Semantic Bottleneck for Computer Vision Tasks

Maxime Bucher[1,2(✉)], Stéphane Herbin[1], and Frédéric Jurie[2]

[1] ONERA, Université Paris-Saclay, 91123 Palaiseau, France
bucher.maxime@gmail.com
[2] Normandie Univ, UNICAEN, ENSICAEN, CNRS, Mont-Saint-Aignan, France

Abstract. This paper introduces a novel method for the representation of images that is semantic by nature, addressing the question of computation intelligibility in computer vision tasks. More specifically, our proposition is to introduce what we call a *semantic bottleneck* in the processing pipeline, which is a crossing point in which the representation of the image is entirely expressed with natural language, while retaining the efficiency of numerical representations. We show that our approach is able to generate semantic representations that give state-of-the-art results on semantic content-based image retrieval and also perform very well on image classification tasks. Intelligibility is evaluated through user centered experiments for failure detection.

1 Introduction

Image-to-text tasks have made tremendous progress since the advent of deep learning approaches (see *e.g.*, [1]). The work presented in this paper builds on these new types of image-to-text functions to evaluate the capacity of textual representations to semantically and fully encode the visual content of images for demanding applications, in order to allow the prediction function to host a *semantic bottleneck* somewhere in its processing pipeline (Fig. 1). The main objective of a semantic bottleneck is to play the role of an *explanation* of the prediction process since it offers the opportunity to examine meaningfully on what ground will further predictions be made, and potentially decide to reject them either using human common sense knowledge and experience, or automatically through dedicated algorithms. Such an explainable semantic bottleneck instantiates a good tradeoff between prediction accuracy and interpretability [2].

Reliably evaluating the quality of an explanation is not straightforward [2–5]. In this work, we propose to evaluate the explainability power of the semantic bottleneck by measuring its capacity to detect failure of the prediction function, either through an automated detector as [6], or through human judgment. Our proposal to generate the surrogate semantic representation is to associate a global and generic textual image description (caption) and a visual quiz in the

C. V. Jawahar et al. (Eds.): ACCV 2018, LNCS 11362, pp. 695–712, 2019.
https://doi.org/10.1007/978-3-030-20890-5_44

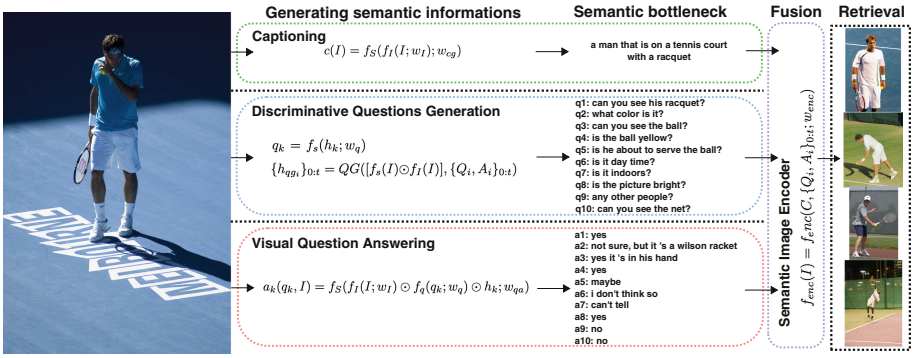


Fig. 1. Semantic bottleneck approach: images are replaced by purely but rich textual representations, for tasks such as multi-label classification or image retrieval.

form of a small list of questions and answers that are expected to refine contextually the generic caption. The production of this representation is adapted to the vision task and learned from annotated data.

The main contributions of this paper are: *(i)* The design of two processing chains for content-based image retrieval and multi-label classification hosting a semantic bottleneck; *(ii)* An original scheme to select sequentially a list of questions and answers to form a semantic visual quiz; *(iii)* A global fusion approach jointly exploiting the various components of the semantic representation for image retrieval or multi-label classification; *(iv)* A complete evaluation on the MS-COCO database exploiting Visual Dialog annotations [1] showing that it is possible to enforce a semantic bottleneck with only 5% of performance loss on multi-label classification, but a 10% performance gain for image retrieval, when compared to image feature-based approaches; *(v)* An evaluation of the semantic bottleneck explanation capacity as a way to detect failure in the prediction process and improve its accuracy by rejection.

2 Related Works

Extracting Semantic Information From Images. The representation of images with semantic attributes has received a lot of attention in the recent literature. However, with the exception of the DAP model [7], which is not performing very well, such models produce vector representations that are not intelligible at all. In contrast, image captioning [8,9] is by nature producing intelligible representations and can be used to index images. As an illustration, Gordo *et al.* [10] addressed the task of retrieving images that share the same semantics as the query image using captions. Despite the success of such recent methods, it has been observed [11] that such approaches produce captions that are similar when they contain one common object, despite their differences in other aspects. Addressing this issue, [12] proposed a contrastive learning method for image

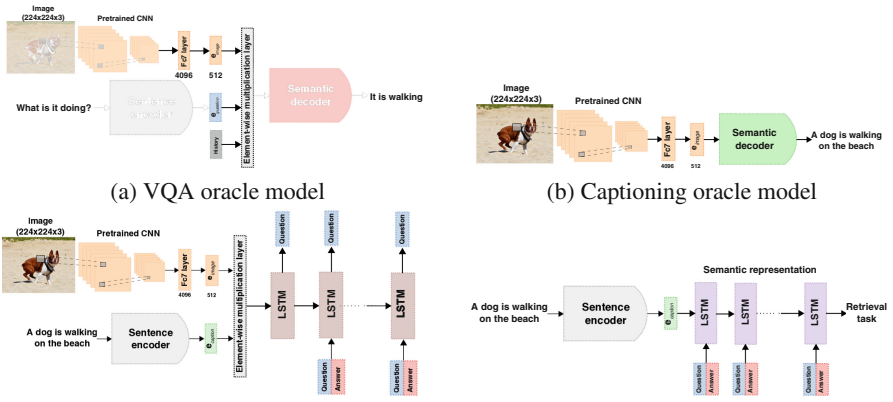
captioning encouraging distinctiveness, while maintaining the overall quality of the generated captions. Another way to enrich the caption is to generate a set of questions/answers such as proposed in the Visual Dialog framework [1,13,14]. This is what we propose to do by learning how to build dialogs complementary to image captions.

Transferring Information From Other Domains. Producing semantic description of images in natural languages is barely possible without transferring semantic information – expressed as semantic attributes, natural language expressions, dictionaries, *etc.*– from auxiliary datasets containing such information to novel images. This is exactly what Visual Question Answering models can do, the VQA challenge offering important resources, in the form of semantic images, questions, or possible answers. Research on VQA has been very active during the last two years. [15] proposed an approach relying on Recurrent Neural Network using Long Short Term Memory (LSTM). In their approach, both the image (CNN features) and the question are fed into the LSTM to answer a question about an image. Once the question is encoded, the answers can be generated by the LSTM. [16] study the problem of using VQA knowledge to improve image-caption ranking. [17], motivated by the goal of developing a model based on grounded regions, introduces a novel dataset that extends previous approaches and proposes an attention-based model to perform this task. On their side, [18] proposed a model receiving the answers as input and predicts whether or not an image-question-answer triplet is correct. Finally, [19] proposes another VQA method combining an internal representation of the image content with the information extracted from general knowledge bases, trying to make the answering of more complex questions possible.

Producing Intelligible Representations. The ubiquitousness of deep neural networks in modern processing chains, their structural complexity and their opacity have motivated the need of bringing some kind of intelligibility in the prediction process to better understand and control its behavior. The vocabulary and concepts connected to intelligibility issues are not clearly settled. (explanation, justification, transparency, *etc.*) Several recent papers have tried to clarify those expressions [2–5,20–23] and separate the various approaches in two goals: build interpretable models and/or provide justification of the prediction. [24] for instance, described an interpretable proxy (a decision tree) able to explain the logic of each prediction of a pretrained convolutional neural networks. The generation of explanations as an auxiliary justification has been addressed in the form of a visual representation of informative features in the input space, usually heat maps or saliency maps [25–27], as textual descriptions [28], or both [29]. A large body of studies [30–33] have been interested in visually revealing the role of deep network layers or units. Our semantic bottleneck approach fuses those two trends: it provides a directly interpretable representation, which can be used as a justification of the prediction, and it forces the prediction process itself to be interpretable in some way, since it causally relies on an intermediate semantic representation.

Evaluating Explanations. The question of clearly evaluating the quality or usability of explanations remains an active problem. [25] described a human-centered experimental evaluation assessing the predictive capacity of the visual explanation. [27] proposed to quantify explanation quality by measuring two desirable features: continuity and selectivity of the input dimensions involved in the explanation representation. [34,35] described geometric metrics to assess the quality of the visual explanation with respect to landmarks or objects in the image. [36] questioned the stability of saliency based visual explanations by showing that a simple constant shift may lead to uninterpretable representations. In our work, we take a dual approach: rather than evaluating the capacity of the explanation to be used as a surrogate or a justification of an ideal predictive process, we evaluate its quality as an ability of detecting bad behavior, *i.e.* detect potential wrong predictions.

Generating Distinctive Questions. If the generation of questions about text corpora has been extensively studied (see *e.g.*, [37]), the generation of questions about images has driven less attention. We can, however, mention the interesting work of [38] where discriminative questions are produced to disambiguate pairs of images, or [39] which introduced the novel task (and dataset) of visual question generation. We can also mention the recent work of Das *et al.* [40] which bears similarity with our approach but differs in the separation between the question generator and semantic representation encoder, and is not applied to the same tasks. Our work builds on the observation made by [41,42] – questions that are asked about an image provide information regarding the image and can help to acquire relevant information about an image – and proposes to use automatically generated discriminative questions as cues for representing images.



(a) VQA oracle model

(b) Captioning oracle model

(c) Discriminative Question Generator

(d) Semantic Representation Encoder

Fig. 2. Functional diagrams of the various components of the global algorithm.

3 Approach

This paper proposes a method allowing to turn raw images into rich semantic representations—a semantic bottleneck—which can be used alone (without using the image itself) to compare images or classify them. At the heart of this image encoder is a process for generating an ordered set of questions related to the image content, which are used, jointly with the answers to these questions, as an image substitute. Questions are generated sequentially, each question (as well as its answer) inducing the next questions to be asked. Such a set of questions/answers should semantically represent the visual information of images and be useful for disambiguating image representations in retrieval and classification task. Answering the question is done with a VQA model, used as an oracle, playing the role of a fine-grained information extractor. Furthermore, this sequence of Q/A is designed to be complementary to an image caption which is also automatically generated. We can use the analogy of human reasoning, starting with the image caption as a starting point and asking question to an oracle to get iteratively more information on the image. The proposed visual dialog allows to enrich image caption representations and leads to a stronger semantic representation. Finally, captions and visual dialogs are combined and turned into a compact representation that can be used easily to compare images (for retrieval task) or to infer class labels (classification tasks).

Consequently, our model is composed of two main components: (i) a discriminative visual question generator (ii) an encoding block taking Q/A and captions as inputs and producing an image representation. These two blocks, trained end-to-end, rely on two oracles: (i) an image caption generator which visually describes images with natural language sentences. (ii) a visual question answering model capable of answering free-from questions related to images. We call these last 2 parts of the model *oracles* as they are trained independently of the main tasks and used as external knowledge base.

3.1 Vector Space Embedding, Encoders, Decoders

The core objective of our approach is to generate semantic expressions in natural language (questions, answers or captions) that could represent images compactly and informatively. We define the various natural language elements as sequences of words from a fixed vocabulary of words and punctuation $\{w_0, \ldots, w_{n_w}\}$, where w_0 has the special meaning of marking the end of all phrases as a "full stop". The space of all possible sequences in natural language is denoted by $\mathcal{P}$. Any caption (c), question (q) or answer (a) belongs to the same set $\mathcal{P}$.

Most of learning based algorithms exploit vector space representations as inner states, or in the optimized criterion. A first issue is therefore to make possible the embedding of natural language expressions, and also images, into a vector space. For simplicity of design, we made all the necessary embeddings belong to a S-dimensional real valued vector space. Typically, to give an order of magnitude, we took $S = 512$ in our experiments. We therefore define semantic *encoders* as mappings from $\mathcal{P}$ to $\mathbb{R}^S$, and semantic *decoders* or *generators* as mappings from the vector space $\mathbb{R}^S$ to $\mathcal{P}$.

Image Encoder. The first element to be encoded is the source image $I \in \mathcal{I}$. The encoder (denoted as f_I) is a mapping $f_I : \mathcal{I} \rightarrow \mathbb{R}^S$, provided by last FC layer (fc7) of a VGG-VeryDeep-19 network [43] (pre-trained on Imagenet [44]) followed by a non-linear projection (fully connected layer + tanh non-linearity unit) reducing the dimensionality of the fc7 features (4096-d) to S. The parameters of the non-linear projection, denoted as w_I, are the only parameters of this embedding that have to be learned, the parameters of the VGG-VeryDeep-19 being considered as fixed. We write $f_I(I; w_I)$ to make apparent the dependency on the parameters w_I, when needed.

Natural Language Encoder. This encoder maps any set of words and punctuation $s \in \mathcal{P}$ (captions, questions, answers) to the embedding space $\mathbb{R}^S$. We will use 3 different natural language encoders in the global algorithm, for captions, questions and answers, all sharing the same structure and the same weights. We hence refer to them using the same notation (f_p). The encoder uses a standard Long Short Term Memory network (LSTM) [45]. Used as an encoder, the LSTM is simply a first order dynamic model: $y_t = \text{LSTM}_p(y_{t-1}, p_t)$ where y_t is the concatenation of the long-term cell state and the short-term memory, and p_t is the current input at time t. Given a natural language sequence of words $p = \{p_t\}_{t=1:T_p}$, its encoding $f_p(p)$ is equal to the memory state y_t after iterating on the LSTM T_p times and receiving the word p_t at each iteration. Denoting w_p the set of weights of the LSTM model, the natural language embedding is therefore defined as: $y_{T_p} = f_p(p; w_p)$ with the initial memory cell $y_0 = 0$.

In practice, instead of using words by their index in a large dictionary, we encode the words in a compact vector space, as in the word2vec framework [46]. We found it better since synonyms can have similar encodings. More precisely, this local encoding is realized as a linear mapping w_{w2vec}, where w_{w2vec} is a matrix of size $n_{w2vec} \times n_w$ and each original word is encoded as a one-hot vector of size n_w, the size of the vocabulary. The size of the word embedding n_{w2vec} was 200 in our experiments. This local word embedding simply substitutes $w_{w2vec} \cdot p_t$ to p_t in the LSTM input.

Semantic Decoder. The semantic decoder is responsible for taking an embedded vector $s \in \mathbb{R}^S$ and producing a sequence of words in natural language belonging to $\mathcal{P}$. It is denoted by f_s. All the semantic decoders we exploit have the structure of an LSTM network. Several semantic decoders will be used for questions, answers and captions in the overall algorithm, but with distinct weights and different inputs. These LSTMs have an output predicting word indexes according to a softmax classification network: $p_{t+1} = \text{Softmax}(y_t)$ which is re-injected as input of the LSTM at each iteration.

Formally, we can write the semantic decoder as a sequence of words generated by an underlying LSTM first order dynamic process with observations p_t as $y_t = \text{LSTM}_s(y_{t-1}, p_t)$. At time t the input receives the word generated at the previous state p_t, and predicts the next word of the sentence p_{t+1}. When the word w_0 meaning "full stop" is generated at time T_s, it ends the generation. The global decoding is therefore a sequence of words of length $T_s - 1$: $f_s(s; w) = \{p_t\}_{t=1:T_s-1}$

with the initial state of the LSTM being the embedded vector to decode ($y_0 = s$) and the first input being null ($p_0 = 0$). The symbol w refers to the learned weights of the LSTM and the softmax weights, and is different for each type of textual data that will be generated (captions, questions and answers).

3.2 Captioning Model

The visual captioning part of the model is used as an external source of knowledge, and is learned in a separate phase. It takes an image (I) and produces a sentence in natural language describing the image. We used an approach inspired by the winner of the COCO-2015 Image Captioning challenge [8]. This approach, trainable end-to-end, combines a CNN image embedder with a LSTM-based Sentence Generator (Fig. 2(b)). More formally, it combines an image encoder and a semantic decoder, such are described previously, which can be written as: $c(I) = f_s(f_I(I; w_I); w_{cg})$ where w_{cg} is the specific set of learned weights of the decoder. This caption model acts as an oracle, providing semantic information for the Visual Discriminative Question Selection component, and to the Semantic Representation Encoder.

3.3 Visual Question Answering Model

The VQA model is the second of the two components of our model used as oracles to provide additional information on images. Its role is to answer independent free-form questions about an image. It receives questions (Q) in natural language and an image (I) and provides answers in natural language ($a(Q, I)$). Our problem is slightly different from standard VQA because the VQA model now has to answer questions sequentially from a dialog. It means that the question Q_k can be based on the answer of the previous questions Q_{k-1} and answers $a(Q_{k-1}, I)$. We first present the formulation of a standard VQA and then show how to extend it so it can answer questions from a dialog.

Inspired by [47], our VQA model combines two encoders and one decoder of those described previously (Fig. 2-a):

$$a(Q, I) = f_s(f_I(I; w_{Iqa}) \odot f_p(Q; w_{pqa}); w_{qa}) \tag{1}$$

The fusion between image and question embeddings is done by an element wise product ($\odot$) between the two embeddings, as proposed by [47].

We now consider the case where questions are extracted from a dialog by extending Eq. (1), where k represents the k-th step of the dialog. We introduce another term h_k in the element-wise product to encode the history of the dialog as:

$$a_k(Q_k, I) = f_s(f_I(I; w_{Iqa}) \odot f_p(Q_k; w_{pqa}) \odot h_k; w_{qa}) \tag{2}$$

The history h_k is simply computed as the mean of the previously asked questions/answers, and encoded using f_p. This state integrates past questions and is

expected to help the answering process. We tried other schemes to summarize history (concatenation, LSTM) without clear performance increase.

We prefer the VQA model to the Visual Dialog model [1], as this latter is optimized for the task of image guessing, while we want to fine-tune the question/answer sequence for different tasks (multi-label prediction and image retrieval).

3.4 Discriminative Question Generation

This part of the model is responsible for taking an image and a caption – which is considered as the basic semantic representation of the image – and produces a sequence of questions/answers that are complementary to the caption for a specific task (multi-label classification or retrieval).

The caption describing the image is generated using the captioning model presented in Sect. 3.2, and denoted $c(I) \in \mathcal{P}$. This caption is encoded with $f_p(c(I); w_p) \in \mathbb{R}^S$. Image and caption embeddings are then combined by an element-wise product $f_p(c(I)) \odot f_I(I)$ used as an initial encoded representation of the image.

This representation is then updated iteratively by asking and answering question, one by one, hence iteratively proposing a list of discriminative questions. Again, we use a LSTM network (Fig. 2-c), but instead of providing a word at each iteration as for $f_s(s; w)$ we inject a question/answer $[\tilde{q}_k, \tilde{a}_k]$ pair encoded in a vector space from the natural language question/answer $[Q_k, A_k]$ using f_p, with initial memory $y_0 = f_I(I) \odot f_p(c(I))$ and initial input $\tilde{q}_0 = \tilde{a}_0 = 0$: $y_k = \text{LSTM}_q(y_{k-1}, [\tilde{q}_k, \tilde{a}_k])$ The actual questions are then decoded from the inner LSTM memory y_k and fed to the VQA model to obtain the answer using Eq. (2): $Q_{k+1} = f_s(y_k; w_{sq})$, and, $A_{k+1} = a_{k+1}(Q_{k+1}, I)$

Using this iterative process, we generate, for each image, a sequence of questions-answers refining the initial caption: $f_q(I; w_q) = \{Q_k, A_k\}_{k=1:K}$ where w_q is the set of weights of the underlying LSTM network of the previous equation, and K is an arbitrary number of questions.

3.5 Semantic Representation Encoder

Our objective is to evaluate the feasibility of substituting a rich semantic representation to an image and achieve comparable performance than an image feature based approach, for several computer vision tasks. This representation has to be specifically generated to the target task, to be efficient.

Once again, many modern computer vision approaches relying on a learning phase require that data are given as fixed dimension vectors. The role of the module described here is to encode the rich semantic representation in $\mathbb{R}^S$ to feed the retrieval or the multi-label classification task.

The encoder makes use of a LSTM network where the question/answer sequence $\{Q_k, A_k\}_{k=1:K}$ is used as input, $y_k = \text{LSTM}_e(y_{k-1}, [\tilde{q}_k, \tilde{a}_k])$ and the initial memory state y_0 is equal to the encoded caption $f_p(c(I); w_p)$ (Fig. 2(d)).

If w_e is the set of weights from the underlying LSTM, the rich semantic representation y_K is encoded as: $y_K = f_{enc}\left([\{Q_k, A_k\}_{k=1:K}, c(I)]\,; w_e\right)$.

3.6 Training the Model

The global training is divided in two phases. The first phase learns the two so-called oracles independently: image captioning and VQA. The second phase learns to generate the visual quiz, the image encoder and the semantic encoder jointly for a specific task, based on information provided by the two oracles.

The parameters of the VQA, namely w_{Iqa} and w_{pqa} for the encoders, and w_{qa} (Eq. 2) are learned in a supervised way on the Visual Dialog dataset [1]. The learning criterion consists in comparing the answer words generated by the model with those of the ground truth for each element of the sequence of questions, and is measured by a cross entropy.

The captioning is also learned by comparing each word sequentially generated by the algorithm to a ground truth caption, and is also measured by a cross entropy loss.

The question/answer generator $f_q(I; w_q)$ and the semantic representation encoder f_{enc} are learned jointly end to end. Each of the modules manages its own loss: for the question generator, the sequence of questions is compared to the ground truth of questions associated with each image using a cross entropy at each iteration. The semantic encoding, however, is specifically evaluated by a task-dependent loss: a cross entropy loss for each potential label for the multi-label classification task, a ranking loss for the image retrieval task. When the question generation model converges, only the task-dependent loss is kept in order to fine-tune the question selection part.

The retrieval loss is a bit more complex than the others (cross entropy). Basically, it is based on the assumption that ground truth captions are the most informative image representations and that any other representation should follow the same similarity ranking as captions provide. We follow the approach proposed in [10] to define the retrieval loss as a function of triplet data q, d^+ (positive pair) and d^- (negative pair) to be $L(q, d_+, d_-) = \max(0, m - \phi(q)^T\phi(d^+) + \phi(q)^T\phi(d^-))$ where q and d^+ are expected to be more similar than q and d^-, and ϕ is the representation function to be learned, *i.e.* the output of f_{enc}, and m is a free coefficient playing the role of a margin. The reference similarity comparison is computed from ground truth captions using *tf-idf* representations, as suggested by [10].

4 Experiments

We validated the proposed method on 2 tasks: (i) content based image retrieval (CBIR) based on semantics, where queries are related to the semantic content of the images – which is more general and harder than searching for visually similar images. We adopted the evaluation protocol proposed by Gordo *et al.* [10]. It uses captions as a proxy for semantic similarity and compares *tf-idf* representations of

captions, and measures the performance as the normalized discounted cumulative gain (NDCG), which can be seen as a weighted mean average precision, the relevance of one item with respect to the query being the dot product between their tf-idf representations. (ii) Multi-label image classification: each image can be assigned to different classes (labels), generally indicating the presence of the object type represented by the label. Per-class average precision and mAP are the performance metrics for this task.

Both series of experiments are done on the Visual Dialog dataset [1], relying on images of MS COCO [48]. Each image is annotated with 1 caption and 1 dialog (10 questions and answers), for a total of 1.2M questions-answers. Ground truth Dialog has been made in order to retrieve a query image from a pool of candidate images. A dialog should visually describe a query image and be suitable for retrieval and classification tasks. We use the standard train split for learning and validation split for testing, as the test set is not publicly available.

Our approach has several hyper-parameters: the word embedding size, $LSTM$ state size, learning rate, m. They are obtained through cross-validation. In this procedure, 20% of training data is considered as validation set, allowing to choose the hyper-parameters maximizing the NDCG/mean average precision on this so-obtained validation set. In practice, typical value for $LSTM$ state size (resp. embedding size) is 512 (resp. 200). The margin m is in the range [1.0–2.0]. Model parameters are initialized according to a centered Gaussian distribution ($\sigma = 0.02$). They are optimized with the Adam solver [49] with a cross-validated learning rate (typically of 10^{-4}), using mini-batches of size 128. In order to avoid over-fitting, we use dropout [50] for each layer (probability of a drop of 0.2 for the input layers and of 0.5 for the hidden layers). Both oracles (captioning and VQA) are fine-tuned on the tasks. Finally, while it would be interesting to average the performance on several runs, in order to evaluate the stability of the approach, this would be prohibitive in terms of computational time. In practice, we have observed that the performance is very stable and does not depend on initialization.

4.1 Experiments on Semantic Image Retrieval

We now evaluate our approach on semantic content-based retrieval, where the images sharing similar semantic content with an image query have to be returned by the system. As described before, the retrieval loss is optimized with triplets: an image query and two similar/dissimilar images. For triplet selection, we applied hard negative mining by sampling images according to the loss value (larger loss meaning higher probability to be selected). We found hard negative mining to be useful in our experiments.

Table 1 reports the NDCG performance for 3 values of R (R = k means that the top k images are considered for computing the NDCG), and the area under the curve (for R between 1 and 128) on 4 different models. The visual baseline exploits a similarity metric between image features extracted from the FC7 layer of a VGG19 network, which is learned on the train set using the same triplet approach as described in Sect. 3.6. $I+$ [10] corresponds to the visual embedding

Table 1. NDCG on semantic retrieval. Performance/Area Under the Curve for different values of R.

Method/R	8	32	128	AUC
$f_i(I)$ + ML (baseline)	45.8	51.7	59.3	69.7
I + [10]	47.6	55.9	62.3	72.7
$\{I, c(I)\}$ + [10]	57.0	58.5	63.3	75.1
Our approach $f_{enc}(I)$	**59.3**	**61.7**	**67.1**	**79.9**

Table 2. Semantic retrieval. NDCG/AUC after removing some components of the model.

Modality/R	8	32	128	AUC
$c(I)$	55.1	56.3	62.4	73.6
$\{Q_k, A_k\}_{1:10}$ generic	41.8	50.4	57.7	65.7
$\{Q_k, A_k\}_{1:10}$ task adapted	45.8	55.7	60.0	71.9
$tf - idf\{c(I), \{Q_k, A_k\}_{1:10}\}$	54.9	57.2	63.4	75.1
Our approach $f_{enc}(I)$	**59.3**	**61.7**	**67.1**	**79.9**

Table 3. Multi-label classification performance.

Modality	mAP
$f_i(I)$ (baseline)	61.1
$c(I)$	51.6
$\{Q_k, A_k\}_{1:10}$	49.9
$f_{enc}(I)$	56.0
$\{I, f_{enc}(I)\}$	**64.2**

noted (V, V) in [10]. $\{I, c(I)\}$ + [10] is the joint visual and textual embedding (V+T, V+T) with the difference that we don't feed the ground truth captions but the generated one, for fair comparison.

We observed that the area under the curve improves by +4.8% with our semantic bottleneck approach compared to the image feature similarity approach. We stress here that, unlike [10], we only exploit a semantic representation and not image features.

Empirical results of Table 2 show the usefulness of our semantic encoder. Indeed, with the same modalities (caption, questions and answers), $tf - idf\{c(I), \{Q_k, A_k\}_{1:10}\}$ performs 4.8% lower. Table 2 also shows the importance of adapting the VQA oracle to the task with +6.2% gain compared to a generic oracle not fine-tuned to the task.

4.2 Experiments on Multi-label Classification

With the MS COCO [48] dataset, each image is labeled with multi-object labels (80 object categories), representing the presence of broad concepts such as animal, vehicle, person, *etc.* in the image. For the baseline approach, we used image features provided by a VGG-VeryDeep-19 network [43] pre-trained on ImageNet [44] with weights kept frozen up to the 4,096-dim top-layer hidden unit activations (fc7), and fed to a final softmax layer learned on the common training set.

Table 3 (bottom) reports the per-class mean average precision for the visual baseline and the various components of our model. Our fully semantic approach $f_{enc}(I)$ underperforms only by 5% the baseline. This is quite encouraging as in

our setting the image is only encoded by a caption and 10 questions/answers. The main advantage of our model is that one can have access to the intermediate semantic representation for inspection, and may provide an explanation of the good or bad result (see Sect. 4.3). Figure 3 also reports the performance given by (i) generated caption $c(I)$ only, (ii) questions/answers $\{Q_k, A_k\}_{1:10}$ only. These experiments shows that captions are more discriminative than questions/answers (+1,7%), at least given the way they are generated. We also report the performance obtained by combining our image representation with image features (denoted as $\{I, f_{enc}(I)\}$). This configuration gives the best performance (+8,2%) and outperforms the baseline (+3.1%). As a sanity check, we also computed the mAP when using ground truth annotations for both the captions and the VQA. We obtained a performance of 72.2%, meaning that with good oracles it's possible for our semantic bottleneck to obtain a performance better than with images (61.1%).

Fig. 3. Combining captions and dialogs: query (top-left images), generated captions, task-specific dialogs, images retrieved using the caption (first rows) and those given by our model (second rows). Dialogs allowed to detect important complementary

Fig. 4. Adapting dialogs to tasks: query (top-left images), generated captions, generic and task-specific dialogs, images retrieved using the caption and generic dialog (first rows), and those given by our model (second rows).

4.3 Semantic Bottleneck Analysis

This section aims at giving some insights on (i) why the performance is improved by combining captions and dialogs and (ii) why making the semantic bottleneck adapted to the task improves the performance.

Regarding the first point, we did a qualitative analysis of the outputs of the semantic retrieval task, by comparing the relevance of the first ranked images when adding the dialogs to the captions (see Fig. 3). The Figure gives both the caption and the dialog automatically generated, as well as the images ranked first accordingly to the caption (first rows) and accordingly to our model combining the caption and the dialog. We marked in green the important complementary information added by the dialog. The dialog was able to detect drinks as an important feature of the image.

Table 4. Statistics of generated words after fine tuning the generators to the tasks

task/words	classes	playing	eating	wearing	doing	color	how many	in/outdoor	day/night
classification	88%	19%	39%	29%	14%	42%	39%	42%	53%
retrieval	78%	14%	28%	21%	16%	85%	81%	54%	79%

Regarding the second point, we compared the quality of the retrieval with and without adapting the dialogs to the task. Figure 4 illustrates our observations by giving both captions and dialogs automatically generated, as well as the images ranked first accordingly to captions combined with generic dialogs (first rows) and with task adapted dialogs (second rows). We marked in red the questions/answers that we found not relevant to the query in the generic dialog and in green those that have been given by the task adapted dialog to emphasize the complementary information they bring. The dialog was able to identify vehicles as an important feature of the image. Figure 5 illustrates the same type of caption correction by the dialog for the multi-label classification task.

Generated captions are, in general, brief and consistent with the images (see examples of Figs. 4, 5 and 7). Because we chose a simple sampling strategy (in order to have a trade-off between computation and interpretation) a few captions are syntactically incorrect. We argue that this should not impact the performance, as the generated captions reflect the image content. We also observed that several questions are repeated. While question repetition is not as critical as it is for the actual dialog generation, it can be overcome if needed by encoding the question history (h_k in Eq. (6)), for instance by explicitly penalizing repetitions in the LSTM criterion, or, by exploiting a reinforcement learning approach such as in [40].

Table 4 illustrates the effect of fine-tuning question generation by showing the percentage of time each word in the first row occurs in a dialog, across the two tasks ('classes' means any of the object class names). We observed that the generated dialogs of the classification task contain more verbs that can be associated to the presence of object classes (eating $\Rightarrow$ food classes, playing $\Rightarrow$ sport classes, wearing $\Rightarrow$ clothes classes). Generated dialogs for the retrieval task contain more words characterizing the scene (in/outdoor, day/night) or referencing specific object features (color, how many).

We also made experiments showing how the semantic bottleneck can be modified manually to make image search more interactive. Figure 6 shows an example

where we changed 2 oracle answers (zebras becomes cows and their number is increased by one). The 2nd row depicts the impact of this modification.

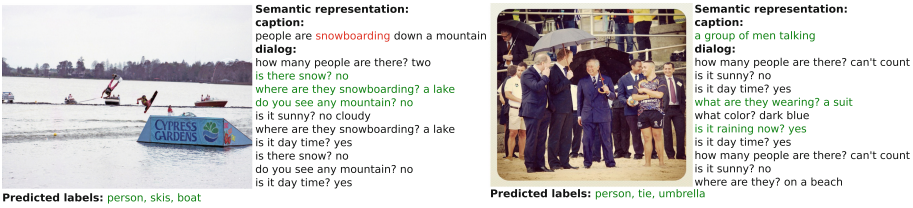


Fig. 5. Left-hand side: incorrect caption corrected by the dialog. Right-hand side: objects missing from the captions discovered by asking relevant questions.

Fig. 6. Incrementally updating the representation.

4.4 Evaluating Failure Predictions

The potential capacity of the semantic bottleneck to detect failure in the prediction process is illustrated by Fig. 7. Failure is detected when the representation contains incorrect semantic information—the caption or dialog are wrong—or insufficient information for further inference. We focus our evaluation on multi-label image classification, since a clear definition of failure in the case of content based image retrieval is complex, can be subjective (how decide if images are completely dissimilar from the request?) and task oriented (what are retrieved images used for?). We developed two evaluation protocols: one with humans in the loop, judging the semantic bottleneck capacity to predict consistent labels, and an automatic model optimized to predict success or failure for each class. We compare our approaches to a baseline based on score prediction thresholding. In order to evaluate the semantic bottleneck capacity, we first train our model for a multi-label classification task and extract the generated semantic representation (caption and dialog) and class prediction.

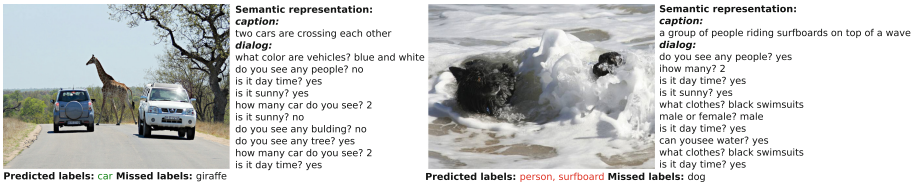


Fig. 7. Predicting failure cases from the proposed semantic representation. Left-hand side: caption and Q/A are consistent but not rich enough to predict the 'giraffe' label. Right-hand side: the semantic representation is incorrect leading to the inference of erroneous labels. In such cases, the bottleneck representation can be used for debugging.

Table 5. Failure prediction statistics.

	False negative		False positive	
	#true	#predicted	#true	#predicted
GT.	614	-	588	-
Users	308	379	213	485
Classifier	250	490	180	530

Table 6. Multi-label classification.

	Label		Image	
	mAP	%	mAP	%
No selection	54.3	100	54.3	100
Users	84.2	96	86.1	53
Classifier	79.8	93	81.7	49
conf. thresh	66.1	93	73.5	49

Human Based Failure Prediction Study. For 1000 randomly chosen test images, users were instructed to evaluate the capacity of the semantic bottleneck to contain enough information to predict the correct classes. The image and the generated semantic representation are shown to the users, which can select for each of the 80 labels of MS-COCO 1 among 3 cases: (i) *false negative*. The semantic representation missed the label (*e.g.* caption and dialog do not mention about the horse in the picture). (ii) *false positive*. The semantic representation hallucinates the object (*e.g.* seeing a car in a kitchen scene). (iii) *correct*. The algorithm has succeeded to predict the label, either its absence or its presence. Table 5 shows failure cases of the multi-label classification (614 false negative and 588 false positive). Human subjects were able to identify half of the failures (308/614 FN and 213/588 FP) with a precision of $\approx 60\%$ (308/379 and 213/485).

Failure detection can also be evaluated through two other sets of experiments: *Label rejection*: suspicious labels are rejected, others are kept. *Image rejection*: when there is a suspicious label, the image is rejected. Table 6 shows both experiments, and reads as follows: classification performance is of 54.3% when evaluating on 100% of the test set. When user rejects 4% of the labels, the performance goes to 84.2%. When our rejection algorithm keeps 93% of the data, the performance improves to 79.8%, which is close to human performance. We see a strong improvement for both our methods. Failure prediction improves the average precision of 30% percent with 4% of deleted image in average for each class.

We also proposed two automatic algorithms for failure prediction. The first one, referenced as 'classifier' in Tables 5 and 6, is based on an independent ternary linear classifier for each class with 3 possible outputs: `correct`, `FN`, `FP`. The input is the image I concatenated with the last hidden state of the semantic representation encoder $[\{Q_k, A_k\}_{k=1:K}, c(I)]$. The ground truth is built by comparing the output from the multi-label classification and the true classes. The model is optimized using a cross entropy loss. It is less accurate (can detect $\approx 41\%$ of false positive with $\approx 51\%$ of precision) but has the advantage of reducing human effort. We also show in the last row of Table 6 the performance of a second algorithm consisting in thresholding the confidence score outputted by the multi-label classifier for each label, and tuned to reach the same rejection rate as the other failure detection algorithm. This confidence thresholding algorithm gives a smaller performance increase after rejection.

5 Conclusions

In this paper we have introduced a novel method for representing images with semantic information expressed in natural language. Our primary motivation was to question the possibility of introducing an intelligible bottleneck in the processing pipeline. We showed that by combining and adapting several state-of-the-art techniques, our approach is able to generate rich textual descriptions that can be substituted for images in two vision tasks: semantic content based image retrieval, and multi-label classification. We quantitatively evaluated the usage of this semantic bottleneck as a diagnosis tool to detect failure in the prediction process, which we think contributes to a clearer metric of explainability, a key concern to mature artificial intelligence.

References

1. Das, A., et al.: Visual dialog. In: CVPR (2016)
2. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: an approach to evaluating interpretability of ML. arXiv:1806.00069 (2018)
3. Doshi-Velez, F., Kim, B.: A roadmap for a rigorous science of interpretability. arXiv:1702.08608 (2017)
4. Biran, O., Cotton, C.: Explanation and justification in ML: a survey. In: IJCAI (2017)
5. Ras, G., Haselager, P., van Gerven, M.: Explanation methods in deep learning: users, values, concerns and challenges. arXiv:1803.07517 (2018)
6. Zhang, P., Wang, J., Farhadi, A., Hebert, M., Parikh, D.: Predicting failures of vision systems. In: CVPR (2014)
7. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. TPAMI (2014)
8. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge. TPAMI (2017)

9. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: adaptive attention via a visual sentinel for image captioning. In: CVPR (2017)
10. Gordo, A., Larlus, D.: Beyond instance-level image retrieval: leveraging captions to learn a global visual representation for semantic retrieval. In: CVPR (2017)
11. Dai, B., Lin, D., Urtasun, R., Fidler, S.: Towards diverse and natural image descriptions via a conditional GAN. In: CVPR (2017)
12. Dai, B., Lin, D.: Contrastive learning for image captioning. In: NIPS (2017)
13. Seo, P.H., Lehrmann, A., Han, B., Sigal, L.: Visual reference resolution using attention memory for visual dialog. In: NIPS (2017)
14. Lu, J., Kannan, A., Yang, J., Parikh, D., Batra, D.: Best of both worlds: transferring knowledge from discriminative learning to a generative visual dialog model. In: NIPS (2017)
15. Malinowski, M., Rohrbach, M., Fritz, M.: Ask your neurons: a neural-based approach to answering questions about images. In: ICCV (2015)
16. Lin, X., Parikh, D.: Leveraging visual question answering for image-caption ranking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 261–277. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_17
17. Zhu, Y., Groth, O., Bernstein, M., Fei-Fei, L.: Visual7w: grounded question answering in images. In: CVPR (2016)
18. Jabri, A., Joulin, A., van der Maaten, L.: Revisiting VQA baselines. In: ECCV (2016)
19. Wu, Q., Wang, P., Shen, C., Dick, A., van den Hengel, A.: Ask me anything: free-form visual question answering based on knowledge from external sources. In: CVPR (2016)
20. Lipton, Z.C.: The mythos of model interpretability. In: ICML (2016)
21. Doran, D., Schulz, S., Besold, T.R.: What does explainable AI really mean? A new conceptualization of perspectives. arXiv:1710.00794 (2017)
22. Hohman, F.M., Kahng, M., Pienta, R., Chau, D.H.: Visual analytics in deep learning: an interrogative survey for the next frontiers. In: TVCG (2018)
23. Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., Giannotti, F.: A survey of methods for explaining black box models. arXiv:1802.01933 (2018)
24. Zhang, Q., Yang, Y., Wu, Y.N., Zhu, S.C.: Interpreting CNNs via decision trees. arXiv:1802.00121 (2018)
25. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
26. Rajani, N.F., Mooney, R.J.: Using explanations to improve ensembling of visual question answering systems. In: IJCAI (2017)
27. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digit. Sign. Process. Rev. J. (2018)
28. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 3–19. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_1
29. Park, D.H., et al.: Multimodal explanations: justifying decisions and pointing to the evidence. In: CVPR (2018)
30. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: CVPR (2015)

31. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
32. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: CVPR (2016)
33. Olah, C., et al.: The building blocks of interpretability. Distill (2018)
34. Zhang, Q., Cao, R., Shi, F., Wu, Y.N., Zhu, S.C.: Interpreting CNN knowledge via an explanatory graph. arXiv:1708.01785 (2017)
35. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: quantifying interpretability of deep visual representations. In: CVPR (2017)
36. Kindermans, P.J., et al.: The (un)reliability of saliency methods. arXiv:1711.00867 (2017)
37. Serban, I.V., et al.: Generating factoid questions with recurrent neural networks: the 30M factoid question-answer corpus. In: ACL (2016)
38. Li, Y., Huang, C., Tang, X., Change Loy, C.: Learning to disambiguate by asking discriminative questions. In: CVPR (2017)
39. Mostafazadeh, N., Misra, I., Devlin, J., Mitchell, M., He, X., Vanderwende, L.: Generating natural questions about an image. In: ACL (2016)
40. Das, A., Kottur, S., Moura, J.M.F., Lee, S., Batra, D.: Learning cooperative visual dialog agents with deep reinforcement learning. In: ICCV 2017 (2017)
41. Ganju, S., Russakovsky, O., Gupta, A.: What's in a question: using visual questions as a form of supervision. In: CVPR (2017)
42. Zhu, Y., Lim, J.J., Fei-Fei, L.: Knowledge acquisition for visual question answering via iterative querying. In: CVPR (2017)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2014)
44. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
45. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. (1997)
46. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR (2013)
47. Antol, S., et al.: VQA: visual question answering. In: ICCV (2015)
48. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
49. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
50. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. JMLR (2014)