used in some of the adaptations we evaluate, described in Section 3.

All models evaluated in this work except for CLIP-BERT are provided by the Huggingface library (Wolf et al., 2020). The pre-trained model weights for all models except for CLIP-BERT are also provided by this library. The CLIP-BERT weights are found in our public repository.

2.1 VisualBERT

VisualBERT is a single-stream model that has been initialized from pre-trained BERT-base weights and then further trained on MS COCO as well as VQA (Lin et al., 2014; Goyal et al., 2017). As a result, it has been trained on 1.27M more texts and 0.12M more images than BERT-base. It utilizes a Faster R-CNN detector (Anderson et al., 2018) as backbone, for which it has been trained on the features of the 36 first detections, meaning that it expects visual input features with shape (36, 2048).

Usage without visual input The single-stream architecture of this model implies that it simply concatenates the embeddings from the visual features with the word embeddings from the text input and then forwards this to the BERT encoder. Therefore, this model can be queried with text only without changing anything in the model architecture, since it simply means that only the word embeddings are fed to the BERT encoder.

2.2 LXMERT

LXMERT is a dual-stream model trained on MS COCO, VQA, VG, GQA and VG-QA (Hudson and Manning, 2019; Zhu et al., 2016). It has not been initialized from BERT-base weights. In total, it has been trained on 9.18M visual texts and 0.18M

¹Code available at <https://github.com/lovhag/adapt-pre-trained-VL-models-to-text>

| Model | Size | Pre-train data | Backbone |
|------------|------|--|--------------|
| BERT-base | 110M | English Wiki, BookCorpus | - |
| FLAVA | 86M | CCNews, BookCorpus, PMD | - |
| CLIP-BERT | 110M | English Wiki, BookCorpus, CLIP-BERT V+L | CLIP |
| LXMERT | 230M | LXMERT V+L | Faster R-CNN |
| VisualBERT | 110M | English Wiki, BookCorpus, VisualBERT V+L | Faster R-CNN |

Table 1: The models evaluated with details on their pre-training data. The V+L datasets refer to model-specific VL datasets. With ‘FLAVA’ we refer to the text encoder.

images. Similarly to VisualBERT, this model expects visual features of the shape (36, 2048) from 36 Faster R-CNN detections.

Usage without visual input The dual-stream architecture of this model implies that it processes the visual embeddings and word embeddings in separate encoders before it fuses the information from them in a so called Cross-Modality Encoder. For this model it does not suffice to simply omit the visual input since it is expected by a separate visual encoder. However, the language output of the model is only affected by the visual input at a set of cross-attention sub-layers with residual connections in the Cross-Modality Encoder. Consequently, we can set the added residual from the cross-attention layer to zero and remove the visual encoder of the model.

2.3 CLIP-BERT

CLIP-BERT is a single-stream VL model that is architecturally very similar to VisualBERT. The main differences this model introduces are two, 1) it has a CLIP (Radford et al., 2021) backbone that generates visual features of dimension (512,) for each image, and 2) it has been trained on 4.72M visual texts and 2.91M images, a vision-language dataset approximately four times larger than that of VisualBERT, in addition to having been initialized from BERT-base weights.

Usage without visual input Similarly to VisualBERT, the single-stream architecture of this model implies that it can be queried with text only without changing anything in the model architecture, since it simply means that only the word embeddings are fed to the BERT encoder.

2.4 BERT-base

Since all VL models we evaluate to some extent are based on BERT-base, we use this unimodal model as a baseline in our evaluations seen in Section 4.

We also create two additional baseline versions of BERT-base by further training the pre-trained model on LXMERT text data² and a subset of the English Wikipedia corpus from the Huggingface Datasets library (Lhoest et al., 2021) sampled to match the LXMERT text data in size, respectively. We do this to enable more fair comparisons to the evaluated VL models, since they have received additional training on text and images. These model versions are denoted by `trained-LXMERT` and `trained-Wikipedia`. The unchanged BERT-base model is denoted by `default`.

Since the original LXMERT model developed by Tan and Bansal (2019) was not initialized from BERT weights, we also develop a third baseline version of BERT that has been trained from scratch on LXMERT text data for comparison. This model version is denoted by `trained-LXMERT-scratch`.

More information about the datasets used to train the BERT-base baselines and training procedures can be found in Appendices A and B respectively.

2.5 FLAVA

FLAVA is a multimodal model that works for all combinations of VL modalities without any need for adaptation (Singh et al., 2022). It sidesteps all issues related to the aforementioned VL models and can directly be evaluated for its linguistic capabilities. It consists of three separate parts: an image encoder, a text encoder, and a multimodal encoder that combines the input from the unimodal encoders. The unimodal encoders are pretrained on unimodal datasets and the full model is then trained end-to-end on the Public Multimodal Datasets (PMD) corpus (Singh et al., 2022). We use the text encoder of this model as a baseline in our evaluations.

²The data is described in <https://github.com/airsplay/lxmert>.

3 Adaptations to text-only input

There are several ways to adapt a VL model to a text-only input. In this work we investigate and compare seven possible adaptations, as described below. Two of the adaptations described here have already been used for investigating the linguistic capabilities of VL models (Frank et al., 2021; Iki and Aizawa, 2021). Common for all adaptations is that their intended use is for evaluation of a pre-trained VL model (encoder) on text-only input. When we refer to the word *adaptation* we refer to the adaptation of a VL model to text-only input.

The adaptations can be grouped into three different categories based on how they are implemented. For the first category, we simply remove the visual input to the VL model (Sections 3.1 and 3.2). For the second category, we provide the model with visual features that are constant and can be viewed as fillers, (Sections 3.3 to 3.6). For the third category, we provide the model with visual features predicted from text (Section 3.7).

3.1 Using model as-is without visual input

All VL models considered in this work can be queried with text only, or after performing a small set of alterations to the model architecture without changing any pre-trained model weights, as described in Section 2. Thus, we can directly evaluate the pre-trained models on the text-only task of interest. This adaptation is denoted by `default`.

This adaptation is very simple to apply and does not require any additional computations, while it assumes that the VL model can be queried without visual input. It is also not certain that the models will function as intended due to the imposed train/test shift of this adaptation. To our knowledge, this approach has not been tested before.

3.2 Fine-tuning model on text-only input

Before evaluating the pre-trained VL model we fine-tune it on a small text-only fine-tuning task, similarly to how several natural language understanding tasks are performed (Wang et al., 2018, 2019). The idea is that this will acclimatize the model to the aforementioned domain shift. Similarly to the `default` adaptation, this also relies on being able to use the model without visual input.

We create two separate fine-tuning sets for this adaptation. The sets have been extracted from the text part of the LXMERT training data and from English Wikipedia. Their sizes have been adapted

to match those of typical fine-tuning sets for e.g. SuperGLUE (Wang et al., 2019) and we have ensured that the number of tokens in each fine-tuning set is roughly equal. More information about the datasets can be found in Appendix A.

Finetuning the VL models on each of these sets should give us results on both the performance of the method, and on how dependent it is on the chosen fine-tuning set. These adaptations are denoted by `no-visual-features-finetuned-LXMERT` and `no-visual-features-finetuned-Wikipedia` respectively.

This method avoids having to work with image feature extractors and image data. However, it requires setting up a training algorithm and additional computations. Moreover, since the full model needs to be trained for this adaptation, it is more sensitive to hyperparameter choices. It is also not certain whether it is sufficient to perform fine-tuning on text to acclimatize VL models to a text-only input. More information on tuning procedures can be found in Appendix B.

3.3 Using averaged visual features from the training dataset

In this method we give the VL model a constant visual feature input together with the text of interest at evaluation, where the visual features are the average of all the visual features in the train or evaluation data of the model. The provided visual-features should then be kept in-distribution, while they also are uninformative. This adaptation has already been used by Frank et al. (2021) for ablating visual input to VL models. We denote it by `avg-visual-features`.

No assumptions or changes to the model architecture are necessary for this method. However, it requires access to the datasets used to train the model of interest and the computation of the averaged visual feature vector.

We calculate the averaged visual feature vector for CLIP-BERT based on the CLIP features of its training data. We also calculate the averaged visual features and position vectors for LXMERT from its corresponding training data. We take the average across training samples per detection for the LXMERT visual features such that we get one average feature vector for the first detection, another for the second detection and so forth up to the 36th.

The original released VisualBERT visual features are not compatible with the Huggingface im-

plementation of the model used in this work. We instead provide VisualBERT with the LXMERT averaged visual features, since they are compatible.

3.4 Using visual features from a black image

The idea is yet again to give the VL model a constant visual feature input together with the text of interest. In this case, the visual features are extracted from a black image using the model backbone. The model then receives a visual input similar to what it has been trained on, while it does not contain any information. This adaptation has already been used by Iki and Aizawa (2021) for evaluating e.g. VisualBERT and LXMERT on GLUE. We denote it by `zero-image-visual-features`.

Similarly to the averaged features adaptation, this adaptation makes no assumptions about the model of interest. However, it requires access to the backbone of the model and the computation of the visual features from a black image.

We use the LXMERT feature extractor to extract 36 detections with their visual features and bounding boxes from a black image. The extractor is a Faster R-CNN model developed by Anderson et al. (2018). These features are then given to LXMERT and VisualBERT during evaluation. For CLIP-BERT we use CLIP to extract visual features from the same black image.

3.5 Using constant zero vector visual features

We give the model of interest constant visual features, and the positions of bounding boxes in the case of LXMERT, that are zeros. There are no guarantees that this method will work well for adapting VL models to a text-only input. It is however easy to implement and can be seen as a baseline to be compared with the other adaptations. To the knowledge of the authors, this method has not been used previously. We denote it by `zeroed-visual-features`.

3.6 Using tuned visual features

We tune the visual features to a frozen version of the model of interest, and then use these constant features at evaluation together with the text of interest. To the knowledge of the authors, this method has not been used previously to adapt VL models to a text-only input. However, the key idea of tuning the input to the model has been used in previous works (Qin and Eisner, 2021; Tsimpoukelli et al., 2021).

We tune visual features to frozen and pre-trained versions of CLIP-BERT, VisualBERT and LXMERT respectively. We tune on the same LXMERT and Wikipedia sets used for the adaptation described in Section 3.2. More information on tuning procedures can be found in Appendix B.

This method offers more flexibility for finding the most suitable constant visual features for a VL model evaluated on text-only tasks. However, it also requires setting up the training, more computations and is more sensitive to hyperparameter tuning. We denote these adaptations on the different fine-tuning sets by `finetuned-LXMERT-visual-features` and `finetuned-Wikipedia-visual-features` respectively.

3.7 Predicting visual features from text

Some feature extractors map text representations and visual representations to the same parametric space. Consequently, they can be used to “imagine” visual features from text. The CLIP model serving as a backbone for the CLIP-BERT model has this capability and can be used to generate visual features from text during evaluation on text-only tasks. We implement it for the CLIP-BERT model and denote it by `imagined-visual-features`.

This method is quite simple to implement, while it requires access to CLIP and computing the visual features from the evaluation corpus. It is also not clear how well CLIP representations work for text that is not specifically related to visual concepts.

4 Evaluation methods

To assess the performance of our text-only adaptations, we firstly evaluate them on the GLUE benchmark, described in Section 4.1. These evaluations will give us results on how well the models and their adaptations work for general natural language understanding tasks. This benchmark has been used by both Devlin et al. (2019) and Iki and Aizawa (2021) to evaluate natural language understanding capabilities of BERT and VL models.

Furthermore, to assess the performance of the adaptations on text domains that are more focused on visual concepts, we perform evaluations on VPN, further described in Section 4.2. This will provide us with results on tasks the VL models potentially are more attuned to, complementing the general GLUE results.

| MLM query | Gold labels |
|----------------------------------|--------------|
| a cow usually is [...] | black, white |
| a mug has a [...] | handle |
| q: a greeting card has? a: [...] | pictures |

Table 2: Query samples from the VPN dataset for the concepts *cow*, *mug* and *greeting card* using three out of nine possible query templates. The [...] is typically replaced with a [MASK] token.

4.1 GLUE

The General Language Understanding Evaluation (GLUE) benchmark has the aim to evaluate model performance across several NLU tasks. It was developed by Wang et al. (2018) and has since then been used to evaluate the natural language understanding of several LMs, including BERT.

GLUE contains nine different tasks testing for grammatical correctness understanding (CoLA), sentiment classification (SST-2), semantic equivalence detection on different text domains (MRPC, QQP, STS-B), textual entailment (MNLI, RTE), answer extraction from text (QNLI) and reading comprehension (WNLI). All tasks are sentence classification tasks and have corresponding train and validation sets for fine-tuning.

VL models have already been evaluated on GLUE by Iki and Aizawa (2021) using the black image adaptation method listed in Section 3.4. We extend the GLUE evaluation to include all alternative adaptation approaches listed in Section 3.

To evaluate the performance of our adaptations on GLUE, we first fine-tune our selected multi-modal models with each adaptation on the training sets of the GLUE tasks. We then report the validation scores of the models and their adaptations. More information about the fine-tuning procedures can be found in Appendix B.

4.2 Visual Property Norms

Our current VL models are not necessarily the best fit for general NLU tasks such as GLUE (Iki and Aizawa, 2021; Yun et al., 2021). Therefore, we also evaluate them on a task we assume they are more suitable to. Visual Property Norms (VPN) essentially queries a model for the basic visual properties of a set of concepts (Hagström and Johansson, 2022). It is a text-only task, while it explicitly focuses on visual properties and concepts. Thus, if the VL models should perform particularly well on any text-only task, this would be the one. Table 2

displays examples of queries from the VPN dataset.

The VPN dataset is a zero-shot evaluation task that evaluates a model using masked language modelling (MLM), an objective our models already have been trained on. To mitigate issues with prompt-sensitivity of LMs, nine different query templates are applied during evaluation.

VPN is a version of the CSLB concept property norms dataset (Devereux et al., 2014) filtered to only contain visual conceptual features. The original property norms dataset was created with the help of 123 human participants asked to list the features of a set of concepts. Each concept has in total been exposed to 30 humans and the maximum frequency of a feature reported for a concept is then 30 and the minimum 2. This frequency is referred to as Production Frequency (PF).

VPN has been segmented into five partitions based on thresholding of PFs. We evaluate our adaptations on the segment for which $PF \geq 10$, such that ten or more annotators jointly have produced the visual features in this set. It consists of 2,001 feature entries for 621 different concepts.

5 Results

We report the evaluation results for CLIP-BERT, LXMERT and VisualBERT with the seven potential adaptations to text-only input in Figure 2. We also report the results for our four BERT-base baselines. For GLUE we report the macro-averaged score over the GLUE tasks. The score for each task is measured using its corresponding predefined metric described by Wang et al. (2018). For VPN evaluation we report the mean average precision (mAP) score averaged over each concept and relation per query template.

We also report the evaluation results on GLUE for each task and model for the best performing adaptation measured by average GLUE score in Table 3. Table 4 similarly reports the model scores on VPN for the best adaptation on average. We compare these results to those of the FLAVA text encoder that requires no adaptation. Complete numerical results can be found in Appendix C.

We format our discussion around a set of statements that can be made with respect to the results of this work, as follows.

Model performance on GLUE is more sensitive to pre-training than to adaptation Model performance on GLUE varies insignificantly between different adaptations for each model in Figure 2a.

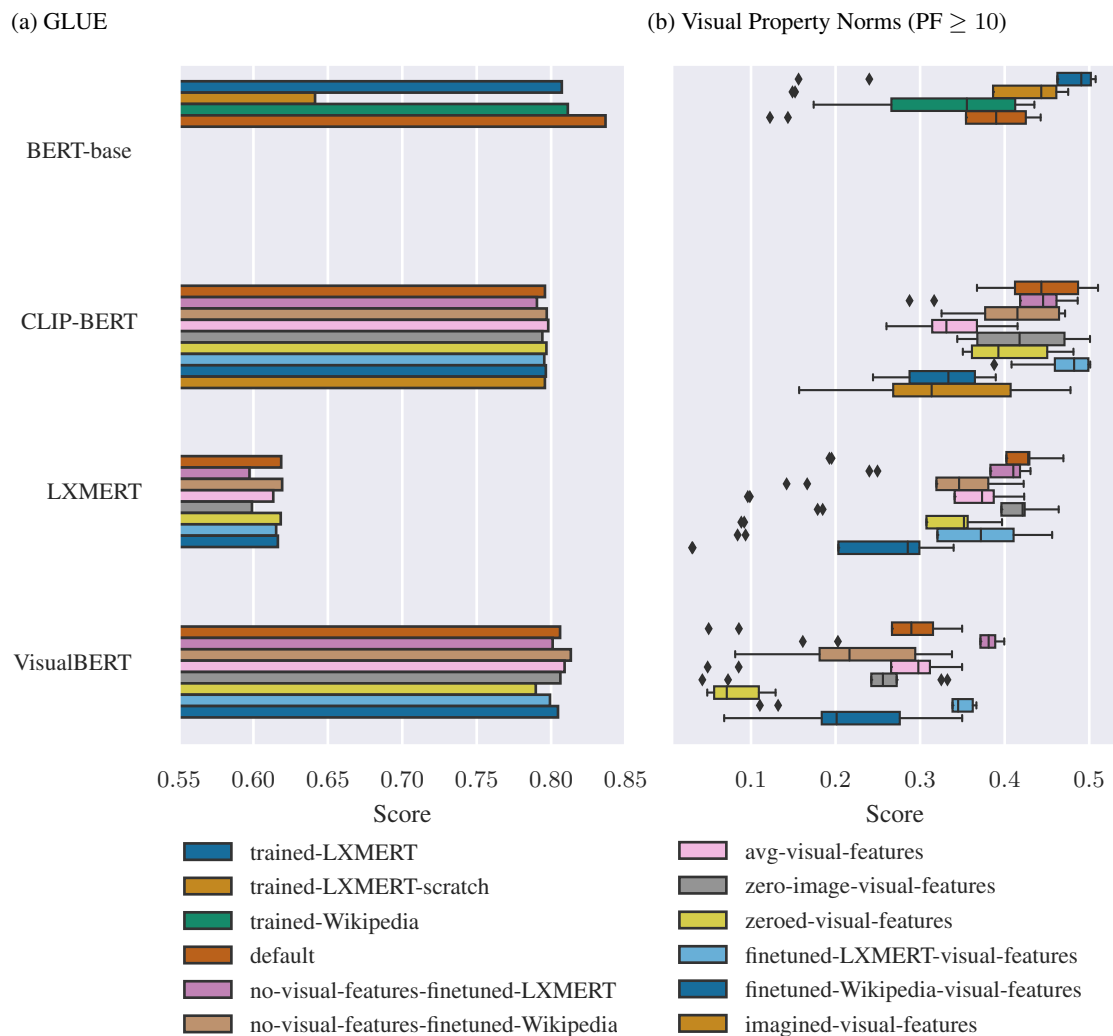


Figure 2: Results on GLUE and VPN from evaluating different adaptations to text-only input. The GLUE results are given by the mean of the scores for the development sets of all tasks, excluding WNLI. The metric used for each task is F1 score for QQP and MRPC, Matthews correlation for CoLA, Spearman correlation for STS-B, and accuracy for the remaining tasks. For VPN the box length indicates prompt sensitivity over nine different query templates.

The CLIP-BERT performance varies with less than 0.01 score points between adaptations and the VisualBERT performance with at most 0.02 score points between 'no-visual-features-finetuned-Wikipedia' and 'zeroed-visual-features'. The LXMERT performance also has a performance difference of at most 0.02 score points between 'default' and 'zero-image-visual-features'.

The largest performance difference on the GLUE benchmark can be observed between models, where LXMERT and BERT-base trained from scratch on LXMERT data perform significantly worse in comparison to the other models. Most likely, this is due to that the models were not initialized from BERT weights and consequently were not tuned to more general language usage.

A possible explanation for why the adaptation methods seem to matter so little for the GLUE results is that the benchmark is not zero-shot. The fine-tuning performed on each task might provide a sufficient of signal to the model for it to adapt to the unimodal domain.

Lastly, our results on GLUE for LXMERT and VisualBERT differ from those obtained by Iki and Aizawa (2021). Especially with respect to LXMERT for which we observe a significant performance difference compared to the other VL models, while the same cannot be observed in the results by Iki and Aizawa (2021). However, this should not raise any concerns about the robustness of the results, since we evaluated the original released models, while Iki and Aizawa (2021) eval-

| | CoLA | MNLI | MRPC | QNLI | QQP | RTE | SST-2 | STS-B |
|------------|------|------|-----------|------|-----------|------|-------|-------|
| BERT-base | 61.1 | 84.6 | 87.3/91.2 | 91.9 | 91.1/88.0 | 70.4 | 93.7 | 88.2 |
| FLAVA | 50.1 | 81.6 | 83.6/88.3 | 87.8 | 90.4/87.2 | 55.6 | 92.4 | 87.1 |
| CLIP-BERT | 55.4 | 83.2 | 75.5/84.1 | 89.8 | 91.1/88.0 | 58.1 | 92.0 | 87.8 |
| LXMERT | 15.9 | 68.1 | 69.9/81.6 | 68.0 | 84.1/76.8 | 58.5 | 86.6 | 40.1 |
| VisualBERT | 53.3 | 83.7 | 80.4/86.4 | 90.7 | 90.9/87.6 | 67.5 | 91.7 | 89.6 |

Table 3: GLUE development set results per task for the best performing adaptation on average. The best performing adaptation for each model is ‘default’ for BERT-base, ‘avg-visual-features’ for CLIP-BERT, ‘no-visual-features-finetuned-Wikipedia’ for LXMERT and ‘no-visual-features-finetuned-Wikipedia’ for VisualBERT. We report Matthew’s correlation for CoLA, average accuracy for MNLI, accuracy/F1 score for MRPC, accuracy for QNLI, accuracy/F1 for QQP and accuracy for RTE and SST-2 and Spearman correlation for STS-B.

| | VPN Score |
|------------|-----------------|
| BERT-base | 49.1 \pm 13.2 |
| FLAVA | 30.7 \pm 6.9 |
| CLIP-BERT | 48.2 \pm 4.2 |
| LXMERT | 42.8 \pm 10.8 |
| VisualBERT | 38.1 \pm 9.1 |

Table 4: VPN results for the best performing adaptations. The best performing adaptation for each model is ‘trained-LXMERT’ for BERT-base, ‘finetuned-LXMERT-visual-features’ for CLIP-BERT, ‘default’ for LXMERT and ‘no-visual-features-finetuned-LXMERT’ for VisualBERT. We report the results as median \pm standard deviation over the nine query templates.

uated models that had been unified and trained on the same data by Bugliarello et al. (2021). And as we observed previously, GLUE results are sensitive to pre-training process.

Performance on Visual Property Norms is sensitive to adaptation In contrast to the observations made for GLUE, performance on VPN differs significantly between different adaptations in Figure 2b. In contrast to GLUE, this task is zero-shot and may provide a greater challenge for models that are not sufficiently tuned to the unimodal text domain.

Additionally, it is worth noting that the model performance is quite sensitive to the choice of query template. However, this is not entirely unexpected since it has been shown that LMs are prompt-sensitive in prompt-based retrieval evaluations (Cao et al., 2021; Jiang et al., 2020).

Different adaptations perform differently for different models on Visual Property Norms For CLIP-BERT, the most suitable adaptation for evaluation on VPN is to provide the model with visual features that have been tuned on LXMERT

text data. For LXMERT, the best approach is to use the model as-is without visual input, and for VisualBERT the best adaptation is to fine-tune the model on LXMERT data without visual features. Common for all of these adaptations is that they involve some kind of prior tuning on LXMERT text data. A potential explanation for this is that the LXMERT data is more similar to VPN and results in a smaller domain shift.

An explanation for the varying adaptation fits between VL models is potentially found by looking at the different pre-training datasets and architectures of the models. VisualBERT has been tuned on much less data compared to the other models, and may therefore benefit from more training in general. Additionally, the single-stream CLIP-BERT and VisualBERT models process all linguistic and visual information in a joint manner, without the same ability to disentangle signals as the dual-stream LXMERT model.

FLAVA does not outperform adapted VL models On both GLUE and especially VPN, FLAVA is not better than the adapted VL models. This contrasts the GLUE results reported for FLAVA and other VL models by Singh et al. (2022). The difference in results may arise from differences in fine-tuning methods for GLUE and that we do not evaluate unified VL models.

Based on our results, adapting VL models to text-only input works better or equal to developing a model to work for all modalities from the start, as was done for FLAVA. However, since all models evaluated have been trained on different datasets with different objectives we cannot draw certain conclusions related to model design.

BERT-base baselines outperform vision-and-language models regardless of adaptation

Lastly, we can observe in Figure 2 how default BERT-base and BERT-base trained on LXMERT text data each have among the best performances on GLUE and VPN respectively. For GLUE it is expected: Iki and Aizawa (2021) have already observed that VL models on average perform worse on GLUE. Yun et al. (2021) also found similar results when they compared the quality of the linguistic representations of VisualBERT to those of BERT-base. The general natural language understanding capabilities evaluated in GLUE are potentially not easy to learn from a visual modality, explaining why the VL models did not perform better on this task. Our results on VPN are perhaps more surprising.

From their visual training, the VL models should more easily have gained natural language understanding capabilities necessary for better performance on VPN. Three potential explanations for why a LM still outperforms a VL model on VPN are 1) BERT-base is better tuned and therefore has a better overall performance, 2) the VL models evaluated in this work do not learn more about visual concepts from images compared to text that has been curated to contain visual information, or 3) the VPN task does not accurately measure the visual conceptual information we have in mind. More investigations are necessary to accurately determine the reason. In support of explanation (2), Abdou et al. (2021) found that there are similarities between color representations in LMs and actual perceptual color spaces, indicating that visual perceptual information may be found in text.

We should also note that none of the models evaluated in this work were developed with the goal of achieving better natural language understanding by multimodal training. This potentially explains some of our results, and provides an interesting avenue for future research in developing models that have a better performance on both unimodal and multimodal tasks.

6 Related Work

As previously mentioned, Iki and Aizawa (2021) have already looked at the language understanding capabilities of VL models, while they only looked at one way of adapting these models to a text-only input and only evaluated on GLUE. Yun et al. (2021) also look at the language understanding capabilities of VL models by evaluating the linguistic representations of VisualBERT and com-

pare them to a BERT-base model that has been trained on the same text data. In contrast to their work, we investigate several VL models and evaluate their performance on language generation tasks.

Bugliarello et al. (2021), Hessel and Lee (2020), Thrush et al. (2022) and Frank et al. (2021) also perform extensive evaluations of several VL models such as LXMERT and VisualBERT. In contrast to our work, they primarily focus on the VL performance of the models, and do not consider the model performance on text-only input.

Tan and Bansal (2020) introduce a new method for enriching the textual representations of a model by training on visual information. Their method results in a model that can be directly applied to text-only tasks and outperforms its standard BERT model counterpart on GLUE. This method provides a parallel research avenue compared to adapting VL models to text-only input.

7 Conclusions

We have investigated and compared seven possible adaptations of CLIP-BERT, LXMERT and VisualBERT to text-only input by evaluation on GLUE and Visual Property Norms. We can conclude that care should be put into adapting these pre-trained VL models to text-only input for better performance on zero-shot tasks, while the choice of adaptation method seems to be less impactful on tasks coupled with fine-tuning sets.

Finally, we have observed that a unimodal LM has a performance on text-only tasks that is better or comparative to that of its VL model counterparts, regardless of how these counterparts were adapted to text-only input. Seemingly, improved pure text capabilities are not guaranteed from simply training a model on arbitrary multimodal tasks. This agrees with and solidifies previous research results on VL models.

Acknowledgements

We would like to thank the reviewers for their valuable feedback and advice on related work.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC), partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

References

- Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. 2021. [Can language models encode perceptual structure without grounding? a case study in color](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 109–132, Online. Association for Computational Linguistics.
- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. [Bottom-up and top-down attention for image captioning and visual question answering](#). In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086.
- Emily M. Bender and Alexander Koller. 2020. [Climbing towards NLU: On meaning, form, and understanding in the age of data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.
- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. [Experience grounds language](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online. Association for Computational Linguistics.
- Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. 2021. [Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language BERTs](#). *Transactions of the Association for Computational Linguistics*, 9:978–994.
- Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. [Knowledgeable or educated guess? revisiting language models as knowledge bases](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Uniter: Universal image-text representation learning](#). In *ECCV*.
- Barry J Devereux, Lorraine K Tyler, Jeroen Geertzen, and Billi Randall. 2014. The centre for speech, language and the brain (cslb) concept property norms. *Behavior research methods*, 46(4):1119–1127.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. [Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9847–9857, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jonathan Gordon and Benjamin Van Durme. 2013. [Reporting bias and knowledge acquisition](#). In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30.
- 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lovisa Hagström and Richard Johansson. 2022. [What do models learn from training on more than text? measuring visual commonsense knowledge](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 252–261, Dublin, Ireland. Association for Computational Linguistics.
- Jack Hessel and Lillian Lee. 2020. [Does my multimodal model learn cross-modal interactions? it’s harder to tell than you might think!](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 861–877, Online. Association for Computational Linguistics.
- Drew A Hudson and Christopher D Manning. 2019. [Gqa: A new dataset for real-world visual reasoning and compositional question answering](#). *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Taichi Iki and Akiko Aizawa. 2021. [Effect of visual extensions on natural language understanding in vision-and-language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2189–2196, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How Can We Know What Language Models Know?](#) *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Guntjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas

- Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. [Datasets: A community library for natural language processing](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- 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](#). *CoRR*, abs/1908.03557.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. 2014. [Microsoft coco: Common objects in context](#). In *ECCV*. European Conference on Computer Vision.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Tobias Norlund, Lovisa Hagström, and Richard Johansson. 2021. [Transferring knowledge from vision to language: How to achieve it and how to measure it?](#) In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–162, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Cory Paik, Stéphane Aroca-Ouellette, Alessandro Roncone, and Katharina Kann. 2021. [The World of an Octopus: How Reporting Bias Influences a Language Model’s Perception of Color](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 823–835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Guanghai Qin and Jason Eisner. 2021. [Learning how to ask: Querying LMs with mixtures of soft prompts](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#).
- Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. [FLAVA: A foundational language and vision alignment model](#). In *CVPR*.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. [Vi-bert: Pre-training of generic visual-linguistic representations](#). In *International Conference on Learning Representations*.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.
- Hao Tan and Mohit Bansal. 2020. [Vokenization: Improving language understanding with contextualized, visual-grounded supervision](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080, Online. Association for Computational Linguistics.
- Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. [Winoground: Probing vision and language models for visio-linguistic compositionality](#). *arXiv preprint arXiv:2204.03162*.
- Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. [Multimodal few-shot learning with frozen language models](#). *Proc. Neural Information Processing Systems*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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.
- Tian Yun, Chen Sun, and Ellie Pavlick. 2021. [Does vision-and-language pretraining improve lexical](#)

grounding? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4357–4366, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

A Datasets for training and tuning

More detailed information on the datasets used for training and fine-tuning the models investigated in this work can be found here.

A.1 Training of BERT baselines

More information about the LXMERT and Wikipedia training datasets used for the BERT-base baselines can be found in Table 5. By training the BERT model on LXMERT text data, it will have seen the same textual information as LXMERT. And by training it on the Wikipedia data, it will have seen the same amount of text as LXMERT.

A.2 Fine-tuning model on text-only input

The LXMERT and Wikipedia datasets used for fine-tuning text-only versions of VL models are further described in Table 6. The two fine-tuning sets cover quite different domains. This is already visible from the tokens/sample count in the table, in which the Wikipedia corpus generally contains long sentences and the LXMERT corpus generally contains shorter sentences more suitable for image captions.

B Training procedures

More detailed information on our training procedures can be found here.

B.1 Training BERT-base baselines

For the training of BERT-base on both the LXMERT and Wikipedia datasets we use an MLM objective, a batch size of 16,384 and learning rate 5×10^{-5} until the model performance on the dev set had converged. The maximum training time was at most 23 hours on 32 Tesla T4 GPUs.

B.2 Fine-tuning model on text-only input

We fine-tune the models using an MLM objective, batch size of 256 and learning rate 5×10^{-5} until the model performance on the dev set had converged. The maximum training time was two hours on eight Tesla T4 GPUs.

B.3 Using tuned visual features

We tune the visual features using an MLM objective, batch size 64 and a learning rate of 0.05 until the model performance had converged on the dev set. The maximum training time was 18 hours on one Tesla T4 GPU.

B.4 Fine-tuning on GLUE

For the GLUE fine-tuning, we tune our models for four epochs with a learning rate of 3×10^{-5} , weight decay of 0.01 and batch size of 32. The longest tuning time was four hours on two A100 GPUs. We then pick the model checkpoint with the best validation score during training for evaluation.

C Complete numerical results

The complete numerical results on GLUE and VPN can be viewed in Tables 7 and 8 respectively.

| Corpus | Partition | # of samples | # of tokens | # of tokens/sample |
|-----------|-----------|--------------|-------------|--------------------|
| LXMERT | train | 9.0M | 59.0M | 6.6 |
| | dev | 0.2M | 1.4M | 6.8 |
| Wikipedia | train | 4.4M | 59.0M | 13.4 |
| | dev | 0.1M | 1.3M | 13.4 |

Table 5: The two text datasets used for developing two additional BERT-base baselines. The number of samples are roughly equal to the number of sentences for these datasets. The LXMERT data is the text part of the LXMERT training data. Wikipedia is a subset of general English Wikipedia texts that has been adapted to match the LXMERT data in total number of tokens.

| Corpus | Partition | # of samples | # of tokens | # of tokens/sample |
|-------------|-----------|--------------|-------------|--------------------|
| LXMERT-f | train | 9,500 | 63,000 | 6.6 |
| | dev | 3,300 | 22,000 | 6.6 |
| Wikipedia-f | train | 4,600 | 63,000 | 13.7 |
| | dev | 1,600 | 22,000 | 13.5 |

Table 6: The two text datasets used for fine-tuning, denoted by the “-f” ending. The number of samples are roughly equal to the number of sentences for these datasets.

| Model | Adaptation | Score |
|-------------------------------------|--|-------|
| BERT-base | trained-LXMERT | 80.7 |
| | trained-LXMERT-scratch | 64.1 |
| | trained-Wikipedia | 81.1 |
| | default | 83.7 |
| FLAVA | default | 78.8 |
| CLIP-BERT | default | 79.6 |
| | no-visual-features-finetuned-LXMERT | 79.0 |
| | no-visual-features-finetuned-Wikipedia | 79.7 |
| | avg-visual-features | 79.8 |
| | zero-image-visual-features | 79.4 |
| | zeroed-visual-features | 79.7 |
| | finetuned-LXMERT-visual-features | 79.5 |
| | finetuned-Wikipedia-visual-features | 79.6 |
| LXMERT | default | 61.9 |
| | no-visual-features-finetuned-LXMERT | 59.7 |
| | no-visual-features-finetuned-Wikipedia | 61.9 |
| | avg-visual-features | 61.3 |
| | zero-image-visual-features | 59.9 |
| | zeroed-visual-features | 61.8 |
| | finetuned-LXMERT-visual-features | 61.5 |
| | finetuned-Wikipedia-visual-features | 61.6 |
| VisualBERT | default | 80.6 |
| | no-visual-features-finetuned-LXMERT | 80.1 |
| | no-visual-features-finetuned-Wikipedia | 81.3 |
| | avg-visual-features | 80.9 |
| | zero-image-visual-features | 80.6 |
| | zeroed-visual-features | 79.0 |
| | finetuned-LXMERT-visual-features | 79.9 |
| finetuned-Wikipedia-visual-features | 80.5 | |

Table 7: The adaptation and baseline results for GLUE seen in Figure 2a.

| Model | Adaptation | Score | |
|------------|--|--------|--------------------|
| | | Median | Standard deviation |
| BERT-base | trained-LXMERT | 49.1 | 13.2 |
| | trained-LXMERT-scratch | 44.3 | 13.2 |
| | trained-Wikipedia | 35.5 | 9.5 |
| | default | 39.0 | 12.4 |
| FLAVA | default | 30.7 | 6.9 |
| CLIP-BERT | default | 44.3 | 4.7 |
| | no-visual-features-finetuned-LXMERT | 44.5 | 7.0 |
| | no-visual-features-finetuned-Wikipedia | 41.5 | 5.4 |
| | avg-visual-features | 33.1 | 5.0 |
| | zero-image-visual-features | 41.8 | 5.8 |
| | zeroed-visual-features | 39.3 | 4.9 |
| | finetuned-LXMERT-visual-features | 48.2 | 4.2 |
| | finetuned-Wikipedia-visual-features | 33.3 | 4.8 |
| LXMERT | imagined-visual-features | 31.4 | 10.2 |
| | default | 42.8 | 10.8 |
| | no-visual-features-finetuned-LXMERT | 41.0 | 7.5 |
| | no-visual-features-finetuned-Wikipedia | 34.6 | 10.0 |
| | avg-visual-features | 37.3 | 12.9 |
| | zero-image-visual-features | 42.1 | 11.0 |
| | zeroed-visual-features | 35.2 | 12.0 |
| | finetuned-LXMERT-visual-features | 37.2 | 13.9 |
| VisualBERT | finetuned-Wikipedia-visual-features | 28.5 | 11.7 |
| | default | 29.0 | 10.9 |
| | no-visual-features-finetuned-LXMERT | 38.1 | 9.1 |
| | no-visual-features-finetuned-Wikipedia | 21.6 | 9.0 |
| | avg-visual-features | 29.8 | 11.0 |
| | zero-image-visual-features | 25.6 | 10.2 |
| | zeroed-visual-features | 7.1 | 3.1 |
| | finetuned-LXMERT-visual-features | 34.5 | 10.3 |
| | finetuned-Wikipedia-visual-features | 20.1 | 9.9 |

Table 8: The adaptation and baseline results for VPN.