Visual Definition Modeling: Challenging Vision & Language Models to Define Words and Objects

Bianca Scarlini* [1], **Tommaso Pasini*** [2], **Roberto Navigli**[1]

[1] Sapienza NLP Group, Sapienza University of Rome
[2] Department of Computer Science, University of Copenhagen
scarlini@di.uniroma1.it, tommaso.pasini@di.ku.dk, navigli@diag.uniroma1.it

Abstract

Architectures that model language and vision together have received much attention in recent years. Nonetheless, most tasks in this field focus on end-to-end applications without providing insights on whether it is the underlying semantics of visual objects or words that is captured. In this paper we draw on the established Definition Modeling paradigm and enhance it by grounding, for the first time, textual definitions to visual representations. We name this new task Visual Definition Modeling and put forward DEMETER and DIONYSUS, two benchmarks where, given an image as context, models have to generate a textual definition for a target being either i) a word that describes the image, or ii) an object patch therein. To measure the difficulty of our tasks we finetuned six different baselines and analyzed their performances, which show that a text-only encoder-decoder model is more effective than models pretrained for handling inputs of both modalities concurrently. This demonstrates the complexity of our benchmarks and encourages more research on text generation conditioned on multimodal inputs. The datasets for both benchmarks are available at https://github.com/SapienzaNLP/visual-definition-modeling as well as the code to reproduce our models.

Introduction

Two of the most important building blocks of human cognition and Artificial Intelligence are text and image understanding, which are widely known, respectively, as Natural Language Processing and Image Processing. Recently, these two worlds have joined forces and modality-specific models are losing ground to more hybrid architectures that were developed to leverage both textual and visual data (Su et al. 2019; Tan and Bansal 2019). In fact, these models showed that the key to achieving more comprehensive reasoning skills lies in exploiting both sources of information at the same time. However, most multimodal tasks aim at investigating the high-level understanding[1] capabilities of a model, such as Visual Semantic Role Labeling (Yatskar, Zettlemoyer, and Farhadi 2016) and Image Captioning (Lin et al. 2014). Indeed, while being apt at showing what kind of patterns models can capture, these tasks do not provide any fine-grained information about how models "perceive" image and text components (objects and words). In fact, the semantic understanding of these aspects is often overlooked, i.e., objects are just classified with coarse-grained classes (Zhang et al. 2013) and words have been investigated by categorizing them according to a predefined discrete sense inventory (Gella, Lapata, and Keller 2016; Gella, Elliott, and Keller 2019). Therefore, while we may have a picture reflecting the extent to which multimodal architectures perform, we still do not know how these approaches cope with a more in-depth understanding of words and objects in context.

(a) A long walk usually for exercise or pleasure.

(b) An aircraft that has a fixed wing and is powered by propellers or jets.

Figure 1: An excerpt from the DEMETER (left) and DIONYSUS (right) datasets.

In this paper, with the goal of shedding some light on these aspects, we draw inspiration from the recent Definition Modeling task (Noraset et al. 2017), where a word in context has to be associated with a textual definition of the concept it represents, and shift it to a multimodal perspective. Hence, we propose the new task of Visual Definition Modeling (VDM), where a model is required to generate a definition for i) a given word representative of the concept depicted in an input image (Figure 1a), or ii) a visual object

*The authors contributed equally.

[1] In line with Bender and Koller (2020), in this paper we do not refer to the human process of "understanding", but rather to the ability of a machine to draw conclusions based on data, albeit multimodal.

therein (Figure 1b). For the first of these two settings we introduce the DEMETER (DEfining Multimodally-contExtualized TERms) dataset, whereas for the second we put forward the DIONYSUS (DefIning Objects Narrowed bY viSUal Subregions) dataset. To briefly illustrate how these benchmarks work, given the context image in Figure 1a and the word *"hike"* (DEMETER dataset), or the visual context in Figure 1b and the outlined visual object (DIONYSUS dataset), a model has to generate definitions such as *"a long walk usually for exercise or pleasure"* and *"an aircraft that has a fixed wing and is powered by propellers or jets"*, respectively.

Furthermore, we also introduce six baselines relying on state-of-the-art vision-and-language and text-only models and test them on both our benchmarks, showing the impact that a pretrained decoder can have on the final performance.

To summarize, our contributions are threefold:

1. **Visual Definition Modeling**: we extend the Definition Modeling task to the visual modality and design a task to probe the semantic understanding of neural models;
2. **DEMETER** and **DIONYSUS**: two new benchmarks aimed at measuring the ability of neural models to define visual and word objects in different settings.
3. **Experimental Analysis**: an extensive experimental analysis, presenting various baselines for both our tasks and an evaluation of the capability of a pretrained text-only model to deal with multimodal inputs.

Related Work

Models for vision and language can mainly be divided into two different classes: *single-* and *dual-* stream architectures. While single-stream models process both visual and language inputs through a single transformer architecture (Su et al. 2019), dual-stream models first treat the two inputs separately with different transformer layers, then apply a mechanism of intra-modality attention to fuse the two representations (Tan and Bansal 2019; Lu et al. 2019). Unfortunately, while some effort has recently been put into unifying the evaluation of vision-and-language models (Bugliarello et al. 2020), the test bed of these architectures is still scattered across different benchmarks. These datasets cover both basic tasks, where models need to find and tag objects with a predefined set of labels (Plummer et al. 2015, Flickr30k), as well as more complex generation tasks, such as producing a description of a picture (Lin et al. 2014, MSCOCO), answering questions regarding a given image (Goyal et al. 2017; Hudson and Manning 2019, VQA2, GQA) or generating expressions referring to a visual object (Yu et al. 2016, RefCOCO).

Over the years many other classification tasks have been proposed, such as Visual Entailment (Xie et al. 2019, SNLI-VE), that requires models to tag a sentence as either entailed, neutral or contradictory with respect to the premise that is presented as an image, and the Visual Reasoning benchmark (Suhr et al. 2019, NLVR^2) that asks models whether a natural language statement is true given a pair of pictures as context. Many of these tasks require architectures to have a high-level "understanding" of the image, e.g., capturing the situation that is depicted and generating a description (MSCOCO), or having knowledge of the objects and their relations within the picture in order to answer questions (VQA). However, none of the proposed tasks aims at investigating the extent to which vision-and-language models capture the semantics of words and visual objects, that is, from these tasks, we cannot infer if a model knows what a given image patch or word represents.

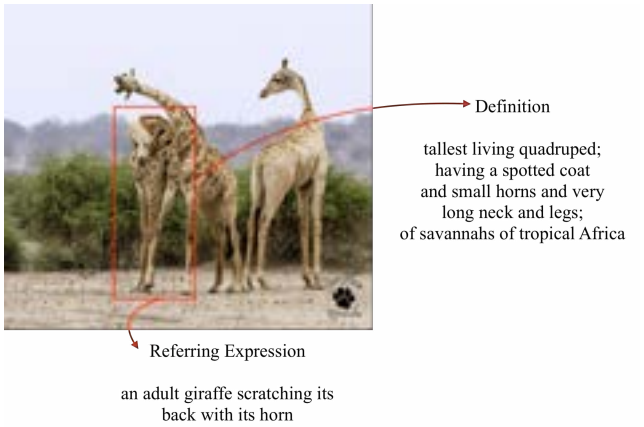

Figure 2: Example of a referring expression from RefCOCO (Yu et al. 2016) compared to a definition from the DIONYSUS dataset given the same visual object.

Recently, Noraset et al. (2017) introduced the task of Definition Modeling (DM). This benchmark challenges a model to generate a definition for a given word in context, and is thus arguably suitable for measuring the level of "comprehension" of word semantics. DM is useful for second language learners and has also been leveraged as an alternative and more flexible method for performing Word Sense Disambiguation (Bevilacqua, Maru, and Navigli 2020). We found the task of Definition Modeling particularly suited for measuring the capability of systems to model semantics and thus, for the first time, we transfer this paradigm to the multimodal context. Differently from RefCOCO, our new DIONYSUS dataset requires a definition of the framed object to be provided and not for its actions within the specific image to be described (see Figure 2). Another task related to ours is Visual Word Sense Disambiguation (V-WSD), where an image has to be tagged with a label representing its meaning in English (Saenko and Darrell 2009; Gella, Lapata, and Keller 2016) and in different languages (Gella, Elliott, and Keller 2019). While similar as regards the objective of analyzing the semantics of a picture, our tasks differ substantially differ from V-WSD. In fact, we require models to generate a definition for a word or an object given a visual context, rather than to output a class drawn from a finite inventory given an image as input. This allows us to dispose of the dependency on a predefined and discrete inventory, hence enabling our task to be not only more flexible, but also easily extendable.

Visual Definition Modeling

Visual Definition Modeling (VDM) is a new task that transfers the Definition Modeling (Noraset et al. 2017) paradigm from a textual to a visual context. We define VDM as the

task of generating a definition for a word or an object patch, given a visual context. Within this setting, we designed the two VDM datasets of DEMETER and DIONYSUS, that require models to generate an English definition for a concept represented by either a word or an object in a visual context, respectively.

DEMETER

We now introduce the first of the two Visual Definition Modeling benchmarks, DEMETER. First, we define the goal and organization of the dataset. Then, we describe its creation and provide relevant statistics of its training, development and test sets.

Dataset Definition The goal of the DEMETER dataset is, given a multimodal pair made up of i) an image and ii) a word or multi-word that describes it, to provide a textual definition of the concept brought about by the input (see Figure 1a). Note that the instances of our DEMETER dataset are not only associated with concepts that refer to concrete and physical things, e.g., *dog, airplane*, but also to non-concrete and abstract entities, e.g., *sadness, smell*.

This benchmark helps to evaluate the capabilities of a system to create meaningful connections between words and images, and hence to test their "understanding" of concepts in a more general and interpretable manner. In fact, generating a textual definition allows us, first, to remove any dependency on a discrete list of labels, and second, to have an output that is more informative and of clearer interpretation than a single category.

Dataset Creation In order to ensure a good coverage of both abstract and concrete concepts, we built DEMETER from the BabelPic[2] (Calabrese, Bevilacqua, and Navigli 2020) and ImageNet[3] (Russakovsky et al. 2015) repositories. BabelPic provides an image-meaning association for a subset of the concepts in BabelNet (Navigli and Ponzetto 2012; Navigli et al. 2021), i.e., a large multilingual semantic network providing lexicalizations of concepts in different languages. Each concept in BabelPic has been either manually or automatically linked to multiple images that are representative of it. Similarly to BabelPic, ImageNet provides different images for a subset of the concepts in WordNet (Miller et al. 1990), i.e., the most used English lexical knowledge base for Natural Language Processing applications. While BabelPic provides images for both abstract and concrete concepts, ImageNet focuses on physical and tangible objects.

For each concept in these two repositories, we retrieved their corresponding lemmas and definitions in WordNet[4]. In fact, each concept in WordNet comes with a human-made definition and a set of synonyms that express the target concept. Thus, each instance in the DEMETER dataset comprises: i) either 5 (for training) or 1 (for validation and test) randomly-sampled images representing the target concept, ii) a lemma that describes the target concept, iii) a definition for the target concept. We reserve all instances with 5 images for training, so as to test the capability of models to generalize and avoid overfitting on a single image per concept. As for the validation and test instances, instead, we retain only one image in order to make the evaluation straightforward. The entries in the test dataset are gold, as we derived them from the manually-annotated parts of BabelPic and ImageNet.

As a result of this process, the total number of instances in the DEMETER dataset is 90,880. We adopted a 80%, 10%, 10% train, validation and test split, resulting in 72,704, 9,088 and 9,088 instances, respectively. These figures along with number of distinct words, images and other in-depth statistics are reported in Table 1.

	Train	Val	Test
instances	72,704	9,088	9,088
words	65,734	8,951	8,931
images	184,981	9,088	9,088
avg. words per concept	1.7	1.0	1.0
avg. images per concept	4.0	1.0	1.0

Table 1: Statistics on the training validation and test splits of the DEMETER dataset.

DIONYSUS

We now describe the second Visual Definition Modeling benchmark, namely, DIONYSUS. First, we describe its organization and define the benchmark itself. Then, we present how the dataset was created along with reporting relevant statistics of its data. Finally, we outline the process of annotation undertaken for building the dataset.

Dataset Definition The goal of DIONYSUS is to produce a definition given an image and a visual object representing a concept therein. The visual input consists of: i) an image and ii) a subregion of this image highlighting a specific object. The target annotation of the instance, instead, is a textual definition of the object in the visual context, as depicted in Figure 1b. Differently from DEMETER, DIONYSUS's input is entirely visual, because the focus of this benchmark is to test the ability of models to understand the semantics behind visual objects in an image, without leaving any room for ambiguity as can be the case in Image Captioning. For example, the image in Figure 3 is associated with the ambiguous caption *"an image of a keyboard with a mouse next to it"* where the word *mouse* in the given textual context can refer to either the electronic device or the animal. In our dataset, the visual region depicting the *mouse* object is, instead, associated with the definition *"a hand-operated electronic device that controls the coordinates [...]"*, which clearly describes the correct meaning of the word *mouse*.

Both DIONYSUS and DEMETER contain a number of images and instances that are in line with other datasets in the

[2]https://sapienzanlp.github.io/babelpic/

[3]https://image-net.org/download. We exploited the ImageNet data available for the ILSVRC challenge as, at the time of performing the experiments, the full ImageNet repository was not available for download.

[4]All the concepts in BabelPic can be linked to a corresponding concept in WordNet thanks to BabelNet.

Figure 3: Example of an equivocal caption compared to a non-ambiguous definition in the DIONYSUS dataset.

	Train	Val	Test
instances	111,119	6,619	4,824
objects	74	74	74
images	55,689	3,353	1,893
avg. subregions per concept	1,501	89	65
avg. images per concept	1,079	65	48

Table 2: Statistics on the training, validation and test splits of the DIONYSUS dataset.

literature, e.g., Visual Commonsense Reasoning (Zellers et al. 2019) (290K instances / 110K images) and Visual Semantic Role Labeling (Gupta and Malik 2015) (16K instances / 10K images).

Dataset Creation In order to create the DIONYSUS dataset we leveraged the information within MSCOCO training data (Lin et al. 2014)[5]. MSCOCO is a large-scale object classification, segmentation and image captioning dataset, and images therein are associated with multiple captions and a list of bounding boxes. Each bounding box delimits a subregion of the input image that is annotated with a coarse-grained label giving a rough description of the object therein[6].

Our goal was to associate each object subregion of MSCOCO images with its corresponding definition. To this end, we manually annotated the 80 MSCOCO object categories with a WordNet concept and definition[7]. We note that each object category in MSCOCO is implicitly unambiguous, that is, a category is always referred to the same object from a semantic point of view. For example, the category named *mouse* is always associated with a region portraying the mouse as an electronic device and never as an animal.

Once every object category had been associated with a definition, we needed to identify the subregions within each image where an object appeared. While MSCOCO already provides object bounding boxes, most vision-and-language models rely on object features extracted by the Faster R-CNN model (Ren et al. 2015)[8], which automatically detects 36 boxes and computes their representations. Thus, to facilitate other vision-and-language models to perform this task, we needed to match MSCOCO object boxes with those extracted by the Faster R-CNN. To this end, we processed all MSCOCO training images with Faster R-CNN and discarded all the bounding boxes that did not match those extracted by the Faster R-CNN within a window of 50 pixels.

At the end of this process, every image in the MSCOCO training dataset had been associated with a list of subregions overlapping with those extracted through Faster R-CNN, each of which was annotated with a coarse-grained label thanks to MSCOCO, and to a definition, thanks to our annotation process. The final training, validation and test splits of DIONYSUS were created by using the extensively used Karpathy split (Karpathy and Fei-Fei 2017). All the subregions we extracted for an image of the Karpathy's training split are part of the training set of the DIONYSUS dataset[9]. The same line of reasoning applied also for the validation and test data. We decided to randomly select a maximum number of instances per object category for each dataset, that we set to 5,000 so as to balance the distribution of object labels and not skew it towards only a small subset of categories, such as *person*. Finally, the instances of the test data were manually validated by annotators since the bounding boxes were automatically extracted, and all those that were considered wrong were removed from the test data. The final DIONYSUS dataset consists of 111,119 instances for training, 6,619 for validation and 4,824 for test[10]. These figures, along with other relevant statistics, are reported in Table 2.

Dataset Annotation We employed two annotators to perform the tasks needed for the creation of the DIONYSUS dataset: associating each MSCOCO category label with a WordNet concept and validating the image-definition instances of the test set. The annotators were required to be proficient in English and have a prior knowledge of WordNet's structure.

The first task of annotation consisted in associating a definition with one of the MSCOCO object labels. To this end, one of the two annotators was presented with 10 images randomly selected from the ones associated with a target object label and a list of definitions linked to the meanings in WordNet[11] of the word representing the MSCOCO category (the subregion representing the target object was outlined in every image). The annotator was asked to select one definition among those provided. There was no case in which the annotator was undecided between two different definitions, thus all categories were associated with one definition only. Once the process of annotation had been completed, the second annotator was asked to validate the associations created in the previous step. In case of conflict, the two annotators were asked to discuss until they reached an agreement. For this

[5]https://cocodataset.org/

[6]Labels are from the 2017 Object Detection challenge.

[7]We recall that each concept in WordNet has a definition.

[8]We used the Faster R-CNN model trained on Visual Genome provided by Anderson et al. (2018).

[9]Each subregion of an image corresponds to a different instance in the DIONYSUS dataset.

[10]2,035 instances were removed from the original test set after the validation process.

[11]Each meaning is associated with one unique definition.

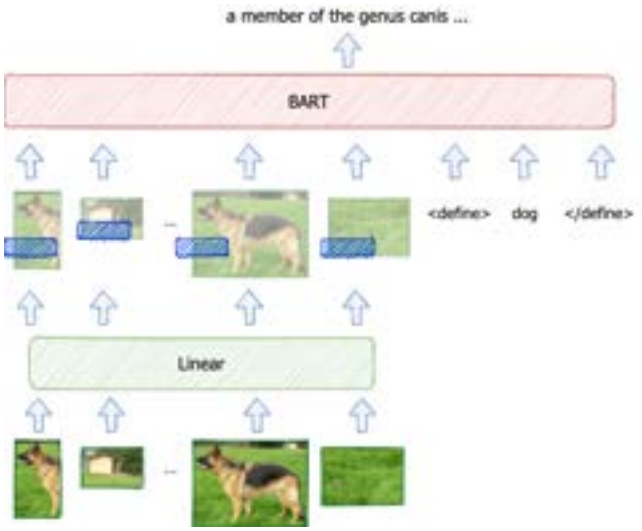

Figure 4: Input/output example of veBART on the DEMETER dataset.

task the annotators worked for a total of 4 hours (1.5 hours per annotator plus 1 hour of discussion).

For the second task, both the annotators were asked to validate the instances of the DIONYSUS test set. For each instance, they were given an image in which a subregion representing the automatically-extracted bounding box was outlined, along with the corresponding MSCOCO object category and the definition retrieved thanks to the mapping between object categories and WordNet definitions. The annotators were requested to give a binary answer as to whether the association between the subregion and the definition was correct or not[12]. Each annotator was presented with 62.5% of the dataset, hence granting a 25% share of annotations to be validated by both annotators. At the end of the annotation process, we hired an external validator who had the same qualifications as the two annotators. The validator was given the same task and was asked to annotate the set of instances shared by both original annotators. We computed the inter-annotator agreement on this portion of the dataset validated by both the annotators and the validator, resulting in an average pairwise Cohen's $\kappa = 0.79$. Depending on the magnitude guidelines one follows, this agreement should be considered either substantial (Landis and Koch 1977) or excellent (Fleiss and Cohen 1973). While the annotators worked for an average of 48 hours to perform this task (24 hours per annotator), the validator worked for 6 hours. Both the annotators and the validator were paid in accordance with the standard wages of their country of residence.

Annotators' Guidelines

We provided the annotators with a user-friendly interface built in Python 3.8. Annotators were instructed to validate entries according to their intuition. For each entry, the annotators were presented with a context image in which the target object was highlighted by means of a red box, along with the automatically-generated definition for the target and the label identifying the class of the object in MSCOCO. Annotators were simply required to validate the appropriateness of the definition used to describe the target by typing in the interface either *T* or *F* to validate or discard the given entry, respectively. We provided each of them with the following list of exceptional cases:

1. If the subregion depicts two objects of the same category and the definition is appropriate for the representative object, then the instance must be labeled as *T* (see Figure 5a).
2. If the definition associated with the object is correct but the visual object is not clearly the focus of the highlighted subregion, then the instance must be tagged as *F* (see Figure 5b).
3. If the object highlighted in the subregion is not clearly framed, the instance must be labeled *F* (see Figure 5c).

[12] See next Section for Annotators' guidelines.

(a) A small ring-shaped friedcake.

(b) A board with wheels that is ridden in a standing or crouching position and propelled by foot.

(c) A seat for one person, with a support for the back.

Figure 5: Exceptional cases provided along with annotators' guidelines.

Experimental Setup

In this Section we describe the evaluation we performed on our newly-created Visual Definition Modeling datasets.

Datasets

We use DEMETER and DIONYSUS created and split as previously described, i.e., each dataset is split into three subsets, i.e., training, validation and test.

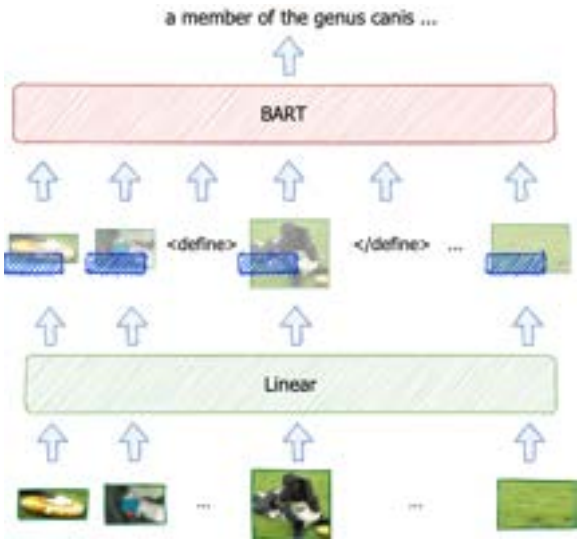

Figure 6: Input/output example of veBART on the DIONYSUS dataset.

Baselines

We devised two distinct baselines, one representative of each class, i.e., single and dual stream. For the single-stream model we used VL-BERT (Su et al. 2019) and LXMERT (Tan and Bansal 2019) as a dual-stream architecture. Both models take as input a list of words and image subregions features and, while VL-BERT passes them all through the same transformer encoder, LXMERT first encodes them separately, secondly applies a cross-modality attention, and finally outputs a vision and a language vectors. We concatenate the vision and language outputs into a unique sequence and pass them through 12 decoder transformer layers. For each architecture, we created two versions, one initializing the decoder layers randomly (VL-BERT$_{\text{R}}$ and LXMERT$_{\text{R}}$) and one initializing the decoder layers with BART weights (Lewis et al. 2020) (VL-BERT$_{\text{B}}$ and LXMERT$_{\text{B}}$, respectively)[13]. The weights of VL-BERT and LXMERT encoders, instead, were initialized with those of their pretrained models. Both models take as visual inputs the 36 features extracted by the Faster R-CNN model (Ren et al. 2015).

Visually-Enhanced BART (veBART) We also experimented with BART pretrained model (Lewis et al. 2020). BART is an encoder-decoder language model that has been trained on text with the goal of reconstructing masked input sequences of arbitrary length[14]. In order to make BART capable of processing visual inputs, we added a linear transformation layer that projects visual vectors into the BART input space and named this model veBART (see Figure 4 and 6). As visual input for the veBART model, we experimented with two different representations: i) the pretrained Faster R-CNN model used for all the other baselines[15] (veBART$_{\text{FR}}$); and ii) the visual output of LXMERT (veBART$_{\text{LX}}$).

[13]For these and the following baselines, BART pretrained weights are the pretrained weights of the `bart-base` architecture in the transformers library.

[14]We acknowledge the existence of KM-BART (Xing et al. 2021) and VL-BART (Cho et al. 2021), however both these models are mostly contemporaneous to this work and their developers have not released either the code or the pretrained models.

Model	BL	R-L	MT	BS
VL-BERT$_{\text{R}}$	15.5	11.9	5.0	6.0
LXMERT$_{\text{R}}$	14.1	12.6	6.0	12.0
VL-BERT$_{\text{B}}$	18.8	13.8	6.0	12.0
LXMERT$_{\text{B}}$	18.2	14.1	6.5	13.2
veBART$_{\text{LX}}$	**26.7**	**23.3**	**11.6**	**25.3**
veBART$_{\text{FR}}$	25.5	21.8	10.6	23.1

Table 3: Results on the DEMETER test set. Columns: BLEU, ROUGE-L, METEOR, BERTScore (BL, R-L, MT, BS).

Training

All models were trained end-to-end with teacher forcing on each benchmark in order to minimize the cross-entropy between the generated definition and the gold one. Across experiments, we used a workstation with a x86_64 architecture and a NVIDIA V100 and 16GBs of RAM. For both benchmarks, we enclosed the object to be defined between the special tags `<define>` and `</define>`, as depicted in Figures 4 and 6. In the case that the object to be defined was an image patch, we surrounded its visual embedding with the word embeddings of the two special tags. We trained all models on the training split of each of our datasets for $40{,}000$ steps with batch size equal to 10 and 16 steps of gradient accumulation. We adopted an early stopping strategy and stopped the training when the validation loss ceased decreasing for 10 subsequent evaluation steps[16]. For each architecture, we tested the set of weights attaining the lowest loss on the validation set. The learning rate was set to $3\text{e}{-5}$ and weight decay to 0.01. As for veBART$_{\text{LX}}$, we kept LXMERT weights frozen during training.

The number of parameters of VL-BERT-based models (VL-BERT$_{\text{R}}$ and VL-BERT$_{\text{B}}$) and LXMERT-based (LXMERT$_{\text{R}}$ and LXMERT$_{\text{B}}$) models is 208M and 304M, respectively. The number of parameters of veBART$_{\text{FR}}$ and veBART$_{\text{LX}}$ is, instead, approximately 139M.

Metrics

In order to evaluate the system performance on our benchmarks, we adopted some of the most commonly used metrics employed for scoring generation systems.

As string-matching evaluation measures, we considered BLEU (Papineni et al. 2002) and ROUGE-L (Lin 2004). However, as these metrics only take into consideration the overlap between the generated strings and the gold ones, we also included in our pool METEOR (Banerjee and Lavie 2005) and

[15]We used the 36-feature pretrained model available at https://github.com/peteanderson80/bottom-up-attention

[16]Evaluation every $5{,}000$ steps.

Model	BL	R-L	MT	BS
VL-BERT$_{R}$	19.0	24.5	19.3	13.3
LXMERT$_{R}$	18.6	24.1	18.4	14.7
VL-BERT$_{B}$	23.9	24.1	12.6	26.0
LXMERT$_{B}$	57.8	54.8	30.5	41.1
veBART$_{LX}$	48.1	45.6	26.9	43.8
veBART$_{FR}$	**76.4**	**69.0**	**47.0**	**53.8**

Table 4: Results on the DIONYSUS test set. Columns: BLEU, ROUGE-L, METEOR, BERTScore (BL, R-L, MT, BS).

BERTScore (Zhang et al. 2020), which also take into account the semantic aspect of a sentence. In fact, while METEOR exploits WordNet synonyms and stemming, BERTScore leverages the similarity between BERT embeddings (Devlin et al. 2019) of the reference and the generated sentence.

Results

In what follows, we report the results of all the proposed baselines on the test sets of DEMETER and DIONYSUS.

DEMETER

In Table 3, we provide the scores for all the baselines on the DEMETER test set. As can be seen, when the vision-and-language models do not leverage a pretrained encoder, i.e., VL-BERT$_{R}$ and LXMERT$_{R}$, they achieve the lowest score across the board. In fact, neither VL-BERT nor LXMERT were exposed to a generation task during their pretraining, hence they are penalized when it comes to generating a definition. When equipped with a pretrained decoder (VL-BERT$_{B}$ and LXMERT$_{B}$), their performance increases according to all metrics. These results suggest that further studies – in line with KM-BART (Xing et al. 2021) and VL-BART (Cho et al. 2021) – are needed to combine different pretraining objectives for vision-and-language models, thus allowing them to also generate text conditioned on different modalities effectively. In fact, when leveraging a pretrained encoder-decoder, i.e., veBART models, we observe a consistent increase in performance. This result is interesting since the underlying model, i.e., BART, was never exposed to visual inputs during its pretraining, and the only weights dedicated to the visual part are those of a linear layer that transforms the input image features. Nonetheless, this seems sufficient for veBART to handle images effectively and to define the input word accordingly. Interestingly enough, LXMERT features seem to bring slightly more benefits than the raw Faster R-CNN vectors, since LXMERT had already been used to consider both image and textual features.

DIONYSUS

In Table 4, we show the models' performance on the DIONYSUS dataset. As can be seen, the scores are higher than the DEMETER benchmark overall. This is partially due to the nature of the task. Indeed, while a single word may assume different meanings and thus be defined differently according to the context, a visual object is usually less ambiguous, as in most cases it already identifies a specific concrete object. At the same time, DIONYSUS test instances refer to objects' classes that have been seen during training, hence making the task easier in general. Aside from this, VL-BERT$_{R}$ and LXMERT$_{R}$ still perform poorly, confirming that training a transformer decoder from scratch is not effective. Indeed, when using a pretrained decoder (VL-BERT$_{B}$ and LXMERT$_{B}$), both attain significantly better performance, with LXMERT$_{B}$ reporting from roughly 12 to 39 points increments depending on the measure.

Interestingly enough, veBART$_{LX}$ performs worse than LXMERT$_{B}$ on all measures but BERTScore. While this may seem surprising, we note that the input of this task is only visual, a setting which was not included in the LXMERT pretraining. In fact, when using Faster R-CNN vectors, which are tailored to represent visual objects, veBART outperforms all the other models by a large margin.

All in all, both our benchmarks highlighted that adapting a pretrained language model specialized on text to the multimodal setting is more effective than pretraining multimodal architectures from scratch. Indeed, veBART, while being exposed to images at finetuning time only, outperformed all its alternatives, showing a better ability to encode word and visual object semantics.

Conclusion

In this paper, we introduced the new task of Visual Definition Modeling, i.e., a multimodal task, where, given a context represented by an image and an object (either a visual patch of the image or a word that represents the concept shown by the image), a model has to generate a textual definition. The task aims at investigating whether modern multimodal architectures have a deep comprehension of words and objects in a visual context. To this end we put forward two challenging datasets: DEMETER, where a concept has to be defined given a word and a visual context, and DIONYSUS, where the object to be defined is a visual patch of the input image. By conducting experiments with six different baselines, four of which were specialized to deal with multimodal inputs and two featuring a text-only pretrained encoder-decoder architecture, we showed that pretraining only on language inputs is a more effective way to learn word semantics. To the contrary, multimodal baselines struggle to attain competitive performance on both tasks, suggesting that including a pure text-only pretraining objective could bring much benefit to these architectures. All the data produced and the code to run the experiments are available at https://github.com/SapienzaNLP/visual-definition-modeling.

As future work, we plan to further extend the Definition Modeling paradigm to new modalities as we believe it to be an essential probe to test real semantic comprehension. Furthermore, we will use our tasks as additional pretrained data for multimodal architectures and measure their impact on other downstream tasks.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

This work was supported in part by the MIUR under the grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University and by the Innovation Fund Denmark under the LEGALESE project.

References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proc. of CVPR*.

Banerjee, S.; and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proc. of ACL*, 65–72.

Bender, E. M.; and Koller, A. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proc. of ACL*, 5185–5198.

Bevilacqua, M.; Maru, M.; and Navigli, R. 2020. Generationary or "How We Went beyond Word Sense Inventories and Learned to Gloss". In *Proc. of EMNLP*, 7207–7221.

Bugliarello, E.; Cotterell, R.; Okazaki, N.; and Elliott, D. 2020. Multimodal Pretraining Unmasked: Unifying the Vision and Language BERTs. *CoRR*, abs/2011.15124.

Calabrese, A.; Bevilacqua, M.; and Navigli, R. 2020. Fatality Killed the Cat or: BabelPic, a Multimodal Dataset for Non-Concrete Concepts. In *Proc. of ACL*, 4680–4686.

Cho, J.; Lei, J.; Tan, H.; and Bansal, M. 2021. Unifying Vision-and-Language Tasks via Text Generation. *CoRR*, abs/2102.02779.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*, 4171–4186.

Fleiss, J. L.; and Cohen, J. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3): 613–619.

Gella, S.; Elliott, D.; and Keller, F. 2019. Cross-lingual Visual Verb Sense Disambiguation. In *Proc. of NAACL*, 1998–2004.

Gella, S.; Lapata, M.; and Keller, F. 2016. Unsupervised Visual Sense Disambiguation for Verbs using Multimodal Embeddings. In *Proc. of NAACL*, 182–192.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proc. of CVPR*.

Gupta, S.; and Malik, J. 2015. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*.

Hudson, D. A.; and Manning, C. D. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *Proc. of CVPR*.

Karpathy, A.; and Fei-Fei, L. 2017. Deep Visual-Semantic Alignments for Generating Image Descriptions. *Transactions on Pattern Analysis and Machine Intelligence*, 39: 664–676.

Landis, J. R.; and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *biometrics*, 159–174.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proc. of ACL*, 7871–7880.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755.

Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*.

Miller, G. A.; Beckwith, R.; Fellbaum, C. D.; Gross, D.; and Miller, K. 1990. WordNet: an Online Lexical Database. In *IJL*, 235–244.

Navigli, R.; Bevilacqua, M.; Conia, S.; Montagnini, D.; and Cecconi, F. 2021. Ten years of BabelNet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4559–4567.

Navigli, R.; and Ponzetto, S. P. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artiff. Intell.*, 193: 217–250.

Noraset, T.; Liang, C.; Birnbaum, L.; and Downey, D. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proc. of the AAAI*, volume 31.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, 311–318.

Plummer, B. A.; Wang, L.; Cervantes, C. M.; Caicedo, J. C.; Hockenmaier, J.; and Lazebnik, S. 2015. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *ICCV*, 2641–2649.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *NeurIPS*, volume 28.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3): 211–252.

Saenko, K.; and Darrell, T. 2009. Unsupervised Learning of Visual Sense Models for Polysemous Words. In *NeurIPS*.

Su, H.; Shen, X.; Zhang, R.; Sun, F.; Hu, P.; Niu, C.; and Zhou, J. 2019. Improving Multi-turn Dialogue Modelling with Utterance ReWriter. In *Proc. of ACL*, 22–31.

Suhr, A.; Zhou, S.; Zhang, A.; Zhang, I.; Bai, H.; and Artzi, Y. 2019. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proc. of ACL*, 6418–6428.

Tan, H.; and Bansal, M. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proc. of EMNLP*, 5100–5111.

Xie, N.; Lai, F.; Doran, D.; and Kadav, A. 2019. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*.

Xing, Y.; Shi, Z.; Meng, Z.; Ma, Y.; and Wattenhofer, R. 2021. KM-BART: Knowledge Enhanced Multimodal BART for Visual Commonsense Generation. *CoRR*, abs/2101.00419.

Yatskar, M.; Zettlemoyer, L.; and Farhadi, A. 2016. Situation Recognition: Visual Semantic Role Labeling for Image Understanding. In *CVPR*, 5534–5542.

Yu, L.; Poirson, P.; Yang, S.; Berg, A. C.; and Berg, T. L. 2016. Modeling context in referring expressions. In *Proc. of ECCV*, 69–85. Springer.

Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From Recognition to Cognition: Visual Commonsense Reasoning. In *CVPR*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *ICLR*.

Zhang, X.; Yang, Y.-H.; Han, Z.; Wang, H.; and Gao, C. 2013. Object Class Detection: A Survey. In *ACM Comput. Surv.*