University of Modena and Reggio Emilia

XXXV cycle of the International Doctorate School in
Information and Communication Technologies

Doctor of Philosophy dissertation in
Computer Engineering and Science

Transforming Vision and Language with Attention

Image captioning, cross-modal retrieval and gene expression

Matteo Stefanini

Supervisor: Prof. Rita Cucchiara
PhD Course Coordinator: Prof. Sonia Bergamaschi

Modena, 2023

Review committee composed of:

Prof. Fabrizio Falchi, University of Pisa and ISTI-CNR
Dr. Giuseppe Fiameni, NVIDIA

To my Family.

Contents

Chapter 1

Introduction

Attention mechanism and Transformer-based architectures have recently revolutionized the artificial intelligence landscape in almost every field. Ever since their first introduction, they have become ubiquitous components of any deep learning breakthrough, from Natural Language Processing to Computer Vision and Bioinformatics. This boils down mainly to their superior abilities in dealing with long-range interactions across data. In this thesis, I investigate the frontier of Transformer-based architectures at the intersection of Vision and Language, where machines are required to replicate the human ability to semantically connect different domains.

In the first part, we present state-of-the-art solutions for the image captioning task, which consist of automatically describing images with natural language sentences, from the understanding of the visual content, objects and their interactions, to the creation of a syntactically and semantically correct sentence. We first discuss a thorough literature survey in the deep learning era, and we propose a novel image captioning model among the firsts embracing self-attention in place of recurrent networks. Experimentally, our architecture reaches a new state of the art, achieving the first place of the public leaderboard on the most important captioning benchmark.

Further, we explore new training strategies proposing a method based on the interplay between two distinct language models, using the mean teacher paradigm and knowledge distillation, providing state-of-the-art caption quality with a reduced number of parameters. Despite the remarkable results obtained by captioning models, switching to real-life scenarios constitutes a challenge due to the larger variety of visual concepts not covered in existing datasets. For this reason, we propose a novel approach for novel object captioning, that learns to select the most relevant objects of an image, regardless of their presence in the training set, and constrains the generative process accordingly.

In the following, we present solutions for cross-modal retrieval, another task related to vision and language that consists of finding images corresponding to a given textual query and, vice versa, retrieving texts which describe a given query image. Since both images and texts are usually encoded as sets or sequences of elements, we propose an attentive reduction method that transforms a set of elements into a single response, leading to a performance increase. Moreover, we propose an efficient Transformer architecture to fill in the gap between effectiveness and efficiency by learning a shared embedding space and distilling fine-grained scores previously aligned. Our approach competes with state-of-the-art large models while being almost 90 times faster. Switching to more complex and challenging scenarios, we also investigate visual-semantic models in the artistic and digital humanities domain. To this aim, we propose a cross-modal retrieval method that also identifies if sentences describe the visual content or the context of a painting and a visual-semantic embedding that can automatically align illustrations and texts without paired supervision.

Finally, we expand the scope of attentive models to the language of life: the genetic code. We propose a new class of deep learning models based on the Perceiver architecture, built upon Transformer, which leverages asymmetric attention and can scale to longer sequences. We present a model able to predict the gene expression (mRNA level) given its DNA sequence, and a model for the first time predicting the protein expression given its amino-acid sequence. We demonstrate the effectiveness of our methods and promising future opportunities.

Activities carried out during the Ph.D.

Beside the research activities described in this thesis, I also took part in other teaching and service activities, which are reported below together with a list of attended conferences and schools. The complete list of my publications is instead reported in Appendix A.

Teaching activities

- Laboratory Tutor for the Vision and Cognitive Systems graduate course, at University of Modena and Reggio Emilia (2020, 2021).
- Laboratory Tutor of the Neural Network Computing, AI and Machine Learning for Automotive graduate course, at University of Modena and Reggio Emilia (2021)

Participation to national projects

- “AI4CH” AI for Cultural Heritage project. Funded by the Minister of Foreign Affairs and International Cooperation
- NVIDIA AI Technology Center project.
- “IDEHA” - Innovation for Data Elaboration in Heritage Areas, co-funded by the European Union - FESR and FSE.
- “CultMEDIA” project of the National Technological Cluster on Technologies for the Cultural Heritage, co-funded by the Italian Ministry of Education, University and Research.

Journals and Conferences reviewing

- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)
- IEEE Transactions on Multimedia
- IEEE International Conference on Pattern Recognition (ICPR)
- Pattern Recognition Letters (PRL)
- ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)
- ACM International Conference on Multimedia (ACMMM)
- International Journal of Computer Vision (IJCV)
- International Journal of Computer Assisted Radiology and Surgery (IJCARS)

Other academic services

- Doctoral Student Representative for the International Doctorate in ICT at University of Modena and Reggio Emilia (2020-2023)
- Program Committee of FAPER International Workshop at ICPR Milan (2020)

Conferences and schools attended

- ACM International Conference on Multimedia Retrieval (remotely) 2021.
- International Conferences on Pattern Recognition (remotely), 2020.
- International Conference on Image Analysis and Processing (Trento, Italy), 2019.
- PhD School on Advanced Topic in Deep Learning, with Dr. Timothy M. Hospedales and Dr. Henry Gouk. University of Verona. 2019.

Seminars and courses attended

- “Talents for Open Innovation I & II” ATER December, 2020.
- “Public speaking & presentation making” - EROI S3 PhD Video Contest November, 2020.
- “About Time” - Prof. Arnold Smeulders (University of Amsterdam) - September 30, 2020.
- “Academic English Workshop II” – Dott. Silvia Cavalieri July, 2020.
- “Academic English Workshop I” – Dott. Silvia Cavalieri June, 2020.
- “Data Science and Machine Learning: basics and applications to Health Care” - Dr. Paolo Missier 2019-2020.
- “Research, innovation and new challenges of ICT for the biomedical” - Prof. Luigi Rovati December 19, 2019.
- “Complementary Training Courses For PhD Students And Research Fellows” November, 2019.

Master thesis co-advising

- Manuele Barraco "Enhancing Image Captioning through Mean Teacher and Knowledge Distillation", 2022.
- Marco Cagrandi "Novel Object Captioning: Describe objects not present in the training set", 2021.
- Alessio Ruggi "Exploring the Transformer model: a Fully-Attentive architecture for Handwritten Text recognition of modern and historical manuscripts", 2021.
- Paolo Benetti "Stacked Cross-Attention per Visual Question Answering", 2020.

Chapter 2

Image captioning

A fundamental challenge in artificial intelligence and computer vision is that of creating a system able to replicate the human ability of understanding a visual stimuli and describing it in natural language. Indeed, this would open up to new advancements in human-machine interaction and collaboration, bringing a great impact on society. Recent improvements in computer vision and natural language processing, along with the availability of larger datasets, have made it possible, today, to automatically generate sentences describing images with an incredibly high efficacy and reliability.

This task, called image captioning, has recently gained lots of attention thanks to the adoption of deep learning approaches, which improved the performances of algorithms and can effectively describe images in natural language [311, 159, 341, 312, 65, 196]. Image captioning architectures are capable of learning a correspondence between an input image and a probability distribution over time, which can be sampled to generate captions either using a greedy decoding [312], or more elaborated procedures like beam search and its variants [6].

Contributions

In this chapter, we first present a thorough overview of the literature on the image captioning task from the advent of the deep learning era; in the following, we present two different solutions to address the task itself. The first method is based

This chapter is related to publications [4, 6, 7, 8] reported in Appendix A, by the author of the thesis. See Appendix A for details.

on the Transformer model, thus replacing the recurrent relations in favour of the use of fully-attentive mechanisms. The proposed architecture, called Meshed-Memory Transformer, improves both the image encoding and the language generation steps: it learns a multi-level representation of the relationships between image regions integrating learned a priori knowledge, and uses a mesh-like connectivity at decoding stage to exploit low- and high-level features. Experimentally, we investigate the performance of our solution and different fully-attentive models in comparison with recurrent ones. We show that our model achieves a new state of the art on the most important dataset for image captioning (*i.e.* COCO [201]) reaching the first place on the leaderboard of the online test server[1].

While a typical model for image captioning is composed by one language model, our second approach is instead based on the interaction of two interconnected language models, that learn from each other during the training phase. Our approach, called CaMEL, leverages the interplay between the two language models following a mean teacher learning paradigm with knowledge distillation. Experimentally, we assess the effectiveness of our solution in conjunction with different visual feature extractors. When comparing with existing methods, we demonstrate that our model provides state-of-the-art caption quality with a significantly reduced number of parameters. According to the CIDEr metric, we obtain a new state of the art on COCO [201] when training without using external data. Further, although captioning models have achieved impressive results, describing the large variety of visual concepts present in real-life scenarios is still very challenging. For this reason, we propose a novel method for novel object captioning, a variant of the task that consist of describing novel objects unseen during the training phase. Our model learns to select the most relevant objects of an image, regardless of their presence in the training set, and constrains the generative process accordingly.

In details, the rest of the chapter is organized as follows: in Sec. 2.1, we present the image captioning problem with a comprehensive survey on the task since the first deep learning approaches. We explore the most important architectures in literature together with a quantitative experimental analysis. Subsequently, in Sec. 2.2 and in Sec. 2.3 we describe our solutions, respectively Meshed-Memory Transformer and CaMEL, and we show their effectiveness with quantitative and qualitative experiments, where both methods achieve a new state of the art on standard image captioning. Finally, in Sec. 2.4 we present our approach for novel object captioning, along with experiments showing its superior performances.

[1] https://competitions.codalab.org/competitions/3221

2.1 Deep learning survey

2.1.1 Introduction

Image captioning is the task of describing the visual content of an image in natural language, employing a visual understanding system and a language model capable of generating meaningful and syntactically correct sentences. Neuroscience research has clarified the link between human vision and language generation only in the last few years [13]. Similarly, in Artificial Intelligence, the design of architectures capable of processing images and generating language is a very recent matter. The goal of these research efforts is to find the most effective pipeline to process an input image, represent its content, and transform that into a sequence of words by generating connections between visual and textual elements while maintaining the fluency of language.

The early-proposed approaches to image captioning have entailed description retrieval [236, 87, 235, 94, 172, 160] or template filling and hand-crafted natural language generation techniques [355, 3, 351, 193, 116, 230, 178, 181]. While these have been treated in other surveys [24, 16, 131], image captioning is currently based on the usage of deep learning-based generative models. In its standard configuration, the task is an image-to-sequence problem whose inputs are pixels. These inputs are encoded as one or multiple feature vectors in the visual encoding step, which prepares the input for a second generative step, called the language model. This produces a sequence of words or sub-words decoded according to a given vocabulary.

In these few years, the research community has improved model design considerably: from the first deep learning-based proposals adopting Recurrent Neural Networks (RNNs) fed with global image descriptors, methods have been enriched with attentive approaches and reinforcement learning up to the breakthroughs of Transformers and self-attention and single-stream BERT-like approaches. At the same time, the Computer Vision and Natural Language Processing (NLP) communities have addressed the challenge of building proper evaluation protocols and metrics to compare results with human-generated ground-truths. However, despite the investigation and improvements achieved in these years, image captioning is still far from being considered a solved task.

Several domain-specific proposals and variants of the task have also been investigated to accommodate for different user needs and descriptions styles. According to [128, 269], indeed, image captions can be perceptual, when focusing on low-level visual attributes; non-visual, when reporting implicit and contextual information; conceptual, when describing the actual visual content (*e.g.* visual entities and their relations). While the latter is commonly recognized as the target of the image captioning task, this definition encompasses descriptions focusing on different aspects and at various levels of detail (*e.g.* including attributes or not, mentioning named entities or high-level concepts only, describing salient parts only, or also finer details).

With the aim of providing a testament to the journey that captioning has taken so far, and with that of encouraging novel ideas, we trace a holistic overview of techniques, models, and task variants developed in the last years. Furthermore, we review datasets and evaluation metrics and perform quantitative comparisons of the main approaches. Finally, we discuss open challenges and future directions.

To sum up, the contributions of this section are as follows:

- Following the inherent dual nature of captioning models, we develop taxonomies for visual encoding and language modeling approaches and describe their key aspects and limitations.
- We review the training strategies adopted in the literature over the past years and the recent advancement obtained by the pre-training paradigm and masked language model losses.
- We review the main datasets used to explore image captioning, both domain-generic benchmarks and domain-specific datasets collected to investigate specific aspects.
- We analyze both standard and non-standard metrics adopted for performance evaluation and the characteristics of the caption they highlight.
- We present a quantitative comparison of the main image captioning methods considering both standard and non-standard metrics and a discussion on their relationships, which sheds light on performance, differences, and characteristics of the most important models.
- We give an overview of many variants of the task and discuss open challenges and future directions.

Compared to previous surveys on image captioning [131, 16, 209, 271, 24], we provide a comprehensive and updated view on deep learning-based generative captioning models. We perform a deeper analysis of proposed approaches and survey a considerably larger number of papers on the topic. Also, we cover non-standard evaluation metrics, which are disregarded by other works, discuss their characteristics, and employ them in a quantitative evaluation of state-of-the-art methods. Moreover, we tackle emerging variants of the task and a broader set of available datasets.

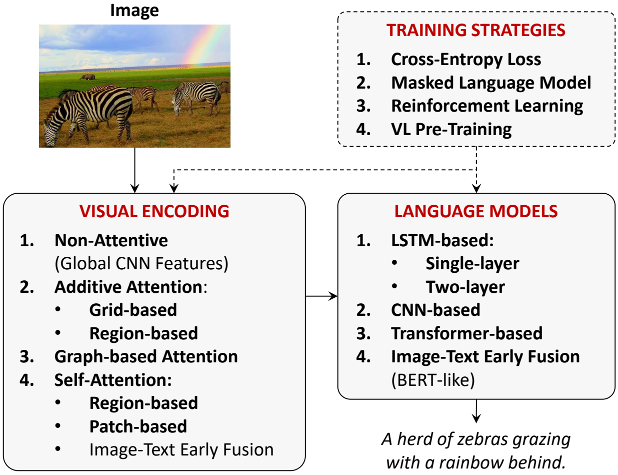


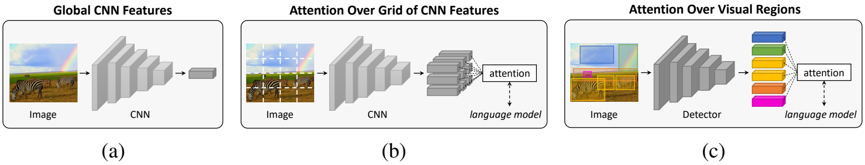
Figure 2.1: Overview of the image captioning task and taxonomy of the most relevant approaches.

2.1.2 Visual encoding

Providing an effective representation of the visual content is the first challenge of an image captioning pipeline. The current approaches for visual encoding can be classified as belonging to four main categories: 1. *non-attentive methods* based on global CNN features; 2. *additive attentive methods* that embed the visual content using either grids or regions; 3. *graph-based methods* adding visual relationships between visual regions; and 4. *self-attentive methods* that employ Transformer-based paradigms, either by using region-based, patch-based, or image-text early fusion solutions. This taxonomy is visually summarized in Fig. 2.1.

Global CNN Features

With the advent of CNNs, all models consuming visual inputs have been improved in terms of performance. The visual encoding step of image captioning is no exception. In the most simple recipe, the activation of one of the last layers of a CNN is employed to extract high-level representations, which are then used as a conditioning element for the language model (Fig. 2.2a). This is the approach employed in the seminal “Show and Tell” paper [311],where the output of GoogleNet [292] is fed to the initial hidden state of the language model. In the same year,



(a) (b) (c)

Figure 2.2: Three of the most relevant visual encoding strategies for image captioning: **(a)** global CNN features; **(b)** fine-grained features extracted from the activation of a convolutional layer, together with an attention mechanism guided by the language model; **(c)** image region features coming from a detector, together with an attention mechanism.

Karpathy *et al.* [159] used global features extracted from AlexNet [176] as the input for a language model. Further, Mao *et al.* [222] and Donahue *et al.* [79] injected global features extracted from the VGG network [280] at each time-step of the language model.

Global CNN features were then employed in a large variety of image captioning models [47, 86, 150, 362, 332, 109, 40, 41]. Notably, Rennie *et al.* [262] introduced the FC model, in which images are encoded using a ResNet-101 [119], preserving their original dimensions. Other approaches [359, 96] integrated high-level attributes or tags, represented as a probability distribution over the most common words of the training captions.

The main advantage of employing global CNN features resides in their simplicity and compactness of representation, which embraces the capacity to extract and condense information from the whole input and to consider the overall context of an image. However, this paradigm also leads to excessive compression of information and lacks granularity, making it hard for a captioning model to produce specific and fine-grained descriptions.

Attention Over Grid of CNN Features

Motivated by the drawbacks of global representations, most of the following approaches have increased the granularity level of visual encoding [341, 262, 214]. For instance, Dai *et al.*[72] have employed 2D activation maps in place of 1D global feature vectors to bring spatial structure directly in the language model. Drawing from machine translation literature, a big portion of the captioning community has instead employed the additive attention mechanism (Fig. 2.2b), which has endowed image captioning architectures with time-varying visual features encoding, enabling greater flexibility and finer granularity.

Definition of additive attention. The intuition behind attention boils down to weighted averaging. In the first formulation proposed for sequence alignment by Bahdanau *et al.* [15] (also known as *additive attention*), a single-layer feed-forward neural network with a hyperbolic tangent non-linearity is used to compute attention weights. Formally, given two generic sets of vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and $\{\mathbf{h}_1, \dots, \mathbf{h}_m\}$, the additive attention score between $\mathbf{h}_i$ and $\mathbf{x}_j$ is computed as follows:

$$f_{\text{att}}\left(\mathbf{h}_i, \mathbf{x}_j\right) = \mathbf{W}_3^\top \tanh\left(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{x}_j\right), \tag{2.1}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, and $\mathbf{W}_3$ is a weight vector that performs a linear combination. A softmax function is then applied to obtain a probability distribution $p\left(\mathbf{x}_j \mid \mathbf{h}_i\right)$, representing how much the element encoded by $\mathbf{x}_j$ is relevant for $\mathbf{h}_i$.

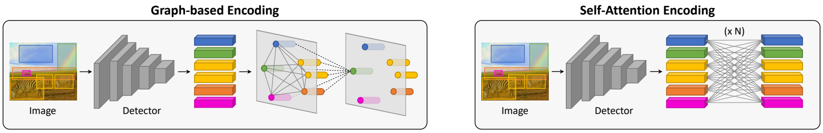
Although the attention mechanism was initially devised for modeling the relationships between two sequences of elements (*i.e.* hidden states from a recurrent encoder and a decoder), it can be adapted to connect a set of visual representations with the hidden states of a language model.

Attending convolutional activations. Xu *et al.* [341] introduced the first method leveraging the additive attention over the spatial output grid of a convolutional layer. This allows the model to selectively focus on certain elements of the grid by selecting a subset of features for each generated word. Specifically, the model first extracts the activation of the last convolutional layer of a VGG network [280], then uses additive attention to compute a weight for each grid element, interpreted as the relative importance of that element for generating the next word.

Other approaches. The solution based on additive attention over a grid of features has been widely adopted by several following works with minor improvements in terms of visual encoding [359, 48, 214, 326, 102, 106].

Review networks – For instance, Yang *et al.* [353] supplemented the encoder-decoder framework with a recurrent review network. This performs a given number of review steps with attention on the encoder hidden states and outputs a “thought vector” after each step, which is then used by the attention mechanism in the decoder.

Multi-level features – Chen *et al.* [43] proposed to employ channel-wise attention over convolutional activations, followed by a more classical spatial attention. They also experimented with using more than one convolutional layer to exploit multi-level features. On the same line, Jiang *et al.* [154] proposed to use



(a)

(b)

Figure 2.3: Summary of the two most recent visual encoding strategies for image captioning: **(a)** graph-based encoding of visual regions; **(b)** self-attention-based encoding over image region features.

multiple CNNs in order to exploit their complementary information, then fused their representations with a recurrent procedure.

Exploiting human attention – Some works also integrated saliency information (*i.e.* what do humans pay more attention to in a scene) to guide caption generation with stimulus-based attention. This idea was first explored by Sugano and Bulling [289] who exploited human eye fixations for image captioning by including normalized fixation histograms over the image as an input to the soft-attention module of [341] and weighing the attended image regions based on whether these are fixated or not. Subsequent works on this line [295, 252, 62, 44] employed saliency maps as a form of additional attention source.

Attention Over Visual Regions

The intuition of using saliency boils down to neuroscience, which suggests that our brain integrates a top-down reasoning process with a bottom-up flow of visual signals. The top-down path consists of predicting the upcoming sensory input by leveraging our knowledge and inductive bias, while the bottom-up flow provides visual stimuli adjusting the previous predictions. Additive attention can be thought of as a top-down system. In this mechanism, the language model predicts the next word while attending a feature grid, whose geometry is irrespective of the image content.

Bottom-up and top-down attention. Differently from saliency-based approaches [44], in the solution proposed by Anderson *et al.* [7] the bottom-up path is defined by an object detector in charge of proposing image regions. This is then coupled with a top-down mechanism that learns to weigh each region for each word prediction (see Fig. 2.2c). In this approach, Faster R-CNN [259, 260] is adopted to detect objects, obtaining a pooled feature vector for each region proposal. One of the key elements of this approach resides in its pre-training strategy, where an auxiliary training loss is added for learning to predict attribute classes alongside object classes on the Visual Genome [175] dataset. This allows the model to predict a dense and rich set of detections, including both salient object and contextual regions, and favors the learning of better feature representations.

Other approaches. Employing image region features has demonstrated its advantages when dealing with the raw visual input and has been the standard de-facto in image captioning for years. As a result, many of the following works have based the visual encoding phase on this strategy [162, 248, 138, 318]. Among them, we point out two remarkable variants.

Visual Policy – While typical visual attention points to a single image region at every step, the approach proposed by Zha *et al.* [366] introduces a sub-policy network that interprets also the visual part sequentially by encoding historical visual actions (*e.g.* previously attended regions) via an LSTM to serve as context for the next visual action.

Geometric Transforms – Pedersoli *et al.* [241] proposed to use spatial transformers for generating image-specific attention areas by regressing region proposals in a weakly-supervised fashion. Specifically, a localization network learns an affine transformation or each location of the feature map, and then a bilinear interpolation is used to regress a feature vector for each region with respect to anchor boxes.

Graph-based Encoding

To further improve the encoding of image regions and their relationships, some studies consider using graphs built over image regions (see Fig. 2.3a) to enrich the representation by including semantic and spatial connections.

Spatial and semantic graphs. The first attempt in this sense is due to Yao *et al.* [357], followed by Guo *et al.* [111], who proposed the use of a graph convolutional network (GCN) [171] to integrate both semantic and spatial relationships between objects. The semantic relationships graph is obtained by applying a classifier pre-trained on Visual Genome [175] that predicts an action or an interaction between object pairs. The spatial relationships graph is instead inferred through geometry measures (*i.e.* intersection over union, relative distance, and angle) between bounding boxes of object pairs.

Scene graphs. With a focus on modeling semantic relations, Yang *et al.* [347] proposed to integrate semantic priors learned from text in the image encoding by exploiting a graph-based representation of both images and sentences. The representation used is the scene graph, *i.e.* a directed graph connecting the objects, their attributes, and their relations. On the same line, Shi *et al.* [277] represented the image as a semantic relationship graph but proposed to train the module in charge of predicting the predicate nodes directly on the ground-truth captions rather than on external datasets.

Hierarchical trees. As a special case of a graph-based encoding, Yao *et al.* [358] employed a tree to represent the image as a hierarchical structure. The root represents the image as a whole, intermediate nodes represent image regions and their contained sub-regions, and the leaves represent segmented objects in the regions.

Graph encodings brought a mechanism to leverage relationships between detected objects, which allows the exchange of information in adjacent nodes and thus in a local manner. Further, it seamlessly allows the integration of external semantic information. On the other hand, manually building the graph structure can limit the interactions between visual features. This is where self-attention proved to be more successful by connecting all the elements with each other in a complete graph representation.

Self-Attention Encoding

Self-attention is an attentive mechanism where each element of a set is connected with all the others, and that can be adopted to compute a refined representation of the same set of elements through residual connections (Fig. 2.3b). It was first introduced by Vaswani *et al.* [306] for machine translation and language understanding tasks, giving birth to the Transformer architecture and its variants, which have dominated the NLP field and later also Computer Vision.

Definition of self-attention. Formally, self-attention makes use of the scaled dot-product mechanism, *i.e.* a multiplicative attention operator that handles three sets of vectors: a set of n_q query vectors $\boldsymbol{Q}$, a set of key vectors $\boldsymbol{K}$, and a set of value vectors $\boldsymbol{V}$, both containing n_k elements. The operator takes a weighted sum of value vectors according to a similarity distribution between query and key vectors:

$$\mathsf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{2.2}$$

where d_k is a scaling factor. In the case of self-attention, the three sets of vectors are obtained as linear projections of the same input set of elements. The success of the Transformer demonstrates that leveraging self-attention allows achieving superior performances compared to attentive RNNs.

Early self-attention approaches. Among the first image captioning models leveraging this approach, Yang *et al.* [348] used a self-attentive module to encode relationships between features coming from an object detector. Later, Li *et al.* [190] proposed a Transformer model with a visual encoder for the region features coupled with a semantic encoder that exploits knowledge from an external tagger. Both encoders are based on self-attention and feed-forward layers. Their output is then fused through a gating mechanism governing the propagation of visual and semantic information.

Variants of the self-attention operator. Other works proposed variants or modifications of the self-attention operator tailored for image captioning [124, 115, 137, 237, 65].

Geometry-aware encoding – Herdade *et al.* [124] introduced a modified version of self-attention that takes into account the spatial relationships between regions. In particular, an additional geometric weight is computed between object pairs and is used to scale the attention weights. On a similar line, Guo *et al.* [115] proposed a normalized and geometry-aware version of self-attention that makes use of the relative geometry relationships between input objects. Further, He *et al.* [120] introduced a spatial graph transformer, which considers different categories of spatial relationship between detections (*e.g.*, parent, neighbor, child) when performing attention.

Attention on Attention – Huang *et al.* [137] proposed an extension of the attention operator in which the final attended information is weighted by a gate guided by the context. Specifically, the output of the self-attention is concatenated with the queries, then an information and a gate vector are computed and finally multiplied together. In their encoder, they employed this mechanism to refine the visual features. This method is then adopted by later models such as [203].

X-Linear Attention – Pan *et al.* [237] proposed to use bilinear pooling techniques to strengthen the representative capacity of the output attended feature. Notably, this mechanism encodes the region-level features with higher-order interaction, leading to a set of enhanced region-level and image-level features.

Memory-augmented Attention – Cornia *et al.* [65, 58] proposed a Transformer-based architecture where the self-attention operator of each encoder layer is augmented with a set of memory vectors. Specifically, the set of keys and values is extended with additional "slots" learned during training, which can encode multi-level visual relationships. More details about this model are presented in the

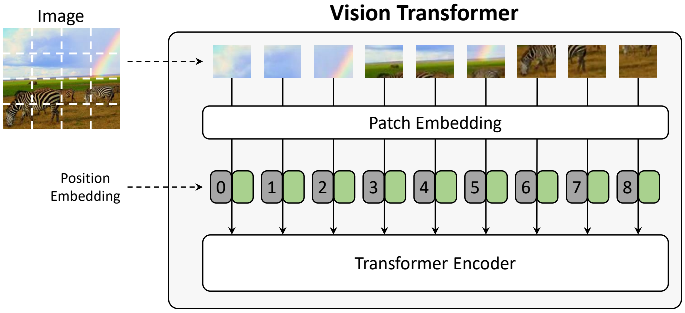


Figure 2.4: Vision Transformer encoding. The image is split into fixed-size patches, linearly embedded, added to position embeddings, and fed to a standard Transformer encoder.

following section of this thesis.

Other self-attention-based approaches. Ji *et al.* [148] proposed to improve self-attention by adding to the sequence of feature vectors a global vector computed as their average. A global vector is computed for each layer, and the resulting global vectors are combined via an LSTM, thus obtaining an inter-layer representation. Luo *et al.* [217] proposed a hybrid approach that combines region and grid features to exploit their complementary advantages. Two self-attention modules are applied independently to each kind of features, and a cross-attention module locally fuses their interactions. On a different line, the architecture proposed by Liu *et al.* [202] is based on an attention module to align grid or detection features with visual words extracted from a concept extractor and to obtain semantic-grounded encodings.

Attention on grid features and patches. Other than applying the attention operator on detections, the role of grid features has been recently re-evaluated [151]. For instance, the approach proposed by Zhang *et al.* [375] applies self-attention directly to grid features, incorporating their relative geometry relationships into self-attention computation. Transformer-like architectures can also be applied directly on image patches, thus excluding the usage of the convolutional operator [80, 301] (Fig. 2.4). On this line, Liu *et al.* [208] devised the first convolution-free architecture for image captioning. Specifically, a pre-trained Vision Transformer network (*i.e.* ViT [80]) is adopted as encoder, and a standard Transformer decoder is employed to generate captions. Interestingly, the same visual encoding approach has been adopted in CLIP [250] and SimVLM [328], with the difference that the visual encoder is trained from scratch on large-scale noisy data. CLIP-based features have then been used by subsequent captioning approaches [273, 232, 60].

Early fusion and vision-and-language pre-training. Other works using self-attention to encode visual features achieved remarkable performance also thanks to vision-and-language pre-training [293, 212] and early-fusion strategies [196, 381]. For example, following the BERT architecture [78], Zhou *et al.* [381] combined encoder and decoder into a single stream of Transformer layers, where region and word tokens are early fused together into a unique flow. This unified model is first pre-trained on large amounts of image-caption pairs to perform both bidirectional and sequence-to-sequence prediction tasks and then fine-tuned.

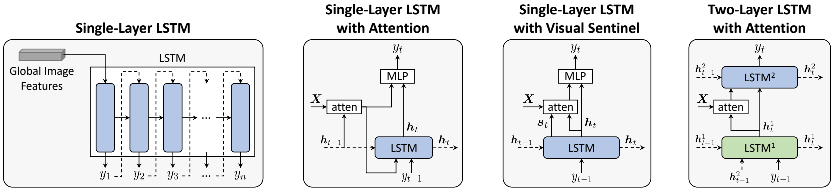
On the same line, Li *et al.* [196] proposed OSCAR, a BERT-like architecture that includes object tags as anchor points to ease the semantic alignment between images and text. They also performed a large-scale pre-train with 6.5 million image-text pairs, with a masked token loss similar to the BERT mask language loss and a contrastive loss for distinguishing aligned words-tags-regions triples from polluted ones. Later, Zhang *et al.* [371] proposed VinVL, built on top of OSCAR, introducing a new object detector capable of extracting better visual features and a modified version of the vision-and-language pre-training objectives. On this line, Hu *et al.* [134] improved the VinVL model by scaling up its size and using larger scale noisy data to pre-train.

Discussion

After the emergence of global features and grid features, region-based features have been the state-of-the-art choice in image captioning for years thanks to their compelling performances. Recently, however, different factors are reopening the discussion on which feature model is most appropriate for image captioning, ranging from the performance of better-trained grid features [151] to the emergence of self-attentive visual encoders [80] and large-scale multi-modal models like CLIP [250]. Recent strategies encompass training better object detectors on large-scale data [371] or employing end-to-end visual models trained from scratch [328]. Moreover, the success of BERT-like solutions performing image and text early-fusion indicates the suitability of visual representations that also integrate textual information.

2.1.3 Language models

The goal of a language model is to predict the probability of a given sequence of words to occur in a sentence. As such, it is a crucial component in image



(a) (b) (c) (d)

Figure 2.5: LSTM-based language modeling strategies: **(a)** Single-Layer LSTM model conditioned on the visual feature; **(b)** LSTM with attention, as proposed in [341]; **(c)** LSTM with attention, in the variant proposed in [214]; **(d)** two-layer LSTM with attention, in the style of the bottom-up top-down approach [7]. In all figures, $\boldsymbol{X}$ represents a set of visual features, $\boldsymbol{h}_t$ is the LSTM hidden state at time t, and $\boldsymbol{s}_t$ is the visual sentinel.

captioning, as it gives the ability to deal with natural language as a stochastic process.

Formally, given a sequence of n words, the language model component of an image captioning algorithm assigns a probability $P\left(y_1, y_2, \ldots, y_n \mid \boldsymbol{X}\right)$ to the sequence as:

$$P\left(y_1, y_2, \ldots y_n \mid \boldsymbol{X}\right) = \prod_{t=1}^{n} P\left(y_t \mid y_1, y_2, \ldots, y_{t-1}, \boldsymbol{X}\right), \tag{2.3}$$

where $\boldsymbol{X}$ represents the visual encoding on which the language model is specifically conditioned. Notably, when predicting the next word given the previous ones, the language model is auto-regressive, which means that each predicted word is conditioned on the previous ones. The language model usually also decides when to stop generating caption words by outputting a special end-of-sequence token.

The main language modeling strategies applied to image captioning can be categorized as: 1. *LSTM-based* approaches, which can be either single-layer or two-layer; 2. *CNN-based* methods that constitute a first attempt in surpassing the fully recurrent paradigm; 3. *Transformer-based* fully-attentive approaches; 4. *image-text early-fusion* (BERT-like) strategies that directly connect the visual and textual inputs. This taxonomy is visually summarized in Fig. 2.1.

LSTM-based Models

As language has a sequential structure, RNNs are naturally suited to deal with the generation of sentences. Among RNN variants, LSTM [127] has been the predominant option for language modeling.

Single-layer LSTM

The most simple LSTM-based captioning architecture is based on a single-layer LSTM and was proposed by Vinyals *et al.* [311]. As shown in Fig. 2.5a, the visual encoding is used as the initial hidden state of the LSTM, which then generates the output caption. At each time step, a word is predicted by applying a softmax activation function over the projection of the hidden state into a vector of the same size as the vocabulary. During training, input words are taken from the ground-truth sentence, while during inference, input words are those generated at the previous step.

Shortly after, Xu *et al.* [341] introduced the additive attention mechanism. As depicted in Fig. 2.5b, in this case, the previous hidden state guides the attention mechanism over the visual features $\boldsymbol{X}$, computing a context vector which is then fed to the MLP in charge of predicting the output word.

Other approaches. Many subsequent works have adopted a decoder based on a single-layer LSTM, mostly without any architectural changes [353, 43, 241], while others have proposed significant modifications, summarized below.

Visual sentinel – Lu *et al.* [214] augmented the spatial image features with an additional learnable vector, called visual sentinel, which can be attended by the decoder in place of visual features while generating “non-visual” words (*e.g.* “the”, “of”, and “on”), for which visual features are not needed (Fig. 2.5c). At each time step, the visual sentinel is computed from the previous hidden state and generated word. Then, the model generates a context vector as a combination of attended image features and visual sentinel, whose importance is weighted by a learnable gate.

Hidden state reconstruction – Chen *et al.* [48] proposed to regularize the transition dynamics of the language model by using a second LSTM for reconstructing the previous hidden state based on the current one. Ge *et al.* [102] enhance context modeling by using a bidirectional LSTM with an auxiliary module. The auxiliary module in a direction approximates the hidden state of the LSTM in the other direction. Finally, a cross-modal attention mechanism combines grid visual features with the two sentences from the bidirectional LSTM to obtain the final caption.

Multi-stage generation – Wang *et al.* [326] proposed to generate a caption from coarse central aspects to finer attributes by decomposing the caption generation process into two phases: skeleton sentence generation and attributes enriching, both implemented with single-layer LSTMs. On the same line, Gu *et al.* [106] devised a coarse-to-fine multi-stage framework using a sequence of LSTM decoders, each operating on the output of the previous one to produce increasingly refined captions.

Semantic-guided LSTM – Jia *et al.* [150] proposed an extension of LSTM that includes semantic information to guide the generation (*e.g.* sentences from a cross-modal retrieval model, vectors from a multi-modal embedding, the image itself). Specifically, the semantic information is used as an extra input to each gate in the LSTM block.

Two-layer LSTM

LSTMs can be expanded to multi-layer structures to augment their capability of capturing higher-order relations. Donahue *et al.* [79] firstly proposed a two-layer LSTM as a language model for captioning, stacking two layers, where the hidden states of the first are the input to the second.

Two-layers and additive attention. Anderson *et al.* [7] went further and proposed to specialize the two layers to perform visual attention and the actual language modeling. As shown in Fig. 2.5d, the first LSTM layer acts as a top-down visual attention model which takes the previously generated word, the previous hidden state, and the mean-pooled image features. Then, the current hidden state is used to compute a probability distribution over image regions with an additive attention mechanism. The so-obtained attended image feature vector is fed to the second LSTM layer, which combines it with the hidden state of the first layer to generate a probability distribution over the vocabulary.

Variants of two-layers LSTM. Because of their representation power, LSTMs with two-layers and internal attention mechanisms represent the most employed language model approach before the advent of Transformer-based architectures [357, 347, 358, 277]. As such, many other variants have been proposed to improve the performance of this approach.

Neural Baby Talk – To ground words into image regions, Lu *et al.* [215] incorporated a pointing network that modulates the content-based attention mechanism. In particular, during the generation process, the network predicts slots in the caption, which are then filled with the image region classes. For non-visual words, a visual sentinel is used as dummy grounding. This approach leverages the object detector both as a feature region extractor and as a visual word prompter for the language model.

Reflective attention – Ke *et al.* [162] introduced two reflective modules: while the first computes the relevance between hidden states from all the past predicted words and the current one, the second improves the syntactic structure of the sentence by guiding the generation process with words common position information.

Look back and predict forward – On a similar line, Qin *et al.* [248] used two modules: the look back module that takes into account the previous attended vector to compute the next one, and the predict forward module that predicts the new two words at once, thus alleviating the accumulated errors problem that may occur at inference time.

Adaptive attention time – Huang *et al.* [138] proposed an adaptive attention time mechanism, in which the decoder can take an arbitrary number of attention steps for each generated word, determined by a confidence network on top of the second-layer LSTM.

Boosting LSTM with Self-Attention

Some works adopted the self-attention operator in place of the additive attention one in LSTM-based language models [137, 237, 203, 385]. In particular, Huang *et al.* [137] augmented the LSTM with the Attention on Attention operator, which computes another step of attention on top of visual self-attention. Pan *et al.* [237] introduced the X-Linear attention block, which enhances self-attention with second-order interactions and improves both the visual encoding and the language model. On a different line, Zhu *et al.* [385] applied the neural architecture search paradigm to select the connections between layers and the operations within gates of RNN-based image captioning language models, using a decoder enriched with self-attention [237].

Convolutional Language Models

A worth-to-mention approach is that proposed by Aneya *et al.* [11], which uses convolutions as a language model. In particular, a global image feature vector is combined with word embeddings and fed to a CNN, operating on all words in parallel during training and sequentially in inference. Convolutions are right-masked to prevent the model from using the information of future word tokens. Despite the clear advantage of parallel training, the usage of the convolutional

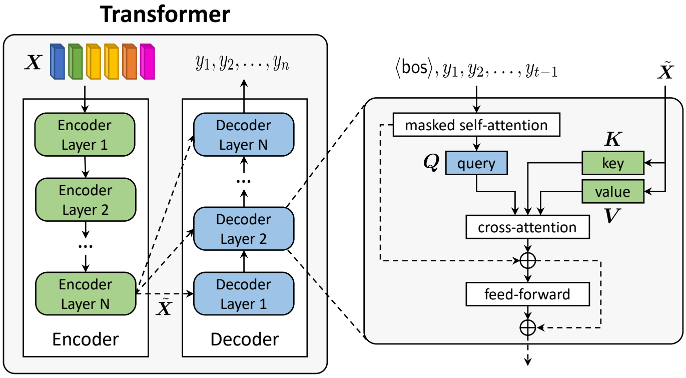


Figure 2.6: Schema of the Transformer-based language model. The caption generation is performed via masked self-attention over previously generated tokens and cross-attention with encoded visual features.

operator in language models has not gained popularity due to the poor performance and the advent of Transformer architectures.

Transformer-based Architectures

The fully-attentive paradigm proposed by Vaswani *et al.* [306] has completely changed the perspective of language generation. Shortly after, the Transformer model became the building block of other breakthroughs in NLP, such as BERT [78] and GPT [251], and the standard de-facto architecture for many language understanding tasks. As image captioning can be cast as a sequence-to-sequence problem, the Transformer architecture has been employed also for this task. The standard Transformer decoder performs a masked self-attention operation, which is applied to words, followed by a cross-attention operation, where words act as queries and the outputs of the last encoder layer act as keys and values, plus a final feed-forward network (Fig. 2.6). During training, a masking mechanism is applied to the previous words to constrain a unidirectional generation process. The original Transformer decoder has been employed in some image captioning models without significant architectural modifications [124, 115, 217, 328]. Besides, some variants have been proposed to improve language generation and visual feature encoding.

Gating mechanisms. Li *et al.* [190] proposed a gating mechanism for the cross-attention operator, which controls the flow of visual and semantic information by combining and modulating image regions representations with semantic attributes coming from an external tagger. On the same line, Ji *et al.* [148] integrated a context gating mechanism to modulate the influence of the global image representation on each generated word, modeled via multi-head attention. Cornia *et al.* [65] proposed to take into account all encoding layers in place of performing cross-attention only on the last one. To this end, they devised the meshed decoder, which contains a mesh operator that modulates the contribution of all the encoding layers independently and a gate that weights these contributions guided by the text query. More details about this method are presented in the following section of this thesis. In [328, 60], the decoder architecture is again employed in conjunction with textual prefixes, also extracted from pre-trained visual-semantic models and employed as visual tags.

BERT-like Architectures

Despite the encoder-decoder paradigm being a common approach to image captioning, some works have revisited captioning architectures to exploit a BERT-like [78] structure in which the visual and textual modalities are fused together in the early stages (Fig. 2.7). The main advantage of this architecture is that layers dealing with text can be initialized with pre-trained parameters learned from massive textual corpora. Therefore, the BERT paradigm has been widely adopted in works that exploit pre-training [196, 381, 371].

The first example is due to Zhou *et al.* [381], who developed a unified model that fuses visual and textual modalities into a BERT-like architecture for image captioning. The model consists of a shared multi-layer Transformer encoder network for both encoding and decoding, pre-trained on a large corpus of image-caption pairs and then fine-tuned for image captioning by right-masking the tokens sequence to simulate the unidirectional generation process. Further, Li *et al.* [196] introduced the usage of object tags detected in the image as anchors points for learning a better alignment in vision-and-language joint representations. To this end, their model represents an input image-text pair as a word tokens-object tags-region features triple, where the object tags are the textual classes proposed by the object detector.

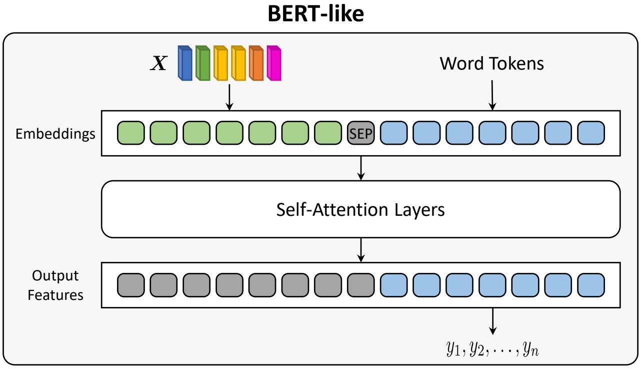


Figure 2.7: Schema of a BERT-like language model. A single stream of attentive layers processes both image regions and word tokens and generates the output caption.

Non-autoregressive Language Models

Thanks to the parallelism offered by Transformers, non-autoregressive language models have been proposed in machine translation to reduce the inference time by generating all words in parallel. Some efforts have been made to apply this paradigm to image captioning [88, 113, 89, 114]. The first approaches towards a non-autoregressive generation were composed of a number of different generation stages, where all words were predicted in parallel and refined at each stage. Subsequent methods, instead, employ reinforcement learning techniques to improve the final results. Specifically, these approaches treat the generation process as a cooperative multi-agent reinforcement system, where the positions in of the words in the target sequence are viewed as agents that learn to cooperatively maximize a sentence-level reward [113, 114]. These works also leverage knowledge distillation on unlabeled data and a post-processing step to remove identical consecutive tokens.

Discussion

Recurrent models have been the standard for many years, and their application brought to the development of clever and successful ideas that can be integrated also into non-recurrent solutions. However, they are slow to train and struggle to maintain long-term dependencies: these drawbacks are alleviated by autoregressive and Transformer-based solutions that recently gained popularity. Inspired by the success of pre-training on large, unsupervised corpora for NLP tasks, massive pre-training has been applied also for image captioning by employing either encoder-decoder or BERT-like architectures, often in conjunction with textual tags. This strategy led to impressive performance, suggesting that visual and textual semantic relations can be inferred and learned also from not well-curated data [196, 328, 134]. BERT-like architectures are suitable for such a massive pre-training but are not generative architectures by design. Massive pre-training on generative-oriented architectures [328, 60] is currently a worth-exploring direction, which leads to performances that are at least on-pair with the early-fusion counterparts.

2.1.4 Training strategies

An image captioning model is commonly expected to generate a caption word by word by taking into account the previous words and the image. At each step, the output word is sampled from a learned distribution over the vocabulary words. In the most simple scenario, *i.e.* the greedy decoding mechanism, the word with the highest probability is output. The main drawback of this setting is that possible prediction errors quickly accumulate along the way. To alleviate this drawback, one effective strategy is to use the beam search algorithm [173] that, instead of outputting the word with maximum probability at each time step, maintains k sequence candidates (those with the highest probability at each step) and finally outputs the most probable one.

During training, the captioning model must learn to properly predict the probabilities of the words to appear in the caption. To this end, the most common training strategies are based on 1. *cross-entropy loss*; 2. *masked language model*; 3. *reinforcement learning* that allows directly optimizing for captioning-specific non-differentiable metrics; 4. *vision-and-language pre-training* objectives (see Fig. 2.1).

Cross-Entropy Loss

The cross-entropy loss is the first proposed and most used objective for image captioning models. With this loss, the goal of the training, at each timestep, is to minimize the negative log-likelihood of the current word given the previous ground-truth words. Given a sequence of target words $y_{1:T}$, the loss is formally defined as:

$$L_{XE}(\theta) = -\sum_{i=1}^{n} \log\left(P\left(y_i \mid y_{1:i-1}, \boldsymbol{X}\right)\right), \tag{2.4}$$

where P is the probability distribution induced by the language model, y_i the ground-truth word at time i, $y_{1:i-1}$ indicate the previous ground-truth words, and $\boldsymbol{X}$ the visual encoding. The cross-entropy loss is designed to operate at word level and optimize the probability of each word in the ground-truth sequence without considering longer range dependencies between generated words. The traditional training setting with cross-entropy also suffers from the exposure bias problem [255] caused by the discrepancy between the training data distribution as opposed to the distribution of its own predicted words.

Masked Language Model (MLM)

The first masked language model has been proposed for training the BERT [78] architecture. The main idea behind this optimization function consists in randomly masking out a small subset of the input tokens sequence and training the model to predict masked tokens while relying on the rest of the sequence, *i.e.* both previous and subsequent tokens. As a consequence, the model learns to employ contextual information to infer missing tokens, which allows building a robust sentence representation where the context plays an essential role. Since this strategy considers only the prediction of the masked tokens and ignores the prediction of the non-masked ones, training with it is much slower than training for complete left-to-right or right-to-left generation. Notably, some works have employed this strategy as a pre-training objective, sometimes completely avoiding the combination with the cross-entropy [196, 371].

Reinforcement Learning

Given the limitations of word-level training strategies observed when using limited amounts of data, a significant improvement was achieved by applying the reinforcement learning paradigm for training image captioning models. Within this framework, the image captioning model is considered as an agent whose parameters determine a policy. At each time step, the agent executes the policy to choose an action, *i.e.* the prediction of the next word in the generated sentence. Once the end-of-sequence is reached, the agent receives a reward, and the aim of the training is to optimize the agent parameters to maximize the expected reward.

Many works harnessed this paradigm and explored different sequence-level metrics as rewards. The first proposal is due to Ranzato *et al.* [255], which introduced the usage of the REINFORCE algorithm [331] adopting BLEU [238] and ROUGE [199] as reward signals. Ren *et al.* [261] experimented using visual-semantic embeddings obtained from a network that encodes the image and the so far generated caption in order to compute a similarity score to be used as reward. Liu *et al.* [207] proposed to use as reward a linear combination of SPICE [5] and CIDEr [307], called SPIDEr. Finally, the most widely adopted strategy [369, 97, 65], introduced by Rennie *et al.* [262], entails using the CIDEr score, as it correlates better with human judgment [307]. The reward is normalized with respect to a baseline value to reduce variance. Formally, to compute the loss gradient, beam search and greedy decoding are leveraged as follows:

$$\nabla_\theta L(\theta) = -\frac{1}{k}\sum_{i=1}^{k}\left((r(\boldsymbol{w}^i) - b)\nabla_\theta \log P(\boldsymbol{w}^i)\right), \tag{2.5}$$

where $\boldsymbol{w}^i$ is the i-th sentence in the beam or a sampled collection, $r(\cdot)$ is the reward function, *i.e.* the CIDEr computation, and b is the baseline, computed as the reward of the sentence obtained via greedy decoding [262], or as the average reward of the beam candidates [65].

Note that, since it would be difficult for a random policy to improve in an acceptable amount of time, the usual procedure entails pre-training with cross-entropy or masked language model first, and then fine-tuning stage with reinforcement learning by employing a sequence level metric as reward. This ensures the initial reinforcement learning policy to be more suitable than the random one.

Large-scale Pre-Training

In the context of vision-and-language pre-training in early-fusion architectures, one of the most common pre-training objectives is the masked contextual token loss, where tokens of each modality (visual and textual) are randomly masked following the BERT strategy [78], and the model has to predict the masked input based on the context of both modalities, thus connecting their joint representation. Another largely adopted strategy entails using a contrastive loss, where the inputs are organized as image regions-captions words-object tags triples, and the model is asked to discriminate correct triples from polluted ones, in which tags are randomly replaced [196, 371]. Other objectives take into account the text-image alignment at a word-region level and entail predicting the original word sequence given a corrupted one [337].

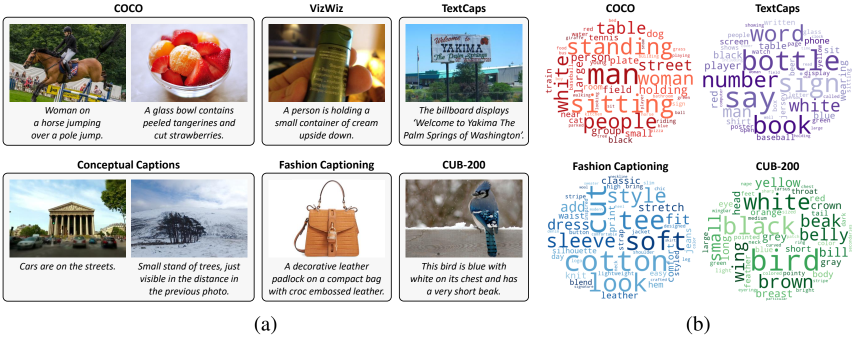

COCO
Woman on a horse jumping over a pole jump.
A glass bowl contains peeled tangerines and cut strawberries.
VizWiz
A person is holding a small container of cream upside down.
TextCaps
The billboard displays 'Welcome to Yakima The Palm Springs of Washington'.
Conceptual Captions
Cars are on the streets.
Small stand of trees, just visible in the distance in the previous photo.
Fashion Captioning
A decorative leather padlock on a compact bag with croc embossed leather.
CUB-200
This bird is blue with white on its chest and has a very short beak.
(a)
COCO
table standing man street woman sitting people white
TextCaps
word bottle number sign say white book
Fashion Captioning
classic style tee fit sleeve soft cotton look
CUB-200
yellow white black beak belly bird brown
(b)

Figure 2.8: Qualitative examples from some of the most common image captioning datasets: **(a)** image-caption pairs; **(b)** word clouds of the captions most common visual words.

On the other hand, cross-entropy has also been used when pre-training on noisy captions [60, 328], sometimes also employing prefixes. PrefixLM [328] has indeed proved to be a valuable strategy that enables bidirectional attention within the prefix sequence, and thus, it is applicable for both decoder-only and encoder-decoder sequence-to-sequence language models. Noticeably, some large-scale models pre-trained on noisy data under this setting can achieve state-of-the-art performance without requiring a fine-tuning stage with Reinforcement [328].

Finally, we notice that image captioning can be used as a pre-training task to efficiently learn visual representations, which can benefit downstream tasks such as image classification, object detection, and instance segmentation [76].

2.1.5 Evaluation protocol

As for any data-driven task, the development of image captioning has been enabled by the collection of large datasets and the definition of quantitative scores to evaluate the performance and monitor the advancement of the field.

Datasets

Image captioning datasets contain images and one or multiple captions associated with them. Having multiple ground-truth captions for each image helps to capture the variability of human descriptions. Other than the number of available captions, also their characteristics (*e.g.* average caption length and vocabulary size) highly influence the design and the performance of image captioning algorithms. Note that the distribution of the terms in the datasets captions is usually long-tailed, thus, when using word-level dictionaries, the common practice is to include in the vocabulary only those terms whose frequency is above a pre-defined threshold. Recently, however, using subword-based tokenization approaches like BPE [267] is a popular choice that allows avoiding dataset pre-processing. The available datasets differ both on the images contained (for their domain and visual quality) and on the captions associated with the images (for their length, number, relevance, and style). A summary of the most used public datasets is reported in Table 3.6, and some sample image-caption pairs are reported in Fig. 2.8, along with some word clouds obtained from the 50 most used visual words in the captions.

Standard captioning datasets

Standard benchmark datasets are used by the community to compare their approaches on a common test-bed, a procedure that guides the development of image captioning strategies by allowing to identify suitable directions. Datasets used as benchmarks should be representative of the task at hand, both in terms of the challenges and ideal expected results (*i.e.* achievable human performance). Further, they should contain a large number of generic-domain images, each associated with multiple captions.

Early image captioning architectures [222, 79, 159] were commonly trained and tested on the **Flickr30K** [364] and **Flickr8K** [128] datasets, consisting of pictures collected from the Flickr website, containing everyday activities, events, and scenes, paired with five captions each. Currently, the most commonly used dataset is **Microsoft COCO** [201], which consists of images of complex scenes with people, animals, and common everyday objects in their context. It contains more than 120,000 images, each annotated with five captions, divided into 82,783 images for training and 40,504 for validation. For ease of evaluation, most of the literature follows the splits defined by Karpathy *et al.* [159], where 5,000 images of the original validation set are used for validation, 5,000 for test, and the rest for training. The dataset has also an official test set, composed of 40,775 images paired with 40 private captions each, and a public evaluation server[2].

[2] `https://competitions.codalab.org/competitions/3221`

Table 2.1: Overview of the main image captioning datasets.

	Domain	**Nb. Images**	**Nb. Caps (per Image)**	**Vocab Size**	**Nb. Words (per Cap.)**
COCO [201]	Generic	132K	5	27K (10K)	10.5
Flickr30K [364]	Generic	31K	5	18K (7K)	12.4
Flickr8K [128]	Generic	8K	5	8K (3K)	10.9
CC3M [272]	Generic	3.3M	1	48K (25K)	10.3
CC12M [37]	Generic	12.4M	1	523K (163K)	20.0
SBU Captions [235]	Generic	1M	1	238K (46K)	12.1
VizWiz [117]	Assistive	70K	5	20K (8K)	13.0
CUB-200 [256]	Birds	12K	10	6K (2K)	15.2
Oxford-102 [256]	Flowers	8K	10	5K (2K)	14.1
Fashion Cap. [350]	Fashion	130K	1	17K (16K)	21.0
BreakingNews [254]	News	115K	1	85K (10K)	28.1
GoodNews [27]	News	466K	1	192K (54K)	18.2
TextCaps [279]	OCR	28K	5/6	44K (13K)	12.4
Loc. Narratives [246]	Generic	849K	1/5	16K (7K)	41.8

Pre-training datasets

Although training on large well-curated datasets is a sound approach, some works [212, 196, 328, 134] have demonstrated the benefits of pre-training on even bigger vision-and-language datasets, which can be either image captioning datasets of lower-quality captions or datasets collected for other tasks (*e.g.* visual question answering [196, 381], text-to-image generation [253], image-caption association [250]). Among the datasets used for pre-training, that have been specifically collected for image captioning, it is worth mentioning **SBU Captions** [235], originally used for tackling image captioning as a retrieval task [128], which contains around 1 million image-text pairs, collected from the Flickr website. Similarly, **YFCC100M** [296] is composed of 100 million media objects in which 14.8 million images are available with automatically-collected textual descriptions. Later, the **Conceptual Captions** [272, 37] datasets have been proposed, which are collections of around 3.3 million (CC3M) and 12 million (CC12M) images paired with one weakly-associated description automatically collected from the web with a relaxed filtering procedure. Differently from previous datasets, **Wikipedia-based Image Text** (WIT) [283] provides images coming from Wikipedia together with various metadata extracted from the original pages, with approximately 5.3 million images available with the corresponding descriptions in English. Although the large scale and variety in caption style make all these datasets particularly interesting for pre-training, the contained captions can be noisy, and the availability of images is not always guaranteed since most of them are provided as URLs.

Pre-training on such datasets requires significant computational resources and effort to collect the data needed. Nevertheless, this strategy represents an asset to obtain state-of-the-art performances. Accordingly, some pre-training datasets are currently not publicly available, such as **ALIGN** [149, 328] and **ALT-200** [134], respectively containing 1.8 billion and 200 million noisy image-text pairs, or the datasets used to train DALL-E [253] and CLIP [250] consisting of 250 and 400 million pairs.

Domain-specific datasets

While domain-generic benchmark datasets are important to capture the main aspects of the image captioning task, domain-specific datasets are also important to highlight and target specific challenges. These may relate to the visual domain (*e.g.* type and style of the images) and the semantic domain. In particular, the distribution of the terms used to describe domain-specific images can be significantly different from that of the terms used for domain-generic images.

An example of dataset-specific in terms of the visual domain is the **VizWiz Captions** [117] dataset, collected to favor the image captioning research towards assistive technologies. The images in this dataset have been taken by visually-impaired people with their phones, thus, they can be of low quality and concern a wide variety of everyday activities, most of which entail reading some text.

Some examples of specific semantic domain are the **CUB-200** [329] and the **Oxford-102** [233] datasets, which contain images of birds and flowers, respectively, that have been paired with ten captions each by Reed *et al.* [256]. Given the specificity of these datasets, rather than for standard image captioning, they are usually adopted for different related tasks such as cross-domain captioning [46], visual explanation generation [121, 122], and text-to-image synthesis [257]. Another domain-specific dataset is **Fashion Captioning** [350] that contains images of clothing items in different poses and colors that may share the same caption. The vocabulary for describing these images is somewhat smaller and more specific than for generic datasets. Differently, datasets as **BreakingNews** [254] and **GoodNews** [27] enforce using a richer vocabulary since their images, taken from news articles, have long associated captions written by expert journalists. The same applies to the **TextCaps** [279] dataset, which contains images with text, that must be “read” and included in the caption, and to **Localized Narratives** [246], whose captions have been collected by recording people freely narrating what they see in the images. Collecting domain-specific datasets and developing solutions to tackle the challenges they pose is crucial to extend the applicability of image captioning algorithms.

Evaluation Metrics

Evaluating the quality of a generated caption is a tricky and subjective task [307, 5], complicated by the fact that captions cannot only be grammatical and fluent but need to properly refer to the input image. Arguably, the best way to measure the quality of the caption for an image is still carefully designing a human evaluation campaign in which multiple users score the produced sentences [161]. However, human evaluation is costly and not reproducible – which prevents a fair comparison between different approaches. Automatic scoring methods exist that are used to assess the quality of system-produced captions, usually by comparing them with human-produced reference sentences, although some metrics do not rely on reference captions.

Standard evaluation metrics

The first strategy adopted to evaluate image captioning performance consists of exploiting metrics designed for NLP tasks. For example, the **BLEU** score [238] and the **METEOR** [19] score were introduced for machine translation. The former is based on *n*-gram precision considering *n*-grams up to length four; the latter favors the recall of matching unigrams from the candidate and reference sentences in their exact form, stemmed form, and meaning. Moreover, the **ROUGE** score [199] was designed for summarization and applied also for image captioning in its variant considering the longest subsequence of tokens in the same relative order, possibly with other tokens in-between, that appears in both candidate and reference caption. Later, specific image captioning metrics have been proposed [307, 5]. The reference **CIDEr** score [307] is based on the cosine similarity between the Term Frequency-Inverse Document Frequency weighted *n*-grams in the candidate caption and in the set of reference captions associated with the image, thus taking into account both precision and recall. The **SPICE** score [5] considers matching tuples extracted from the candidate and the reference (or possibly directly the image) scene graphs, thus favoring the semantic content rather than the fluency.

Table 2.2: Performance analysis of representative image captioning approaches in terms of different evaluation metrics. The † marker indicates models trained by us with ResNet-152 features, while the ‡ marker indicates unofficial implementations. The dataset considered for this analysis is COCO. For all the metrics, the higher the value, the better (↑).

		Standard Metrics						Diversity Metrics				Embedding-based Metrics			Learning-based Metrics			
	#Params (M)	**B-1**	**B-4**	**M**	**R**	**C**	**S**	**Div-1**	**Div-2**	**Vocab**	**%Novel**	**WMD**	**Alignment**	**Coverage**	**TIGEr**	**BERT-S**	**CLIP-S**	**CLIP-S$^{\text{Ref}}$**
Show and Tell† [311]	13.6	72.4	31.4	25.0	53.1	97.2	18.1	0.014	0.045	635	36.1	16.5	0.199	71.7	71.8	93.4	0.697	0.762
SCST (FC)‡ [262]	13.4	74.7	31.7	25.2	54.0	104.5	18.4	0.008	0.023	376	60.7	16.8	0.218	74.7	71.9	89.0	0.691	0.758
Show, Attend and Tell† [341]	18.1	74.1	33.4	26.2	54.6	104.6	19.3	0.017	0.060	771	47.0	17.6	0.209	72.1	73.2	93.6	0.710	0.773
SCST (Att2in)‡ [262]	14.5	78.0	35.3	27.1	56.7	117.4	20.5	0.010	0.031	445	64.9	18.5	0.238	76.0	73.9	88.9	0.712	0.779
Up-Down‡ [7]	52.1	79.4	36.7	27.9	57.6	122.7	21.5	0.012	0.044	577	67.6	19.1	0.248	76.7	74.6	88.8	0.723	0.787
SGAE [347]	125.7	81.0	39.0	28.4	58.9	129.1	22.2	0.014	0.054	647	71.4	20.0	0.255	76.9	74.6	94.1	0.734	0.796
MT [277]	63.2	80.8	38.9	28.8	58.7	129.6	22.3	0.011	0.048	530	70.4	20.2	0.253	77.0	74.8	88.8	0.726	0.791
AoANet [137]	87.4	80.2	38.9	29.2	58.8	129.8	22.4	0.016	0.062	740	69.3	20.0	0.254	77.3	75.1	94.3	0.737	0.797
X-LAN [237]	75.2	80.8	39.5	29.5	59.2	132.0	23.4	0.018	0.078	858	73.9	20.6	0.261	77.9	75.4	94.3	0.746	0.803
DPA [203]	111.8	80.3	40.5	29.6	59.2	133.4	23.3	0.019	0.079	937	65.9	20.5	0.261	77.3	75.0	94.3	0.738	0.802
AutoCaption [385]	-	81.5	40.2	29.9	59.5	135.8	23.8	0.022	0.096	1064	75.8	20.9	0.262	77.7	75.4	94.3	0.752	0.808
ORT [124]	54.9	80.5	38.6	28.7	58.4	128.3	22.6	0.021	0.072	1002	73.8	19.8	0.255	76.9	75.1	94.1	0.736	0.796
CPTR [208]	138.5	81.7	40.0	29.1	59.4	129.4	-	0.014	0.068	667	75.6	20.2	0.261	77.0	74.8	94.3	0.745	0.802
$\mathcal{M}^2$ Transformer [65]	38.4	80.8	39.1	29.2	58.6	131.2	22.6	0.017	0.079	847	**78.9**	20.3	0.256	76.0	75.3	93.7	0.734	0.792
X-Transformer [237]	137.5	80.9	39.7	29.5	59.1	132.8	23.4	0.018	0.081	878	74.3	20.6	0.257	77.7	75.5	94.3	0.747	0.803
Unified VLP [381]	138.2	80.9	39.5	29.3	59.6	129.3	23.2	0.019	0.081	898	74.1	**26.6**	0.258	77.1	75.1	**94.4**	0.750	0.807
VinVL [371]	369.6	**82.0**	**41.0**	**31.1**	**60.9**	**140.9**	**25.2**	**0.023**	**0.099**	**1125**	77.9	20.5	**0.265**	**79.6**	**75.7**	88.5	**0.766**	**0.820**

As expected, metrics designed for image captioning usually correlate better with human judgment than those borrowed from other NLP tasks (with the exception of METEOR [19]), both at corpus-level and caption-level [5, 270, 68]. Correlation with human judgment is measured via statistical correlation coefficients (such as Pearson's, Kendall's, and Spearman's correlation coefficients) and via the agreement with humans' preferred caption in a pair of candidates, all evaluated on sample captioned images.

Diversity metrics

To better assess the performance of a captioning system, it is common practice to consider a set of the above-mentioned standard metrics. Nevertheless, these are somehow gameable because they favor word similarity rather than meaning correctness [32]. Another drawback of the standard metrics is that they do not capture (but rather disfavor) the desirable capability of the system to produce novel and diverse captions, which is more in line with the variability with which humans describe complex images. This consideration brought to the development of diversity metrics [275, 305, 321, 322]. Most of these metrics can potentially be calculated even when no ground-truth captions are available at test time. However, since they overlook the syntactic correctness of the captions and their relatedness with the image, it is advisable to combine them with other metrics.

The overall performance of a captioning system can be evaluated in terms of corpus-level diversity or, when the system can output multiple captions for the same image, single image-level diversity (termed as *global diversity* and *local diversity*, respectively, in [305]). To quantify the former, it can be considered the number of unique words used in all the generated captions (**Vocab**) and the percentage of generated captions that were not present in the training set (**%Novel**). For the latter, it can be used the ratio of unique captions unigrams or bigrams to the total number of captions unigrams (**Div-1** and **Div-2**).

Embedding-based metrics

Another approach to captioning evaluation consists in relying on captions semantic similarity or other specific aspects of caption quality, which are estimated via embedding-based metrics [263, 152, 327]. For example, the **WMD** score [179], originally introduced to evaluate document semantic dissimilarity, can also be applied to captioning evaluation by considering generated captions and ground-truth captions as the compared documents [165]. Moreover, the **Alignment**

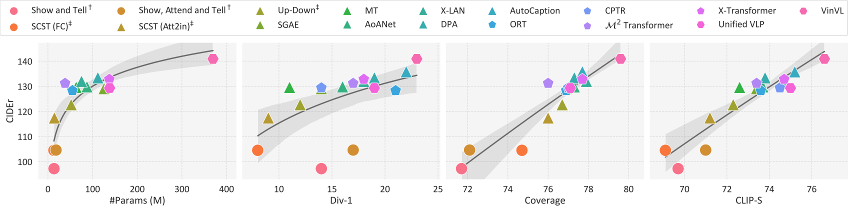


Figure 2.9: Relationship between CIDEr, number of parameters and other scores. Values of Div-1 and CLIP-S are multiplied by powers of 10 for readability.

score [57] is based on the alignment between the sequences of nouns in the candidate and reference sentence and captures whether concepts are mentioned in a human-like order. Finally, the **Coverage** score [58, 25] expresses the completeness of a caption, which is evaluated by considering the mentioned scene visual entities. Since this score considers visual objects directly, it can be applied even when no ground-truth caption is available.

Learning-based evaluation

As a further development towards captions quality assessment, learning-based evaluation strategies [270, 68, 185, 360, 323, 184] are being investigated. To this end, it can be exploited a component of a complete captioning approach, in charge to evaluate the produced caption completeness [69] or how human-like it is [70]. Alternatively, learning-based evaluation is usually based on a pre-trained model. For example, the **BERT-S** score [372], which is used to evaluate various language generation tasks [304], exploits pre-trained BERT embeddings [78] to represent and match the tokens in the reference and candidate sentences via cosine similarity. Moreover, the **TIGEr** score [153] represents the reference and candidate captions as grounding score vectors obtained from a pre-trained model [186] that grounds their words on the image regions and scores the candidate caption based on the similarity of the grounding vectors. Further, the **CLIP-S** score [125] is a direct application of the CLIP [250] model to image captioning evaluation and consists of an adjusted cosine similarity between image and candidate caption representation. Thus, CLIP-S is designed to work without reference captions, although the CLIP-S$^{\text{Ref}}$ variant can exploit also the reference captions.

2.1.6 Experimental evaluation

In Table 2.2, we analyze the performance of some of the main approaches in terms of all the evaluation scores presented in Section 2.1.5 to take into account the different aspects of caption quality these express and report their number of parameters to give an idea of the computational complexity and memory occupancy of the models. The data in the table have been obtained either from the model weights and captions files provided by the original authors or from our best implementation. Given its large use as a benchmark in the field, we consider the domain-generic COCO dataset also for this analysis. In the table, methods are clustered based on the information included in the visual encoding and ordered by CIDEr score. It can be observed that standard and embedding-based metrics all had a substantial improvement with the introduction of region-based visual encodings. Further improvement was due to the integration of information on inter-objects relations, either expressed via graphs or self-attention. Notably, CIDEr, SPICE, and Coverage most reflect the benefit of vision-and-language pre-training. Moreover, as expected, it emerges that the diversity-based scores are correlated, especially Div-1 and Div-2 and the Vocab Size. The correlation of this family of scores and the others is almost linear, except for early approaches, which perform averagely well in terms of Diversity despite lower values for standard metrics. From the trend of learning-based scores, it emerges that exploiting models trained on textual data only (BERT-S, reported in the table as its F1-score variant) does not help discriminating among image captioning approaches. On the other hand, considering as reference only the visual information and disregarding the ground-truth captions is possible with the appropriate vision-and-language pre-trained model (consider that CLIP-S and CLIP-S$^{\text{Ref}}$ are linearly correlated). This is a desirable property for an image captioning evaluation score since it allows estimating the performance of a model without relying on reference captions that can be limited in number and somehow subjective.

For readability, in Fig. 2.9 we highlight the relation between the CIDEr score and other characteristics from Table 2.2. We chose CIDEr as this score is commonly regarded as one of the most relevant indicators of image captioning systems performance. The first plot, depicting the relation between model complexity and performance, shows that more complex models do not necessarily bring to better performance. The other plots describe an almost-linear relation between CIDEr and the other scores, with some flattening for high CIDEr values. These trends confirm the suitability of the CIDEr score as an indicator of the overall performance of an image captioning algorithm, whose specific characteristics in terms of the produced captions would still be expressed more precisely in terms of non-standard metrics.

2.1.7 Image captioning variants

Beyond general-purpose image captioning, several specific sub-tasks have been explored in the literature. These can be classified into five categories according to their scope: 1. *dealing with the lack of training data*; 2. *focusing on the visual input*; 3. *focusing on the textual output*; 4. *application specific*; 5. *addressing user requirements*.

Dealing with the lack of training data

Paired image-caption datasets are very expensive to obtain. Thus, some image captioning variants are being explored that limit the need for full supervision information.

Novel Object Captioning. Novel object captioning focuses on describing objects not appearing in the training set, thus enabling a zero-shot learning setting that can increase the applicability of the models in the real world. Early approaches to this task [123, 309] tried to transfer knowledge from out-domain images by conditioning the model on external unpaired visual and textual data at training time. To explore this strategy, Hendricks *et al.* [123] introduced a variant of the COCO dataset [201], called *held-out COCO*, in which image-caption pairs containing one of eight pre-selected object classes were removed from the training set but not from the test set. To further encourage research on this task, the more challenging *nocaps* dataset, with nearly 400 novel objects, has been introduced [2]. Some approaches to this variant [356, 197] integrate copying mechanisms in the language model to select novel objects predicted from a tagger or generate a caption template with placeholders to be filled with novel objects [336, 215]. On a different line, Anderson *et al.* [6] devised the Constrained Beam Search algorithm to force the inclusion of selected tag words in the output caption, following the predictions of a tagger. Moreover, following the pre-training trend with BERT-like architectures, Hu *et al.* [135] proposed a multi-layer Transformer model pre-trained by randomly masking one or more tags from image-tag pairs. Finally, in the next chapter we present a method for novel object captioning that learns to select the most relevant objects to describe and constrained the caption generation accordingly.

Unpaired Image Captioning. Unpaired Image Captioning approaches can be either unsupervised or semi-supervised. Unsupervised captioning aims at understanding and describing images without paired image-text training data. Following unpaired machine translation approaches, the early work [107] proposes to generate captions in a pivot language and then translate predicted captions to the target language. After this work, the most common approach focuses on adversarial learning by training an LSTM-based discriminator to distinguish whether a caption is real or generated [90, 182]. As alternative approaches, it is worth mentioning [108] that generates a caption from the image scene-graph and [110] that leverages a memory-based network. Moreover, semi-supervised approaches have been proposed, such as [167], which uses both paired and unpaired data with adversarial learning, and [23], which performs iterative self-learning.

Continual Captioning. Continual captioning aims to deal with partially unavailable data by following the continual learning paradigm to incrementally learn new tasks without forgetting what has been learned before. In this respect, new tasks can be represented as sequences of captioning tasks with different vocabularies, as proposed in [74], and the model should be able to transfer visual concepts from one to the other while enlarging its vocabulary.

Focusing on the visual input

Some sub-tasks focus on making the textual description more correlated with visual data.

Dense Captioning. Dense captioning was proposed by Johnson *et al.* [157] and consists of concurrently localizing and describing salient image regions with short natural language sentences. In this respect, the task can be conceived as a generalization of object detection, where caption replaces object tags, or image captioning, where single regions replace the full image. To address this task, contextual and global features [344, 194] and attribute generators [361, 166] can be exploited. Related to this variant, an important line of works [174, 198, 223, 38, 366, 216] focuses on the generation of textual paragraphs that densely describe the visual content as a coherent story.

Text-based Image Captioning. Text-based image captioning, also known as OCR-based image captioning or image captioning with reading comprehension, aims at reading and including the text appearing in images in the generated descriptions. The task was introduced by Sidorov *et al.* [279] with the TextCaps dataset. Another dataset designed for pre-training for this variant is *OCR-CC* [352], which is a subset of images containing meaningful text taken from the CC3M dataset [272] and automatically annotated through a commercial OCR system. The common approach to this variant entails combining image regions and text tokens, *i.e.* groups of characters from an OCR, possibly enriched with mutual spatial information [315, 316], in the visual encoding [279, 384]. Another direction entails generating multiple captions describing different parts of the image, including the contained text [340].

Change Captioning. Change captioning targets changes that occurred in a scene, thus requiring both accurate change detection and effective natural language description. The task was first presented in [147] with the *Spot-the-Diff* dataset, composed of pairs of frames extracted from video surveillance footages and the corresponding textual descriptions of visual changes. To further explore this variant, the *CLEVR-Change* dataset [240] has been introduced, which contains five scene change types on almost 80K image pairs. The proposed approaches for this variant apply attention mechanisms to focus on semantically relevant aspects without being deceived by distractors such as viewpoint changes [276, 139, 168] or perform multi-task learning with image retrieval as an auxiliary task [132], where an image must be retrieved from its paired image and the description of the occurred changes.

Focusing on the textual output

Since every image captures a wide variety of entities with complex interactions, human descriptions tend to be diverse and grounded to different objects and details. Some image captioning variants explicitly focus on these aspects.

Diverse Captioning. Diverse image captioning tries to replicate the quality and variability of the sentences produced by humans. The most common technique to achieve diversity is based on variants of the beam search algorithm [310] that entail dividing the beams into similar groups and encouraging diversity between groups. Other solutions have been investigated, such as contrastive learning [71], conditional GANs [70, 275], and paraphrasing [206]. However, these solutions tend to underperform in terms of caption quality, which is partially recovered by using variational auto-encoders [320, 10, 39, 219]. Another approach is exploiting multiple part-of-speech tags sequences predicted from image region classes [77] and forcing the model to produce different captions based on these sequences.

Multilingual Captioning. Since image captioning is commonly performed in English, multilingual captioning [81] aims to extend the applicability of captioning systems to other languages. The two main strategies entail collecting captions in different languages for commonly used datasets (*e.g.* Chinese and Japanese captions for COCO images [195, 231], German captions for Flick30K [82]), or directly training multilingual captioning systems with unpaired captions [81, 183, 107, 282].

Application-specific Captioning

Image captioning can be applied to ease and automate activities involving text generation from images. For example, captioning systems can be applied for medical report generation, for which they need to predict disease tags and try to imitate the style of real medical reports [155, 204, 346]. Another interesting application is art description generation, which entails describing not only factual aspects of the artworks, but also their context and style, and conveyed message art description [17]. To this end, captioning systems could also rely on external knowledge, *e.g.* metadata. A similar application is automatic caption generation for news articles [254, 27], for which named entities from the article should be described [91, 302], and the rich journalistic style should be maintained [205, 349]. Another important application domain is assistive technology for the visually impaired [334], where image captioning approaches must be able to provide informative descriptions even for low-quality visual inputs [117].

Addressing user requirements

Regular image captioning models generate factual captions with a neutral tone and no interaction with end-users. Instead, some image captioning sub-tasks are devoted to coping with user requests.

Personalized Captioning. Humans consider more effective the captions that avoid stating the obvious and that are written in a style that catches their interest. Personalized image captioning aims at fulfilling this requirement by generating descriptions that take into account the user's prior knowledge, active vocabulary, and writing style. To this end, early approaches exploit a memory block as a repository for this contextual information [54, 239]. On another line, Zhang *et al.* [374] proposed a multi-modal Transformer network that personalizes captions conditioned on the user's recent captions and a learned user representation. Other works have instead focused on the style of captions as an additional controllable input and proposed to solve this task by exploiting unpaired stylized textual corpus [95, 225, 112, 377]. Some datasets have been collected to explore this variant, such as *InstaPIC* [54], which is composed of multiple Instagram posts from the same users, *FlickrStyle10K* [95], which contains images and textual sentences with two different styles, and *Personality-Captions* [278], which contains triples of images, captions, and one among 215 personality traits.

Controllable Captioning. Controllable captioning puts the users in the loop by asking them to select and give priorities to what should be described in an image. This information is exploited as a guiding signal for the generation process. The signal can be sparse, as selected image regions [378, 57] and user-provided visual words [77], or dense, as mouse traces [246, 226]. Eventually, the guiding signal can incorporate some form of structure, such as sequences that encode the mentioning order of concepts (part-of-speech tag as in [77]) or visual objects [57]. Guiding inputs can also encode the relation between objects that is most of interest for the user, as done for example in [42] via verbs and semantic roles (verbs represent activities in the image and semantic roles determine how objects engage in these activities) and in [45, 379] via user-generated or user-selected scene graphs. A different control signal is introduced by [75], which consist of a length-level embedding added as an additional token to each textual word, providing existing models the ability to generate length-controllable image captions.

Image Captioning Editing. Image captioning editing was proposed by Sammani *et al.* [264], following the consideration that generated captions may have repetitions and inconsistencies. This variant focuses on decoupling the decoding stage in a caption generation step and a caption polishing one to correct syntactic errors.

2.2 Meshed-Memory Transformer

In the previous section, we have comprehensively reviewed the captioning task in the deep learning era. As shown, in the last few years attentive models have been improved by replacing this type of attention over a grid of features with attention over image regions coming from an object detector [7, 317, 376]. In these models, the generative process attends a set of regions which are softly selected while generating the caption.

Regarding the language model, the use of recurrent neural networks has remained the dominant approach, with the exception of the investigation of convolutional language models [11], which however did not become a leading choice. The recent advent of fully-attentive models, in which the recurrent relation is abandoned in favour of the use of self-attention, offers unique opportunities in terms of set and sequence modeling performances, as testified by the Transformer [306] and BERT [78] models and their applications to retrieval [281] and video understanding [290]. Also, this setting offers novel architectural modeling capabilities, as for the first time the attention operator is used in a multi-layer and extensible fashion. Nevertheless, the multimodal nature of image captioning demands for specific architectures, different from those employed for the understanding of a single modality.

Following these premises, in this section we investigate the design of a novel fully-attentive approach for image captioning. Our architecture takes inspiration from the Transformer model [306] for machine translation and incorporates two key novelties with respect to all previous image captioning algorithms: (*i*) image regions and their relationships are encoded in a multi-level fashion, in which low-level and high-level relations are taken into account. When modeling these relationships, our model can learn and encode a priori knowledge by using persistent *memory vectors*. (*ii*) The generation of the sentence, done with a multi-layer architecture, exploits both low- and high-level visual relationships instead of having just a single input from the visual modality. This is achieved through a learned gating mechanism, which weights multi-level contributions at each stage. As this creates a mesh connectivity schema between encoder and decoder layers, we name our model *Meshed-Memory Transformer* – $\mathcal{M}^2$ Transformer for short. Figure 2.10 depicts a schema of the architecture.

Experimentally, we explore different fully-attentive baselines and recent proposals, gaining insights on the performance of fully-attentive models in image captioning. Our $\mathcal{M}^2$ Transformer, when tested on the COCO benchmark, achieves a new state of the art on the “Karpathy” test set, on both single-model and en-

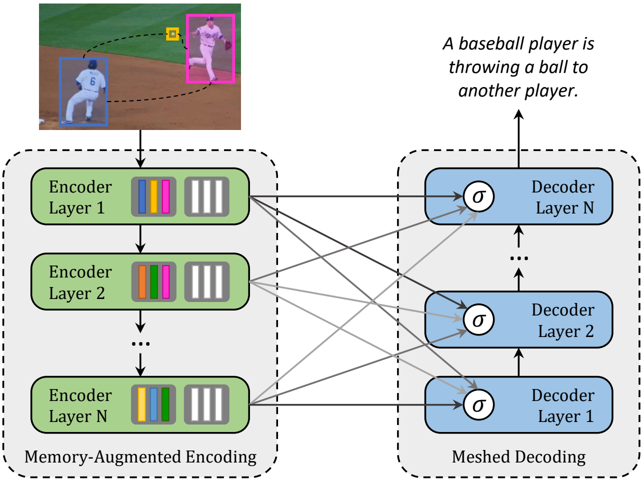


Figure 2.10: Our image captioning approach encodes relationships between image regions exploiting learned a priori knowledge. Multi-level encodings of image regions are connected to a language decoder through a meshed and learnable connectivity.

semble configurations. Most importantly, it surpasses existing proposals on the online test server, ranking first among published algorithms.

To foster researches in this field and the reproducibility of our work, the source code and trained models of our $\mathcal{M}^2$ Transformer are publicly available[3].

2.2.1 Meshed-Memory Transformer architecture

Our model can be conceptually divided into an encoder and a decoder module, both made of stacks of attentive layers. While the encoder is in charge of processing regions from the input image and devising relationships between them, the decoder reads from the output of each encoding layer to generate the output caption word by word. All intra-modality and cross-modality interactions between word and image-level features are modeled via scaled dot-product attention, without using recurrence. Attention operates on three sets of vectors, namely a set of queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$, and takes a weighted sum of value vectors according

[3] https://github.com/aimagelab/meshed-memory-transformer

to a similarity distribution between query and key vectors. In the case of scaled dot-product attention, the operator can be formally defined as

$$\mathsf{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V}, \tag{2.6}$$

where $\boldsymbol{Q}$ is a matrix of n_q query vectors, $\boldsymbol{K}$ and $\boldsymbol{V}$ both contain n_k keys and values, all with the same dimensionality, and d is a scaling factor.

Memory-augmented encoder

Given a set of image regions $\boldsymbol{X}$ extracted from an input image, attention can be used to obtain a permutation invariant encoding of $\boldsymbol{X}$ through the self-attention operations used in the Transformer [306]. In this case, queries, keys, and values are obtained by linearly projecting the input features, and the operator can be defined as

$$\mathcal{S}(\boldsymbol{X}) = \mathsf{Attention}(W_q\boldsymbol{X}, W_k\boldsymbol{X}, W_v\boldsymbol{X}), \tag{2.7}$$

where W_q, W_k, W_v are matrices of learnable weights. The output of the self-attention operator is a new set of elements $\mathcal{S}(\boldsymbol{X})$, with the same cardinality as $\boldsymbol{X}$, in which each element of $\boldsymbol{X}$ is replaced with a weighted sum of the values, *i.e.* of linear projections of the input (Eq. 2.6).

Noticeably, attentive weights depend solely on the pairwise similarities between linear projections of the input set itself. Therefore, the self-attention operator can be seen as a way of encoding pairwise relationships inside the input set. When using image regions (or features derived from image regions) as the input set, $\mathcal{S}(\cdot)$ can naturally encode the pairwise relationships between regions that are needed to understand the input image before describing it[4].

This peculiarity in the definition of self-attention has, however, a significant limitation. Because everything depends solely on pairwise similarities, self-attention cannot model a priori knowledge on relationships between image regions. For example, given one region encoding a man and a region encoding a basketball ball, it would be difficult to infer the concept of *player* or *game* without any a priori knowledge. Again, given regions encoding eggs and toasts, the knowledge that the picture depicts a *breakfast* could be easily inferred using a priori knowledge on relationships.

[4] Taking another perspective, self-attention is also conceptually equivalent to an attentive encoding of graph nodes [308].

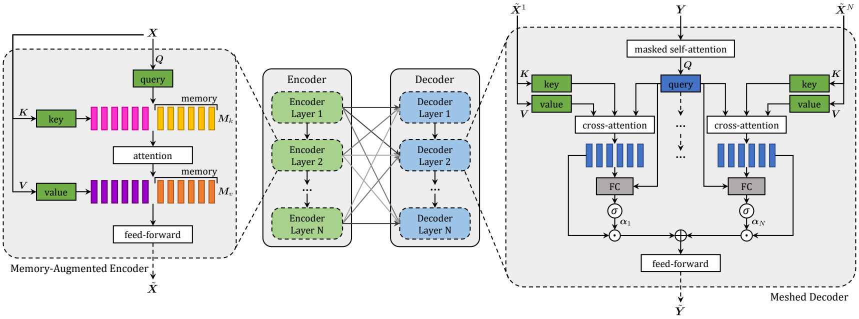


Figure 2.11: Architecture of the $\mathcal{M}^2$ Transformer. Our model is composed of a stack of memory-augmented encoding layers, which encodes multi-level visual relationships with a priori knowledge, and a stack of decoder layers, in charge of generating textual tokens. For the sake of clarity, AddNorm operations are not shown. Best seen in color.

Memory-augmented attention. To overcome this limitation of self-attention, we propose a memory-augmented attention operator. In our proposal, the set of keys and values used for self-attention is extended with additional "slots" which can encode a priori information. To stress that a priori information should not depend on the input set $\boldsymbol{X}$, the additional keys and values are implemented as plain learnable vectors which can be directly updated via SGD. Formally, the operator is defined as:

$$\begin{aligned} \mathcal{M}_{\text{mem}}(\boldsymbol{X}) &= \mathsf{Attention}(W_q\boldsymbol{X}, \boldsymbol{K}, \boldsymbol{V}) \\ \boldsymbol{K} &= [W_k\boldsymbol{X}, \boldsymbol{M}_k] \\ \boldsymbol{V} &= [W_v\boldsymbol{X}, \boldsymbol{M}_v]\,, \end{aligned} \tag{2.8}$$

where $\boldsymbol{M}_k$ and $\boldsymbol{M}_v$ are learnable matrices with n_m rows, and $[\cdot,\cdot]$ indicates concatenation. Intuitively, by adding learnable keys and values, through attention it will be possible to retrieve learned knowledge which is not already embedded in $\boldsymbol{X}$. At the same time, our formulation leaves the set of queries unaltered. Intuitively again, this will help to avoid hallucination, given that knowledge is always retrieved because of similarities with queries which are seen in the image.

Just like the self-attention operator, our memory-augmented attention can be applied in a multi-head fashion. In this case, the memory-augmented attention operation is repeated h times, using different projection matrices W_q, W_k, W_v and different learnable memory slots $\boldsymbol{M}_k, \boldsymbol{M}_v$ for each head. Then, we concatenate the results from different heads and apply a linear projection.

Encoding layer. We embed our memory-augmented operator into a Transformer-like layer: the output of the memory-augmented attention is applied to a position-wise feed-forward layer composed of two affine transformations with a single non-linearity, which are independently applied to each element of the set. Formally,

$$\mathcal{F}(\boldsymbol{X})_i = U\sigma(V\boldsymbol{X}_i + b) + c, \tag{2.9}$$

where $\boldsymbol{X}_i$ indicates the i-th vector of the input set, and $\mathcal{F}(\boldsymbol{X})_i$ the i-th vector of the output. Also, $\sigma(\cdot)$ is the ReLU activation function, V and U are learnable weight matrices, b and c are bias terms.

Each of these sub-components (memory-augmented attention and position-wise feed-forward) is then encapsulated within a residual connection and a layer norm operation. The complete definition of an encoding layer can be finally written as:

$$\begin{aligned} \boldsymbol{Z} &= \mathsf{AddNorm}(\mathcal{M}_{\text{mem}}(\boldsymbol{X})) \\ \tilde{\boldsymbol{X}} &= \mathsf{AddNorm}(\mathcal{F}(\boldsymbol{Z})), \end{aligned} \tag{2.10}$$

where $\mathsf{AddNorm}$ indicates the composition of a residual connection and of a layer normalization.

Full encoder. Given the aforementioned structure, multiple encoding layers are stacked in sequence, so that the i-th layer consumes the output set computed by layer $i-1$. This amounts to creating multi-level encodings of the relationships between image regions, in which higher encoding layers can exploit and refine relationships already identified by previous layers, eventually using a priori knowledge. A stack of N encoding layers will therefore produce a multi-level output $\tilde{\mathcal{X}} = (\tilde{\boldsymbol{X}}^1, ..., \tilde{\boldsymbol{X}}^N)$, obtained from the outputs of each encoding layer.

Meshed decoder

Our decoder is conditioned on both previously generated words and region encodings, and is in charge of generating the next tokens of the output caption. Here, we exploit the aforementioned multi-level representation of the input image while still building a multi-layer structure. To this aim, we devise a meshed attention operator which, unlike the cross-attention operator of the Transformer, can take advantage of all encoding layers during the generation of the sentence.

Meshed cross-attention. Given an input sequence of vectors $\boldsymbol{Y}$, and outputs from all encoding layers $\tilde{\mathcal{X}}$, the Meshed Attention operator connects $\boldsymbol{Y}$ to all elements in $\tilde{\mathcal{X}}$ through gated cross-attentions. Instead of attending only the last encoding layer, we perform a cross-attention with all encoding layers. These multi-level contributions are then summed together after being modulated. Formally, our meshed attention operator is defined as

$$\mathcal{M}_{\text{mesh}}(\tilde{\mathcal{X}}, \boldsymbol{Y}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \odot \mathcal{C}(\tilde{\boldsymbol{X}}^i, \boldsymbol{Y}), \tag{2.11}$$

where $\mathcal{C}(\cdot, \cdot)$ stands for the encoder-decoder cross-attention, computed using queries from the decoder and keys and values from the encoder:

$$\mathcal{C}(\tilde{\boldsymbol{X}}^i, \boldsymbol{Y}) = \mathsf{Attention}(W_q\boldsymbol{Y}, W_k\tilde{\boldsymbol{X}}^i, W_v\tilde{\boldsymbol{X}}^i), \tag{2.12}$$

and $\boldsymbol{\alpha}_i$ is a matrix of weights having the same size as the cross-attention results. Weights in $\boldsymbol{\alpha}_i$ modulate both the single contribution of each encoding layer, and the relative importance between different layers. These are computed by measuring the relevance between the result of the cross-attention computed with each encoding layer and the input query, as follows:

$$\boldsymbol{\alpha}_i = \sigma\left(W_i\left[\boldsymbol{Y}, \mathcal{C}(\tilde{\boldsymbol{X}}^i, \boldsymbol{Y})\right] + b_i\right), \tag{2.13}$$

where $[\cdot, \cdot]$ indicates concatenation, σ is the sigmoid activation, W_i is a $2d \times d$ weight matrix, and b_i is a learnable bias vector.

Architecture of decoding layers. As for encoding layers, we apply our meshed attention in a multi-head fashion. As the prediction of a word should only depend on previously predicted words, the decoder layer comprises a masked self-attention operation which connects queries derived from the t-th element of its input sequence $\boldsymbol{Y}$ with keys and values obtained from the left-hand subsequence, *i.e.* $\boldsymbol{Y}_{\leq t}$. Also, the decoder layer contains a position-wise feed-forward layer (as in Eq. 2.9), and all components are encapsulated within $\mathsf{AddNorm}$ operations. The final structure of the decoder layer can be written as:

$$\begin{aligned}
\boldsymbol{Z} &= \mathsf{AddNorm}(\mathcal{M}_{\text{mesh}}(\mathsf{AddNorm}(\mathcal{S}_{\text{mask}}(\boldsymbol{Y}))) \\
\tilde{\boldsymbol{Y}} &= \mathsf{AddNorm}(\mathcal{F}(\boldsymbol{Z})),
\end{aligned} \tag{2.14}$$

where $\boldsymbol{Y}$ is the input sequence of vectors and $\mathcal{S}_{\text{mask}}$ indicates a masked self-attention over time. Finally, our decoder stacks together multiple decoder layers, helping to refine both the understanding of the textual input and the generation of next tokens. Overall, the decoder takes as input word vectors, and the t-th element of its output sequence encodes the prediction of a word at time $t+1$, conditioned on $\boldsymbol{Y}_{\leq t}$. After taking a linear projection and a softmax operation, this encodes a probability over words in the dictionary.

2.2.2 Experimental evaluation

Training

Following a standard practice in image captioning [255, 262, 7], we pre-train our model with a word-level cross-entropy loss (XE) and finetune the sequence generation using reinforcement learning. When training with XE, the model is trained to predict the next token given previous ground-truth words; in this case, the input sequence for the decoder is immediately available and the computation of the entire output sequence can be done in a single pass, parallelizing all operations over time.

When training with reinforcement learning, we employ a variant of the self-critical sequence training approach [262] on sequences sampled using beam search [7]: to decode, we sample the top-k words from the decoder probability distribution at each timestep, and always maintain the top-k sequences with highest probability. As sequence decoding is iterative in this step, the aforementioned parallelism over time cannot be exploited. However, intermediate keys and values used to compute the output token at time t can be reused in the next iterations.

Following previous works [7], we use the CIDEr-D score as reward, as it well correlates with human judgment [307]. We baseline the reward using the mean of the rewards rather than greedy decoding as done in previous methods [262, 7], as we found it to slightly improve the final performance. The final gradient expression for one sample is thus:

$$\nabla_\theta L(\theta) = -\frac{1}{k}\sum_{i=1}^{k}\big((r(\boldsymbol{w}^i) - b)\nabla_\theta \log p(\boldsymbol{w}^i)\big) \tag{2.15}$$

where $\boldsymbol{w}^i$ is the i-th sentence in the beam, $r(\cdot)$ is the reward function, and $b = \left(\sum_i r(\boldsymbol{w}^i)\right)/k$ is the baseline, computed as the mean of the rewards obtained by the sampled sequences. At prediction time, we decode again using beam search, and keep the sequence with highest predicted probability among those in the last beam.

Datasets

We first evaluate our model on the COCO dataset [201], which is the most commonly used test-bed for image captioning. Then, we assess the captioning of novel objects by testing on the recently proposed nocaps dataset [2].

COCO. As previously mentioned, the dataset contains more than $120,000$ images, each of them annotated with 5 different captions. We follow the splits provided by Karpathy *et al.* [159], where $5,000$ images are used for validation, $5,000$ for testing and the rest for training. We also evaluate the model on the COCO online test server, composed of $40,775$ images for which annotations are not made publicly available.

nocaps. The dataset consists of $15,100$ images taken from the Open Images [180] validation and test sets, each annotated with 11 human-generated captions. Images are divided into validation and test splits, respectively composed of $4,500$ and $10,600$ elements. Images can be further grouped into three subsets depending on the nearness to COCO, namely in-domain, near-domain, and out-of-domain images. In-domain images contain only objects that are described in the COCO captions, out-of-domain images contain object classes that do not appear in COCO captions, and near-domain images contain both in-domain and out-of-domain object classes. Under this setting, we use COCO as training data and evaluate our results on the nocaps test server.

Metrics

Following the standard evaluation protocol, we employ the full set of captioning metrics: BLEU [238], METEOR [19], ROUGE [199], CIDEr [307], and SPICE [5].

Implementation details

To represent image regions, we use Faster R-CNN [259] with ResNet-101 [119] finetuned on the Visual Genome dataset [175, 7], thus obtaining a 2048-dimensional feature vector for each region. To represent words, we use one-hot vectors and linearly project them to the input dimensionality of the model d. We also employ sinusoidal positional encodings [306] to represent word positions inside the sequence and sum the two embeddings before the first decoding layer.

Pre-training with XE is done following the learning rate scheduling strategy of [306] with a warmup equal to $10,000$ iterations. Then, during CIDEr-D optimization, we use a fixed learning rate of 5×10^{-6}. We train all models using the Adam optimizer [170], a batch size of 50, and a beam size equal to 5.

Decoding optimization. As mentioned in Section 2.2.2, during the decoding stage computation cannot be parallelized over time as the input sequence is iteratively built. A naive approach would be to feed the model at each iteration with the previous $t-1$ generated words, $\{w_0, w_1, ..., w_{t-1}\}$ and sample the next predicted word w_t after computing the results of each attention and feed-forward layer over all timesteps. This in practice requires to re-compute the same queries, keys, values and attentive states multiple times, with intermediate results depending on w_t being recomputed $T-t$ times, where T is the length of the sampled sequence (in our experiments T is equal to 20).

In our implementation, we revert to a more computationally friendly approach in which we re-use intermediate results computed at previous timesteps. Each attentive layer of the decoder internally stores previously computed keys and values. At each timestep of the decoding, the model is fed only with w_{t-1}, and we only compute queries, keys and values depending on w_{t-1}.

In PyTorch, this can be implemented by exploiting the `register_buffer` method of `nn.Module`, and creating buffers to hold previously computed results. When running on a NVIDIA 2080Ti GPU, we found this to reduce training and inference times by approximately a factor of 3.

Vocabulary and tokenization. We convert all captions to lowercase, remove punctuation characters and tokenize using the spaCy NLP toolkit[5]. To build vocabularies, we remove all words which appear less than 5 times in training and validation splits. For each image, we use a maximum number of region feature vectors equal to 50.

Model dimensionality and weight initialization. In our model, we set the dimensionality d of each layer to 512, the number of heads to 8, and the number of memory vectors to 40. Using 8 attentive heads, the size of queries, keys and values in each head is set to $d/8 = 64$. We employ dropout with keep probability 0.9 after each attention and feed-forward layer. In our meshed attention operator (Eq. 2.11), we normalize the output with a scaling factor of $\sqrt{N}$.

Weights of attentive layers are initialized from the uniform distribution proposed by Glorot *et al.* [103], while weights of feed-forward layers are initialized using [118]. All biases are initialized to 0. Memory vectors for keys and values are initialized from a normal distribution with zero mean and, respectively, $1/d_k$

[5] `https://spacy.io/`

and $1/m$ variance, where d_k is the dimensionality of keys and m is the number of memory vectors.

Novel object captioning. To train the model on the nocaps dataset, instead of using one-hot vectors, we represent words with GloVe word embeddings [242]. Two fully-connected layers are added to convert between the GloVe dimensionality and d before the first decoding layer and after the last decoding layer. Before the final softmax, we multiply with the transpose of the word embeddings.

Following [2], we use an object detector trained on Open Images [6] and filter detections by removing 39 Open Images classes that contain parts of objects or which are seldom mentioned. We also discard overlapping detections by removing the higher-order of two objects based on the class hierarchy, and we use the top-3 detected objects as constraints based on the detection confidence score. Differently from [2], we do not consider the plural forms or other word phrases of object classes, thus taking into account only the original class names. After decoding, we select the predicted caption with highest probability that satisfies the given constraints.

2.2.3 Ablation study

Performance of the Transformer. In previous works, the Transformer model has been applied to captioning only in its original configuration with six layers and self/cross attention, with the structure of connections that has been successful for uni-modal scenarios like machine translation. As we speculate that captioning requires specific architectures, we compare variations of the original Transformer with our approach.

Firstly, we investigate the impact of the number of encoding and decoding layers on captioning performance. As it can be seen in Table 2.3, the original Transformer (six layers) achieves 121.8 CIDEr, slightly superior to the Up-Down approach [7] which uses a two-layer recurrent language model with additive attention and includes a global feature vector (120.1 CIDEr). Varying the number of layers, we observe a significant increase in performance when using three encoding and three decoding layers, which leads to 123.6 CIDEr. We hypothesize that this is due to the reduced training set size and to the lower semantic complexities of sentences in captioning with respect to those of language understanding tasks. Following this finding, all subsequent experiments will use three layers.

[6] Specifically, the `tf_faster_rcnn_inception_resnet_v2_atrous_oidv2` model from the Tensorflow model zoo.

	B-1	B-4	M	R	C	S
Transformer (w/ 6 layers as in [306])	79.1	36.2	27.7	56.9	121.8	20.9
Transformer (w/ 3 layers)	79.6	36.5	27.8	57.0	123.6	21.1
Transformer (w/ AoA [137])	80.3	38.8	29.0	58.4	129.1	**22.7**
$\mathcal{M}^2$ Transformer$^{\text{1-to-1}}$ (w/o mem.)	80.5	38.2	28.9	58.2	128.4	22.2
$\mathcal{M}^2$ Transformer$^{\text{1-to-1}}$	80.3	38.2	28.9	58.2	129.2	22.5
$\mathcal{M}^2$ Transformer (w/o mem.)	80.4	38.3	29.0	58.2	129.4	22.6
$\mathcal{M}^2$ Transformer (w/ softmax)	80.3	38.4	29.1	58.3	130.3	22.5
$\mathcal{M}^2$ **Transformer**	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	22.6

Table 2.3: Ablation study and comparison with Transformer-based alternatives. All results are reported after the REINFORCE optimization stage.

Attention on Attention baseline. We also evaluate a recent proposal that can be straightforwardly applied to the Transformer as an alternative to standard dot-product attention. Specifically, we evaluate the addition of the “Attention on Attention” (AoA) approach [137] to the attentive layers, both in the encoder and in the decoder. Noticeably, in [137] this has been done with a Recurrent language model with attention, but the approach is sufficiently general to be applied to any attention stage. In this case, the result of dot-product attention is concatenated with the initial query and fed to two fully connected layers to obtain an information vector and a sigmoidal attention gate, then the two vectors are multiplied together. The final result is used as an alternative to the standard dot-product attention. This addition to a standard Transformer with three layers leads to 129.1 CIDEr (Table 2.3), thus underlying the usefulness of the approach also in Transformer-based models.

Meshed connectivity. We then evaluate the role of the meshed connections between encoder and decoder layers. In Table 2.3, we firstly introduce a reduced version of our approach in which the i-th decoder layer is only connected to the corresponding i-th encoder layer (1-to-1), instead of being connected to all encoders. As it can be noticed, using this 1-to-1 connectivity schema already brings an improvement with respect to using the output of the last encoder layer as in the standard Transformer (123.6 CIDEr vs 129.2 CIDEr), thus confirming that exploiting a multi-level encoding of image regions is beneficial. When we instead use our meshed connectivity schema, that exploits relationships encoded at all levels and weights them with a sigmoid gating, we observe a further performance improvement, from 129.2 CIDEr to 131.2 CIDEr. This amounts to a total improvement of 7.6 CIDEr points with respect to the standard Transformer. Also,

Memories	B-1	B-4	M	R	C	S
No memory	80.4	38.3	29.0	58.2	129.4	22.6
20	80.7	38.9	29.0	58.4	129.9	22.7
40	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	22.6
60	80.0	37.9	28.9	58.1	129.6	22.5
80	80.0	38.2	29.0	58.3	128.9	**22.9**

Table 2.4: Captioning results of $\mathcal{M}^2$ Transformer using different numbers of memory vectors.

Layers	B-1	B-4	M	R	C	S
2	80.5	38.6	29.0	58.4	128.5	**22.8**
3	**80.8**	**39.1**	**29.2**	**58.6**	**131.2**	22.6
4	**80.8**	38.6	29.1	58.5	129.6	22.6

Table 2.5: Captioning results of $\mathcal{M}^2$ Transformer using different numbers of encoder and decoder layers.

the result of our full model is superior to that obtained using the AoA.

As an alternative to the sigmoid gating approach for weighting the contributions from different encoder layers (Eq. 2.11), we also test with a softmax gating schema. In this case, the element-wise sigmoid applied to each encoder is replaced with the application of a softmax operation over the rows of α_i. Using this alternative brings to a reduction of around 1 CIDEr point, underlying that it is beneficial to exploit the full potentiality of a weighted sum of the contributions from all encoding layers, rather than forcing a peaky distribution in which one layer is given more importance than the others.

Role of persistent memory. We evaluate the role of memory vectors in both the 1-to-1 configuration and in the final configuration with meshed connections. As it can be seen from Table 2.3, removing memory vectors brings to a reduction in performance of around 1 CIDEr point in both connectivity settings, thus confirming the usefulness of exploiting a priori learned knowledge when encoding image regions.

In Table 2.4, we report the performance of our approach when using a varying number of memory vectors. As it can be seen, the best result in terms of BLEU, METEOR, ROUGE and CIDEr is obtained with 40 memory vectors, while 80 memory vectors provide a slightly superior result in terms of SPICE.

Encoder and decoder layers. We also investigate the performance of the

	B-1	B-4	M	R	C	S
SCST [262]	-	34.2	26.7	55.7	114.0	-
Up-Down [7]	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [154]	79.1	36.5	27.7	57.3	121.9	21.2
Up-Down+HIP [358]	-	38.2	28.4	58.3	127.2	21.9
GCN-LSTM [357]	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [347]	**80.8**	38.4	28.4	58.6	127.8	22.1
ORT [124]	80.5	38.6	28.7	58.4	128.3	**22.6**
AoANet [137]	80.2	38.9	**29.2**	**58.8**	129.8	22.4
$\mathcal{M}^2$ **Transformer**	**80.8**	**39.1**	**29.2**	58.6	**131.2**	**22.6**

Table 2.6: Comparison with the state of the art on the "Karpathy" test split, in single-model setting.

$\mathcal{M}^2$ Transformer when changing the number of encoding and decoding layers. Table 2.5 shows that the best performance is obtained with three encoding and decoding layers, thus confirming the initial findings on the base Transformer model. As our model can deal with a different number of encoding and decoding layers, we also experimented with non symmetric encoding-decoding architectures, without however noticing significant improvements in performance.

2.2.4 Comparison with state of the art

We compare the performances of our approach with those of several recent proposals for image captioning. The models we compare to include SCST [262], which uses attention over the grid of features and a one-layer LSTM language model; Up-Down [7], which introduces attention over regions, and uses a two-layer LSTM language model. Also, we compare to the RFNet approach [154], which uses a recurrent fusion network to merge different CNN features; GCN-LSTM [357], which exploits pairwise relationships between image regions through a Graph Convolutional Neural Network; SGAE [347], which instead uses auto-encoding scene graphs. Further, we compare with the original AoANet [137] approach, which uses attention on attention for encoding image regions and an LSTM language model. Finally, we compare with ORT [124], which uses a plain Transformer, and weights attention scores in the region encoder with pairwise distances between detections.

We evaluate our approach on the COCO "Karpathy" test split, using both single model and ensemble configurations, and on the online COCO evaluation server.

	B-1	B-4	M	R	C	S
Ensemble/Fusion of 2 models						
GCN-LSTM [357]	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [347]	81.0	39.0	28.4	58.9	129.1	22.2
ETA [190]	81.5	**39.9**	28.9	59.0	127.6	22.6
GCN-LSTM+HIP [358]	-	39.1	28.9	**59.2**	130.6	22.3
$\mathcal{M}^2$ **Transformer**	**81.6**	39.8	**29.5**	**59.2**	**133.2**	**23.1**
Ensemble/Fusion of 4 models						
SCST [262]	-	35.4	27.1	56.6	117.5	-
RFNet [154]	80.4	37.9	28.3	58.3	125.7	21.7
AoANet [137]	81.6	40.2	29.3	59.4	132.0	22.8
$\mathcal{M}^2$ **Transformer**	**82.0**	**40.5**	**29.7**	**59.5**	**134.5**	**23.5**

Table 2.7: Comparison with the state of the art on the "Karpathy" test split, using an ensemble of models.

Single model. In Table 2.6 we report the performance of our method in comparison with the aforementioned competitors, using captions predicted from a single model and optimization on the CIDEr-D score. As it can be observed, our method surpasses all other approaches in terms of BLEU-4, METEOR and CIDEr, while being competitive on BLEU-1 and SPICE with the best performer, and slightly worse on ROUGE with respect to AoANet [137]. In particular, it advances the current state of the art on CIDEr by 1.4 points.

Ensemble model. Following the common practice [262, 137] of building an ensemble of models, we also report the performances of our approach when averaging the output probability distributions of multiple and independently trained instances of our model. In Table 2.7, we use ensembles of two and four models, trained from different random seeds. Noticeably, when using four models our approach achieves the best performance according to all metrics, with an increase of 2.5 CIDEr points with respect to the current state of the art [137].

Online Evaluation. Finally, we also report the performance of our method on the online COCO test server[7]. In this case, we use the ensemble of four models previously described, trained on the "Karpathy" training split. The evaluation is done on the COCO test split, for which ground-truth annotations are not publicly available. Results are reported in Table 2.8, in comparison with the top-performing approaches of the leaderboard. For fairness of comparison, they also used an

[7] https://competitions.codalab.org/competitions/3221

ensemble configuration. As it can be seen, our method surpasses the current state of the art on all metrics, achieving an advancement of 1.4 CIDEr points with respect to the best performer.

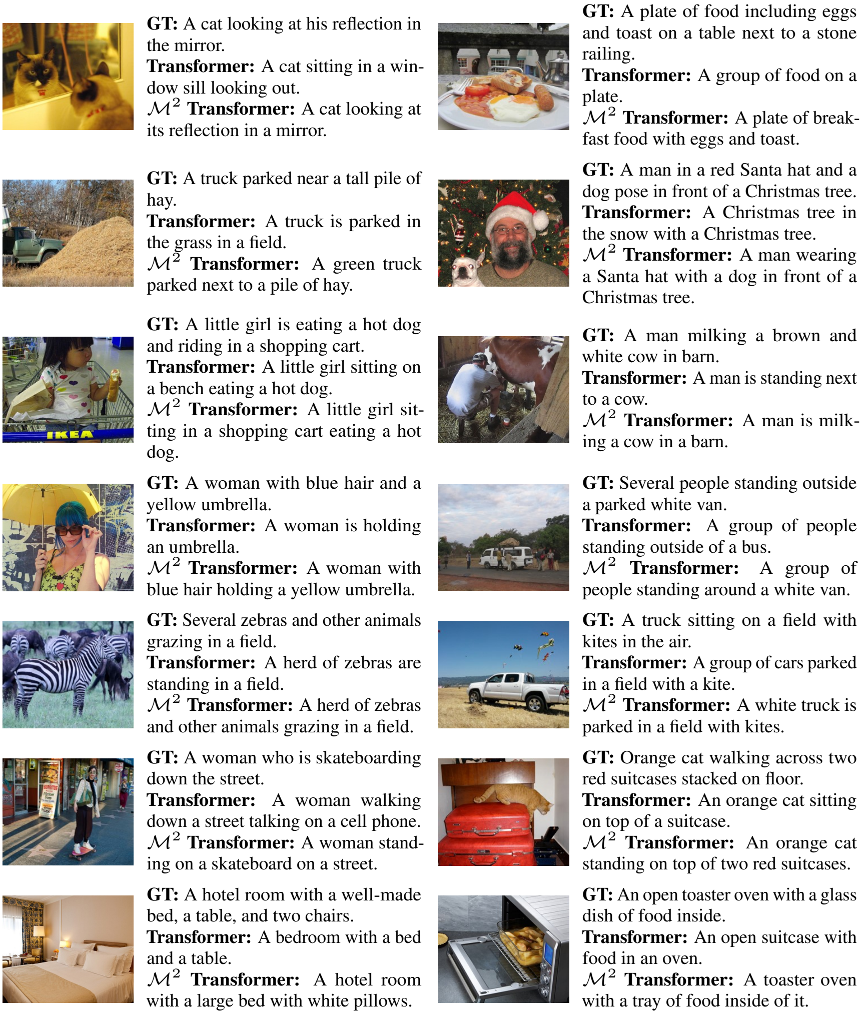
2.2.5 Qualitative results and visualization

Figure 2.12 proposes qualitative results generated by our model and the original Transformer. On average, our model is able to generate more accurate and descriptive captions, integrating fine-grained details and object relations.

Finally, to better understand the effectiveness of our $\mathcal{M}^2$ Transformer, we investigate the contribution of detected regions to the model output. Differently from recurrent-based captioning models, in which attention weights over regions can be easily extracted, in our model the contribution of one region with respect to the output is given by more complex non-linear dependencies. Therefore, we revert to attribution methods and we employ the Integrated Gradients approach [291], which approximates the integral of gradients with respect to the given input. Specifically, the Integrated Gradients approach produces an attribution score for each feature channel of each input region. To obtain the attribution of each region, we average over the feature channels, and re-normalize the obtained scores by their sum. For visualization purposes, we apply a contrast stretching function to project scores in the 0-1 interval. Results are presented in Figure 2.13, where we observe that our approach correctly grounds image regions to words, also in presence of object details and small detections.

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [262]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7
Up-Down [7]	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
RDN [162]	80.2	95.3	-	-	-	-	37.3	69.5	28.1	37.8	57.4	73.3	121.2	125.2
RFNet [154]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
GCN-LSTM [357]	80.8	95.9	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [347]	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
ETA [190]	81.2	95.0	65.5	89.0	50.9	80.4	38.9	70.2	28.6	38.0	58.6	73.9	122.1	124.4
AoANet [137]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
GCN-LSTM+HIP [358]	**81.6**	95.9	66.2	90.4	51.5	81.6	39.3	71.0	28.8	38.1	59.0	74.1	127.9	130.2
$\mathcal{M}^2$ **Transformer**	**81.6**	**96.0**	**66.4**	**90.8**	**51.8**	**82.7**	**39.7**	**72.8**	**29.4**	**39.0**	**59.2**	**74.8**	**129.3**	**132.1**

Table 2.8: Leaderboard of various methods on the online MS-COCO test server.



GT: A cat looking at his reflection in the mirror.
Transformer: A cat sitting in a window sill looking out.
$\mathcal{M}^2$ **Transformer:** A cat looking at its reflection in a mirror.

GT: A truck parked near a tall pile of hay.
Transformer: A truck is parked in the grass in a field.
$\mathcal{M}^2$ **Transformer:** A green truck parked next to a pile of hay.

GT: A little girl is eating a hot dog and riding in a shopping cart.
Transformer: A little girl sitting on a bench eating a hot dog.
$\mathcal{M}^2$ **Transformer:** A little girl sitting in a shopping cart eating a hot dog.

GT: A woman with blue hair and a yellow umbrella.
Transformer: A woman is holding an umbrella.
$\mathcal{M}^2$ **Transformer:** A woman with blue hair holding a yellow umbrella.

GT: Several zebras and other animals grazing in a field.
Transformer: A herd of zebras are standing in a field.
$\mathcal{M}^2$ **Transformer:** A herd of zebras and other animals grazing in a field.

GT: A woman who is skateboarding down the street.
Transformer: A woman walking down a street talking on a cell phone.
$\mathcal{M}^2$ **Transformer:** A woman standing on a skateboard on a street.

GT: A hotel room with a well-made bed, a table, and two chairs.
Transformer: A bedroom with a bed and a table.
$\mathcal{M}^2$ **Transformer:** A hotel room with a large bed with white pillows.

GT: A plate of food including eggs and toast on a table next to a stone railing.
Transformer: A group of food on a plate.
$\mathcal{M}^2$ **Transformer:** A plate of breakfast food with eggs and toast.

GT: A man in a red Santa hat and a dog pose in front of a Christmas tree.
Transformer: A Christmas tree in the snow with a Christmas tree.
$\mathcal{M}^2$ **Transformer:** A man wearing a Santa hat with a dog in front of a Christmas tree.

GT: A man milking a brown and white cow in barn.
Transformer: A man is standing next to a cow.
$\mathcal{M}^2$ **Transformer:** A man is milking a cow in a barn.

GT: Several people standing outside a parked white van.
Transformer: A group of people standing outside of a bus.
$\mathcal{M}^2$ **Transformer:** A group of people standing around a white van.

GT: A truck sitting on a field with kites in the air.
Transformer: A group of cars parked in a field with a kite.
$\mathcal{M}^2$ **Transformer:** A white truck is parked in a field with kites.

GT: Orange cat walking across two red suitcases stacked on floor.
Transformer: An orange cat sitting on top of a suitcase.
$\mathcal{M}^2$ **Transformer:** An orange cat standing on top of two red suitcases.

GT: An open toaster oven with a glass dish of food inside.
Transformer: An open suitcase with food in an oven.
$\mathcal{M}^2$ **Transformer:** A toaster oven with a tray of food inside of it.

Figure 2.12: Examples of captions generated by our approach and the original Transformer model, as well as the corresponding ground-truths.

	In-Domain		Out-of-Domain		Overall	
	CIDEr	SPICE	CIDEr	SPICE	CIDEr	SPICE
NBT + CBS [2]	62.1	10.1	62.4	8.9	60.2	9.5
Up-Down + CBS [2]	80.0	12.0	66.4	9.7	73.1	11.1
Transformer	78.0	11.0	29.7	7.8	54.7	9.8
$\mathcal{M}^2$ Transformer	**85.7**	**12.1**	38.9	8.9	64.5	11.1
Transformer + CBS	74.3	11.0	62.5	9.2	66.9	10.3
$\mathcal{M}^2$ Transformer + CBS	81.2	12.0	**69.4**	**10.0**	**75.0**	**11.4**

Table 2.9: Performances on nocaps validation set, for in-domain and out-of-domain captioning.

2.2.6 Describing novel objects

We also assess the performance of our approach when dealing with images containing object categories that are not seen in the training set. We compare with the Up-Down model [7] and Neural Baby Talk [215], when using GloVe word embeddings and Constrained Beam Search (CBS) [6] to address the generation of out-of-vocabulary words and constrain the presence of categories detected by an object detector. To compare with our model, we use a simplified implementation of the procedure described in [2] to extract constraints, without using word phrases (*e.g.* plurals).

Results are shown in Table 2.9: as it can be seen, the original Transformer is significantly less performing than Up-Down on both in-domain and out-of-domain categories, while our approach can properly deal with novel categories, surpassing the Up-Down baseline in both in-domain and out-of-domain images. As expected, the use of CBS significantly enhances the performances, in particular on out-of-domain captioning.

Figure 2.14 reports sample captions produced by our approach on images from the nocaps dataset. On each image, we compare to the baseline Transformer and show the constraints provided by the object detector. Overall, the $\mathcal{M}^2$ Transformer is able to better incorporate the constraints while maintaining the fluency and properness of the generated sentences.

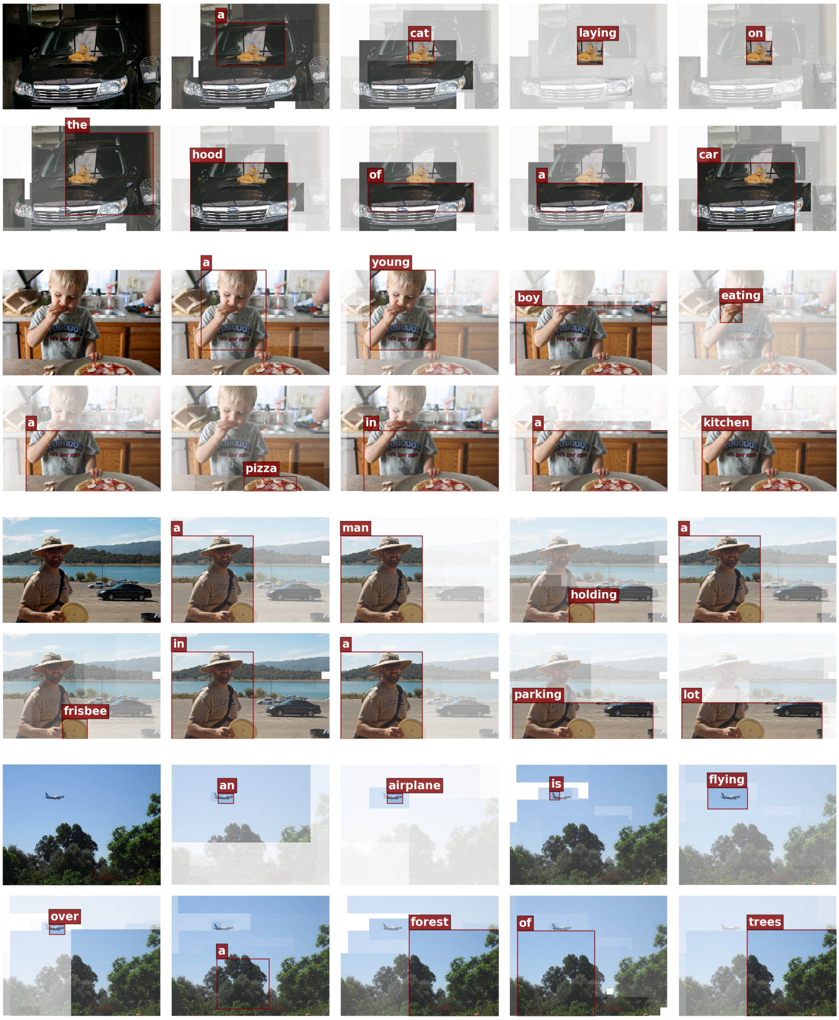
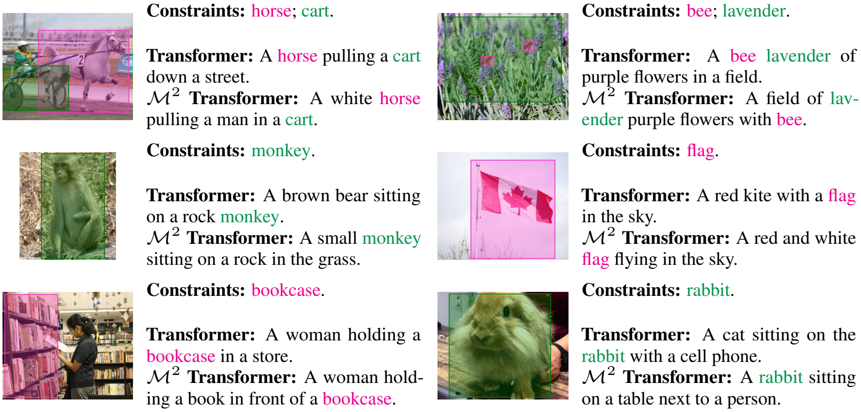


Figure 2.13: Visualization of attention states for four sample captions. For each generated word, we show the attended image regions, outlining the region with the maximum output attribution in red.



Constraints: horse; cart.

Transformer: A horse pulling a cart down a street.
$\mathcal{M}^2$ **Transformer:** A white horse pulling a man in a cart.

Constraints: bee; lavender.

Transformer: A bee lavender of purple flowers in a field.
$\mathcal{M}^2$ **Transformer:** A field of lavender purple flowers with bee.

Constraints: monkey.

Transformer: A brown bear sitting on a rock monkey.
$\mathcal{M}^2$ **Transformer:** A small monkey sitting on a rock in the grass.

Constraints: flag.

Transformer: A red kite with a flag in the sky.
$\mathcal{M}^2$ **Transformer:** A red and white flag flying in the sky.

Constraints: bookcase.

Transformer: A woman holding a bookcase in a store.
$\mathcal{M}^2$ **Transformer:** A woman holding a book in front of a bookcase.

Constraints: rabbit.

Transformer: A cat sitting on the rabbit with a cell phone.
$\mathcal{M}^2$ **Transformer:** A rabbit sitting on a table next to a person.

Figure 2.14: Sample nocaps images and corresponding predicted captions generated by our model and the original Transformer. For each image, we report the Open Images object classes predicted by the object detector and used as constraints during the generation of the caption.

2.3 CaMEL: mean teacher learning

As shown in the previous sections, describing images in natural language is a fundamental step towards the automatic modeling of connections between the visual and textual modalities. In this section we present CaMEL, a novel Transformer-based architecture for image captioning that leverages the interaction of two interconnected language models, learning from each other during the training phase. The interplay between the two language models follows a mean teacher learning paradigm with knowledge distillation. Experimentally, we assess the effectiveness of the proposed solution on the COCO dataset and in conjunction with different visual feature extractors. When comparing with existing proposals, we demonstrate that our model provides state-of-the-art caption quality with a significantly reduced number of parameters. According to the CIDEr metric, we obtain a new state of the art on COCO when training without using external data. The source code and trained models will be made publicly available at: `https://github.com/aimagelab/camel`.

2.3.1 Introduction

Nowadays, captioning approaches have significantly evolved towards the usage of Transformer-based language models [124, 237, 65, 217, 59], while the visual feature extraction stage is rapidly evolving towards the use of grid-like features extracted with multi-modal architectures trained on large-scale data with language supervision [273, 60, 250]. In parallel, while many captioning approaches have been trained on middle-size datasets like COCO, the literature is now investigating the usage of large-scale noisy datasets as well [60, 196, 371, 328, 134].

Regardless of these architectural and structural improvements, the training methodology has remained almost unaltered. Indeed, most of the existing approaches for image captioning are based on the usage of a single language model, trained to reproduce the ground-truth caption through a cross-entropy loss and later, in a fine-tuning stage, through the REINFORCE algorithm [262, 207]. In this section, we take a different path and investigate the development of a training strategy that is based on the interplay between two distinct language models. In particular, we draw inspiration from the Mean Teacher Learning approach [294] – which has been successfully employed to learn visual representation in a self-supervised manner [35] – and propose a schema in which two language models learn and interact together at training time. One of the two language models is employed as teacher, while the other is employed as a student in a knowledge

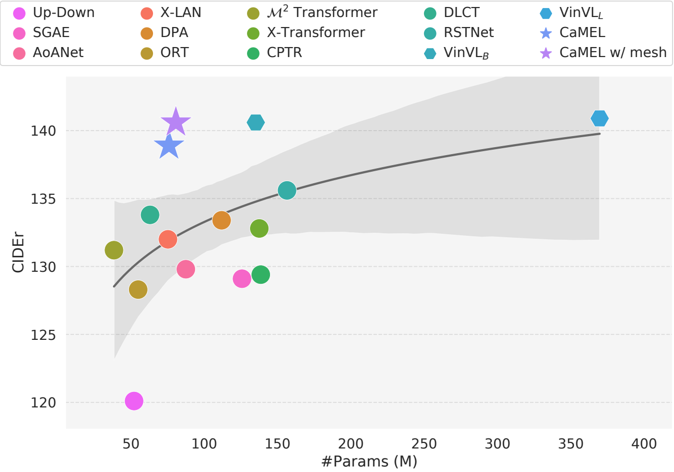


Figure 2.15: Comparison of two different versions of our approach (marked with stars) and existing approaches (marked with bullets, and hexagons if exploiting vision-and-language pre-training) in terms of number of parameters and caption quality. Our method features state-of-the-art caption quality in terms of CIDEr with a significantly reduced number of parameters.

distillation relationship [126, 12, 338]. Parameters update, on the teacher model, is carried out by averaging successive states of the student, through an exponential moving average. In this way, the teacher slowly "follows" the student state through time. We devise and compare strategies to apply this interplay paradigm in both the cross-entropy training stage and during the fine-tuning with reinforcement learning.

Noticeably, at test time one of the two language models can be discarded, so that the number of parameters is kept on pair with traditional models that employ a single language model at training time. We name our model CaMEL – short for Captioner with Mean tEacher Learning. As shown in Fig. 2.15, our model outperforms existing approaches in terms of caption quality, while being significantly less demanding in terms of number of parameters. We assess the performances of the proposed training strategy on the COCO dataset [201], employing different knowledge distillation strategies, and in comparison with other state-of-the-art approaches that have been trained on the same dataset. We also compare our results on the COCO online test server. Results demonstrate the goodness of the proposed solution, which attains a new state of the art on COCO when training without using external data.

2.3.2 Related training strategies

The training strategy for image captioning architectures usually follows the time-wise cross-entropy paradigm. This was later combined with a fine-tuning phase based on the application of the REINFORCE algorithm, to allow using as optimization objectives captioning metrics directly [262, 207], overcoming the issue of their non-differentiability and boosting the final performance. As a strategy to improve both training phases, in [140] it is proposed to exploit a teacher model trained on image attributes to generate additional supervision signals for the captioning model. These are in the form of soft-labels, which the captioning model has to align with in the cross-entropy phase, and re-weighting of the caption words to guide the fine-tuning phase. Additional improvement to the performance of recent self-attention-based image captioning approaches is due to the use of large-scale vision-and-language pre-training [60, 196, 371, 134, 381], which can be done on noisy image-text pairs, also exploiting pre-training losses different from cross-entropy, such as the masked token loss [196, 371]. Different from previous methods, our approach is based on the interplay of two different language models that are trained with the mean teacher learning paradigm and knowledge distillation, without relying on large-scale pre-training.

2.3.3 CaMEL approach

Preliminaries

Most captioning approaches rely on a single language model, which is conditioned on input images and is trained to reproduce ground-truth sentences. Formally, given a dataset of image-caption pairs $\mathcal{D} = \{(\boldsymbol{v}_i, \boldsymbol{t}_i)\}_i$, the language model aims at learning the probability distribution of the next word in a sequence, conditioned on the input image, *i.e.*

$$p(\boldsymbol{w}_\tau | \boldsymbol{w}_{k<\tau}, \boldsymbol{v}), \tag{2.16}$$

where $\boldsymbol{v}$ is an input image, τ indicates time, and $\{\boldsymbol{w}_\tau\}_\tau$ is the sequence of words comprising the generated caption. The model is trained according to a time-wise cross-entropy (XE) loss over the entire dataset, as follows:

$$\mathcal{L}(\theta) = -\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \sum_{\tau} \log p(\boldsymbol{w}_\tau | \boldsymbol{w}_{k<\tau}, \boldsymbol{v}, \theta), \tag{2.17}$$

where θ indicates the set of parameters of the model.

After a training stage with cross-entropy, sequence generation is usually fine-tuned using reinforcement learning. When training with XE, indeed, the model is trained to predict the next token given previous ground-truth words; in reinforcement learning the model is asked to generate an entire sequence and receives a reward that is proportional to the similarity of the generated caption with respect to the ground-truth. A standard practice is to employ a variant of the self-critical sequence training (SCST) approach [262] on sequences sampled using beam search [7]: to decode, the top-k words are sampled from the language model probability distribution at each timestep, and a beam with the top-k sequences with the highest probability is maintained during the generation.

Following previous works [7], the usual practice is to use the CIDEr-D score as reward, as it well correlates with human judgment [307]. In our case, we baseline the reward using the mean of the rewards [65]. The final gradient expression for the SCST training is, therefore

$$\nabla_\theta \mathcal{L}(\theta) = -\frac{1}{k} \sum_{i=1}^{k} \left((r(\boldsymbol{w}^i) - b) \nabla_\theta \log p(\boldsymbol{w}^i) \right), \tag{2.18}$$

where $\boldsymbol{w}^i$ is the i-th sentence in the beam, $r(\cdot)$ is the reward function, and $b = \left(\sum_i r(\boldsymbol{w}^i)\right)/k$ is the baseline, computed as the mean of the rewards obtained by the sampled sequences.

CaMEL Architecture

In CaMEL, instead of training a single language model, we rely on the interplay of two different language models – an *online* and a *target* language model, that interact and learn from each other during the training phase, both during the XE pre-training and during the SCST fine-tuning. At test time, each of the two language models can be used, alone, for captioning input images.

The interaction between the online and target language models at training time is two-fold. The online language model is trained, either via XE or SCST, with respect to ground-truth captions. In addition, it performs knowledge distillation with the target language model. The target language model, in turn, updates its

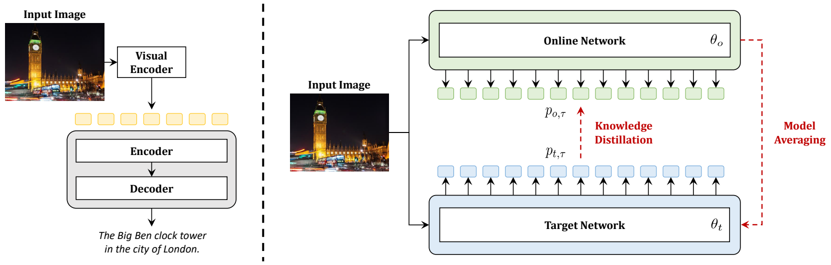


Figure 2.16: Overview of our CaMEL approach and of the interplay between the online and target language models.

weights according to an exponential moving average of the online model weights. An overview of our approach is given in Fig. 2.16.

Knowledge distillation. The target network provides regression targets to train the online network. This is done through knowledge distillation – treating the online language model as a student network and the target model as a teacher.

Given a visual input $\boldsymbol{v}$ and a conditioning partial sentence $\boldsymbol{w}$, at each timestep τ both networks provide output logits over a vocabulary of N tokens, denoted as $p_{t,\tau}$ and $p_{o,\tau}$ for the target model and the online model, respectively. Given the teacher, which is kept fixed, we learn to match these distributions by minimizing a mean squared error loss with respect to the parameters of the online network, *i.e.*

$$\min_{\theta_o} \sum_{\tau} (p_{t,\tau} - p_{o,\tau})^2, \tag{2.19}$$

where θ_o indicates the set of parameters of the online network.

Model averaging. The parameters of the target language model are updated as an exponential moving average [35] of the parameters of the online network θ_o. Formally, the parameters of the target model are given by

$$\theta_t \leftarrow \lambda\theta_t + (1 - \lambda)\theta_o, \tag{2.20}$$

where θ_t indicates the set of parameters of the target network and $\lambda \in [0, 1]$ is a target decay rate. In practice, we keep λ fixed during the entire training process. This strategy results in the target network keeping a weighted average of successive states from the online network, thus performing a form of model ensembling [294].

Algorithm 1: CaMEL PyTorch pseudocode

```
# gt, go: target and online networks
# l_KD: knowledge distillation weight
# l: network momentum
# v, c: training (image, caption) pair
for v, c in dataloader:
    t = gt(v, c)                     # target output
    o = go(v, c)                     # online output

    loss = XE(softmax(o, dim=-1), c) + l_KD * MSE(t, o)
    loss.backward()                  # backpropagate

    update(go)                       # SGD
    gt.params = l * gt.params + (1 - l) * go.params


def MSE(t, s):
    t = t.detach()                   # stop gradient
    return (t - s).square().mean()
```

Objective. During XE pre-training, the final objective we employ to train the online network is a combination of the standard XE loss, which is computed with respect to ground-truth captions, and of the knowledge distillation loss with respect to the target logits:

$$\mathcal{L} = \mathcal{L}_{XE} + \lambda_{KD} \cdot \mathcal{L}_{KD}, \tag{2.21}$$

where λ_{KD} is a weighting hyperparameter. After each SGD update of the online network, the target network is updated through the model averaging. Algorithm 1 provides the PyTorch pseudo-code of the training loop during the XE stage.

Extension to the SCST stage. The same training methodology is also applied during SCST fine-tuning. In this case, given that both language models generate a beam of k captions, there are k^2 pairs of sequences on which the MSE loss can be potentially applied. In the following, we experiment by matching the top-1 caption in each beam, according to the probability assigned by the models themselves or to the score assigned by an external image-text model. Further, we also experiment when matching each caption in the online beam with a caption in the target beam, and then applying the MSE loss on each pair.

Table 2.10: Results on the COCO Karpathy-test split with different visual encoders when training with cross-entropy loss.

				Online Network						Target Network					
	CaMEL	λ_{KD}	w/ mesh	B-1	B-4	M	R	C	S	B-1	B-4	M	R	C	S
Faster R-CNN [7, 260]		-		-	-	-	-	-	-	75.4	35.8	27.9	**56.4**	113.9	20.7
	✓	0.01		75.1	35.0	27.4	55.9	112.2	20.4	**75.8**	36.0	27.9	**56.4**	114.7	20.6
	✓	0.1		75.2	35.1	27.6	55.7	111.9	20.4	75.7	**36.1**	**28.0**	56.3	**114.8**	**20.8**
CLIP-RN50 [250]		-		-	-	-	-	-	-	75.4	35.4	27.5	56.1	112.8	20.6
	✓	0.01		74.1	34.1	27.6	55.4	110.4	20.5	75.0	35.1	27.8	56.0	112.9	20.7
	✓	0.1		75.5	35.4	27.6	56.3	113.7	20.6	**75.7**	**36.2**	**28.1**	**56.7**	**115.9**	**20.8**
CLIP-ViT-B16 [250]		-		-	-	-	-	-	-	77.2	37.1	28.7	57.5	122.0	21.7
	✓	0.01		77.0	36.8	28.6	57.5	119.6	21.5	77.5	37.9	**29.1**	**58.1**	**122.5**	**21.9**
	✓	0.1		76.6	36.3	28.3	56.9	117.9	21.2	**77.8**	**38.0**	**29.1**	58.0	**122.5**	21.8
CLIP-RN50×16 [250]		-		-	-	-	-	-	-	77.6	37.6	28.7	57.6	121.0	21.7
	✓	0.01		78.0	37.4	28.7	57.8	121.4	21.9	78.4	38.7	29.3	**58.5**	124.7	**22.3**
	✓	0.05		77.2	37.4	29.0	57.9	121.4	21.9	78.0	38.4	**29.4**	**58.5**	125.4	22.2
	✓	0.1		77.7	37.8	29.0	58.0	122.3	22.0	78.3	**39.1**	**29.4**	**58.5**	**125.7**	22.2
	✓	0.5		77.9	37.6	28.5	57.7	121.1	21.6	**78.5**	39.0	29.3	**58.5**	125.0	22.2
	✓	1.0		77.7	36.7	28.3	57.3	120.0	21.7	**78.5**	38.9	29.3	**58.5**	125.1	**22.3**
CLIP-RN50×16 [250]		-	✓	-	-	-	-	-	-	77.5	37.6	29.0	58.0	122.6	21.9
	✓	0.01	✓	77.5	37.8	29.0	58.0	123.1	21.9	78.0	**38.8**	**29.4**	**58.6**	**125.0**	**22.2**
	✓	0.05	✓	78.0	37.5	28.7	57.9	121.0	21.6	**78.2**	38.5	29.3	58.4	124.4	22.1
	✓	0.1	✓	77.3	36.8	28.5	57.3	120.1	21.7	78.0	38.5	29.3	58.4	124.2	**22.2**
	✓	0.5	✓	77.4	36.9	28.5	57.3	119.7	21.7	77.9	38.6	29.3	58.3	124.2	22.1
	✓	1.0	✓	77.0	36.7	28.4	57.2	119.7	21.5	77.7	38.3	29.3	58.2	123.7	22.0

Network architecture. CaMEL follows an encoder-decoder Transformer [306] architecture, where the encoder processes visual features via bi-directional attention and the decoder generates captions in an auto-regressive manner. Following [65, 58], our encoder incorporates additional memory slots, enhancing its ability to encode knowledge and relations learned from visual data. Specifically, we expand the set of keys and values in self-attention layers with extra and independent learnable vectors, which can encode a priori knowledge retrieved through attention. Our decoder is composed of a stack of decoder layers, each performing a right-masked self-attention and a cross-attention followed by a position-wise feed-forward network.

We also test the usage of a mesh-like connectivity between the encoder and the decoder, following [65]. In this case, the mesh mechanism further connects each encoder and decoder layer in a mesh-like structure, augmenting its ability to deal with low- or high-level features. The architecture is the same for both online and target models, while using independent parameters updated with different strategies during training.

2.3.4 Experimental evaluation

Dataset. Following the dominant paradigm in literature, we train and evaluate our model on the COCO dataset [201]. As such, we do not rely on large-scale image-text datasets [60]. COCO is composed of more than $120,000$ images, each of them associated with 5 human-collected captions. We follow the splits defined in [159], using $5,000$ images for both validation and testing and the rest for training. We also evaluate our model on the COCO online test server, which includes $40,775$ images for which annotated captions are not publicly available.

Metrics. According to the standard evaluation protocol, we employ the complete set of captioning metrics: BLEU [238], METEOR [19], ROUGE [199], CIDEr [307], and SPICE [5].

Implementation details. To represent words, we use Byte Pair Encoding (BPE) [267] with a vocabulary size of $49,408$, which is then linearly projected to the input dimensionality of the model. We use standard sinusoidal positional encodings [306] to represent word positions. All models comprise three layers in the visual encoder and three layers in the decoder, each with a dimensionality of 512, a feed-forward dimensionality of 2048, and a number of heads equal to 8. We apply dropout at the output of each sub-layer, with a dropout probability equal to 0.1. The number of memory slots is set to 40.

CaMEL: *A man hitting a tennis ball with a tennis racket.*
CaMEL w/ mesh: *A man jumping to hit a tennis ball with a tennis racket.*

CaMEL: *A vase of flowers on a table.*
CaMEL w/ mesh: *A vase of roses on a table with a bird.*

CaMEL: *A woman riding a skateboard on the street.*
CaMEL w/ mesh: *A woman riding a skateboard on the sidewalk in front of a restaurant.*

CaMEL: *A dog running with a frisbee in its mouth.*
CaMEL w/ mesh: *A brown dog carrying a purple frisbee in its mouth.*

CaMEL: *A bowl of cereal with bananas.*
CaMEL w/ mesh: *A bowl of cereal with bananas and a spoon.*

CaMEL: *A brown couch with a couch and a remote control.*
CaMEL w/ mesh: *A living room with a couch and a remote control.*

Figure 2.17: Qualitative results on sample images from the COCO test set.

In all experiments, we employ Adam [170] as optimizer and a beam size equal to 5. During pre-training with XE loss, we use a batch size of 50, following the typical Transformer learning rate scheduling strategy [306] with a warmup equal to $10,000$ iterations. During the SCST finetuning stage, we use a batch size of 30 and a fixed learning rate equal to 5×10^{-6}. The MSE loss is computed by considering only valid tokens and masking the rest. In according with [294], the target network momentum λ is set to 0.999.

Ablation Study

Visual features assessment. We firstly discuss the role of visual features, by comparing traditional detection-based features with grid-based features extracted from modern multi-modal models. In particular, we consider object detection features extracted from a Faster R-CNN model [7] pre-trained on the Visual Genome dataset [175], and grid-based features extracted from CLIP [250], which has been trained with language supervision. Since CLIP visual encoders can be either based on ViT-like or CNN-like architectures, we either employ the output of the last Transformer layer, removing the CLS token, or extract the grid of features produced immediately after the last convolutional layer.

As it can be seen from Table 2.10, Faster R-CNN features can be surpassed by modern multi-modal architectures, which brings a significant advantage in terms of all captioning metrics. While CLIP-RN50 performs worse than Faster R-CNN features when training a single language model without CaMEL and mesh-like connectivity, ViT-based and larger CNN-based models achieve better performance. The best performance is reached by the CLIP-RN50×16 variant, which employs an EfficientNet-style architecture scaling. Overall, this brings an improvement of 7.1 CIDEr points with respect to traditional detection-based features (from 113.9 to 121.0) when training a single language model without the CaMEL technique and without mesh-like connectivity.

Table 2.11: Results on the COCO Karpathy-test split with different knowledge distillation strategies during CIDEr optimization.

	B-1	B-4	M	R	C	S
CaMEL$_{\text{best}}$ – CLIP Text Embeddings	82.6	40.7	**30.3**	**60.1**	138.2	24.3
CaMEL$_{\text{all}}$ – Hungarian Matching	82.7	40.8	30.2	60.0	138.4	24.0
CaMEL$_{\text{best}}$ – Hungarian Matching	82.4	40.4	30.1	59.8	138.6	23.8
CaMEL$_{\text{all}}$	**82.9**	**41.0**	30.2	**60.1**	138.5	24.0
CaMEL$_{\text{best}}$	82.7	40.9	**30.3**	**60.1**	**138.9**	**24.5**

Role of CaMEL. We then assess the impact of the proposed training technique during the XE pre-training stage, by also testing with different weights for the knowledge distillation loss, which we indicate with λ_{KD}. Results are reported in Table 2.10. As it can be seen, CaMEL improves the performance when used with all the previously mentioned features, by a considerable margin. This confirms the appropriateness of using a mean teacher learning paradigm in image captioning. When using the RN50×16 encoder, for instance, applying CaMEL with a distillation weight of 0.1 brings an improvement of 4.7 CIDEr points (121.0 vs 125.7). Finally, the CaMEL training strategy improves the performance also when using a mesh-like connectivity, from 122.6 to 125.0 CIDEr points when using λ_{KD} equal to 0.01.

Overall, during XE pre-training CLIP features bring a 6.2% relative improvement with respect to traditional detection-based features, and CaMEL introduces, in the best feature configuration, a relative advancement of 3.9% with respect to a typical single model training. In the following experiments we employ the CLIP-RN50×16 variant as image encoder and the target network as language model.

Table 2.12: Comparison with the state of the art on the Karpathy-test split.

	B-1	B-4	M	R	C	S
Up-Down [7]	79.8	36.3	27.7	56.9	120.1	21.4
ORT [124]	80.5	38.6	28.7	58.4	128.3	22.6
GCN-LSTM [357]	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [347]	81.0	39.0	28.4	58.9	129.1	22.2
CPTR [208]	81.7	40.0	29.1	59.4	129.4	-
MT [277]	80.8	38.9	28.8	58.7	129.6	22.3
AoANet [137]	80.2	38.9	29.2	58.8	129.8	22.4
$\mathcal{M}^2$ Transformer [65]	80.8	39.1	29.2	58.6	131.2	22.6
X-LAN [237]	80.8	39.5	29.5	59.2	132.0	23.4
TCTS [140]	81.2	40.1	29.5	59.3	132.3	23.5
X-Transformer [237]	80.9	39.7	29.5	59.1	132.8	23.4
DPA [203]	80.3	40.5	29.6	59.2	133.4	23.3
DLCT [217]	81.4	39.8	29.5	59.1	133.8	23.0
RSTNet [375]	81.8	40.1	29.8	59.5	135.6	23.3
CaMEL	82.7	40.9	**30.3**	**60.1**	138.9	**24.5**
CaMEL w/ mesh	**82.8**	**41.3**	30.2	**60.1**	**140.6**	23.9
VinVL$_B$ [371]	*82.0*	*40.9*	*30.9*	*60.7*	*140.6*	*25.1*
VinVL$_L$ [371]	*82.0*	*41.0*	*31.1*	*60.9*	*140.9*	*25.2*

Table 2.13: Leaderboard of various methods on the online COCO test server. The † marker indicates ensemble configurations.

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down [7]†	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5
SGAE [347]†	81.0	95.3	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
TCTS [140]	80.5	94.8	65.3	89.1	50.9	80.5	39.0	70.3	29.0	38.4	58.9	74.0	125.3	127.2
CPTR [208]	81.8	95.0	66.5	89.4	51.8	80.9	39.5	70.8	29.1	38.3	59.2	74.4	125.4	127.3
AoANet [137]†	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
$\mathcal{M}^2$ Transformer [65]†	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
X-Transformer [237]†	81.9	95.7	66.9	90.5	52.4	82.5	40.3	72.4	29.6	39.2	59.5	75.0	131.1	133.5
DPA [203]†	81.8	96.3	66.5	91.2	51.9	83.2	39.8	73.3	29.6	39.3	59.4	75.1	130.4	133.7
RSTNet [375]†	82.1	96.4	67.0	91.3	52.2	83.0	40.0	73.1	29.6	39.1	59.5	74.6	131.9	134.0
DLCT [217]†	82.4	96.6	67.4	91.7	52.8	83.8	40.6	74.0	29.8	39.6	59.8	75.3	133.3	135.4
CaMEL	82.2	96.6	67.2	91.7	52.5	83.8	40.2	73.7	30.0	39.6	59.6	75.2	133.7	136.4
CaMEL w/ mesh	82.6	96.8	67.5	91.9	52.8	83.9	40.5	73.8	29.9	39.4	59.8	74.9	135.1	137.7
CaMEL w/ mesh†	**83.2**	**97.3**	**68.3**	**92.7**	**53.6**	**84.8**	**41.2**	**74.9**	**30.2**	**39.7**	**60.2**	**75.6**	**137.5**	**140.0**

Evaluation of different SCST strategies. Turning to the evaluation of the SCST fine-tuning stage, in Table 2.11 we compare the performance of different knowledge distillation strategies. In particular, we experiment the CaMEL$_{\text{best}}$ version, in which we pair the two logits sequences with the highest probability from both models to compute the KD loss, and the CaMEL$_{\text{all}}$ version where we use all the sequences generated by the beam search algorithm, pairing them in sorted order of log probability. Further, we explore the use of CLIP text embeddings in the MSE loss. In CaMEL$_{\text{best}}$ with CLIP text embeddings, we select the most probable caption in each of the two beams, and then apply the MSE loss between the CLS tokens given by the CLIP text encoder.

Moreover, we investigate the usage of the Hungarian matching algorithm [177] to couple captions in the online and target beams, using their CLIP embedding similarity as distance function. We then compute the MSE loss on the original logits between all pairs – in CaMEL$_{\text{all}}$ version – or only the most probable caption from the target model and its most similar one from the online model – in CaMEL$_{\text{best}}$ version. The best version, based on all metrics except BLEU scores, is CaMEL$_{\text{best}}$ without additional algorithms, while for the BLEU metrics the best one is CaMEL$_{\text{all}}$ always without the use of additional algorithms to pair captions.

Comparison with the state of the art

We compare the results of CaMEL with those of several recent image captioning models trained without large-scale vision-and-language pre-training. In our analysis, we include methods with LSTM-based language models and attention over image regions such as Up-Down [7], either enhanced with graph-based encoding (*i.e.* GCN-LSTM [357], SGAE [347], and MT [277]) or self-attention (*i.e.* AoANet [137], X-LAN [237], DPA [203], and TCTS [140]), and captioning architectures entirely based on the Transformer network such as ORT [124], $\mathcal{M}^2$ Transformer [65], X-Transformer [237], CPTR [208], DLCT [217], and RSTNet [375].

Performance on COCO. As it can be observed from Table 2.12, our proposal reaches 138.9 CIDEr points, beating all the compared approaches. Adding the mesh-like connectivity to the decoder further improves the results to 140.6 CIDEr points. This represents an increase of 5.0 CIDEr points with respect to the current state-of-the-art when training on the COCO dataset only [375]. Further, in Fig. 2.15 we compare the aforementioned approaches in terms of both CIDEr and number of parameters. As it can be noticed, not only CaMEL reports state-of-the-art results in terms of caption quality, but it also features a significant reduction in terms of number of trainable weights. As most of previous literature has increased caption quality by increasing the model capacity, our approach represents an outlier in this trend, and demonstrates that state-of-the-art CIDEr levels can be obtained even with a lightweight model.

Finally, in Table 2.12 and Fig. 2.15 we also report the performance obtained by a recent approach which employs large-scale pre-training on external data, *i.e.* VinVL [371]. While this approach is not directly comparable with CaMEL considering that it employs more data, we notice that our proposal is on pair with the Base version of VinVL, and only 0.3 CIDEr points below its Large version. In terms of model size and number of parameters, VinVL is also extremely more demanding than our proposal (cfr. Fig. 2.15). This further confirms that the usage of proper visual features and of mean teacher learning strategy can achieve a good caption quality with a reduced model size.

Online evaluation. Finally, we also report the performance of our method on the online COCO test server[8]. In this case, we also employ an ensemble of four models trained with the mesh-like connectivity. The evaluation is done on the COCO test split, for which ground-truth annotations are not publicly available. Results are reported in Table 2.13, in comparison with the top-performing approaches of the leaderboard. As it can be seen, our method surpasses the current state of the art on all metrics, achieving an advancement of 4.2 CIDEr points with respect to the best performer.

[8] `https://competitions.codalab.org/competitions/3221`

2.4 Novel object captioning

Image captioning models have lately shown impressive results when applied to standard datasets. Switching to real-life scenarios, however, constitutes a challenge due to the larger variety of visual concepts which are not covered in existing training sets. For this reason, novel object captioning (NOC) has recently emerged as a paradigm to test captioning models on objects which are unseen during the training phase. In this section, we present a novel approach for NOC that learns to select the most relevant objects of an image, regardless of their adherence to the training set, and to constrain the generative process of a language model accordingly. Our architecture is fully-attentive and end-to-end trainable, also when incorporating constraints. We perform experiments on the *held-out* COCO dataset, where we demonstrate improvements over the state of the art, both in terms of adaptability to novel objects and caption quality.

2.4.1 Introduction

Describing images has emerged as an important task at the intersection of computer vision, natural language processing, and multimedia, thanks to the key role it can have to empower both retrieval and multimedia systems [159, 7, 61, 62, 64, 25, 285]. Recent advances in image captioning, indeed, have demonstrated that fully-attentive architectures can provide high-quality image descriptions when tested on the same data distribution they are trained [124, 65, 237, 196]. As existing datasets for image captioning [364, 201] are limited in terms of the number of visual concepts they contain, though, the application of such systems in real-life scenarios is still challenging. For this reason, the task of Novel Object Captioning (NOC) has recently gained a lot of attention due to its affinity towards real-world applications [123, 2, 135]. This setting, indeed, requires a model to describe images containing objects unseen in the training image-text data, also referred to as out-of-domain visual concepts.

Since the language model behind a NOC algorithm can not be trained to predict out-domain words, proper incorporation of such novel words during the generation phase is one of the most relevant issues in this task. Early NOC approaches [123, 309] tried to transfer knowledge from out-domain images by conditioning the model at training time on external unpaired visual and textual data. Further works [356, 197] proposed to integrate coping mechanisms in the language model to select words corresponding to the predictions of a tagger. However, these frameworks do not include a proper and explainable method to identify which

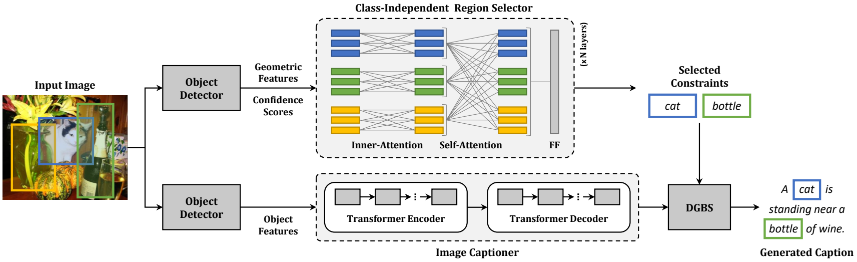


Figure 2.18: Summary of our approach.

objects on the scene are more relevant to be described, and consequently, lack on leveraging all the available information provided by visual inputs. On a different line, Anderson *et al.* [6] devised a Constrained Beam Search algorithm to force the inclusion of selected tag words in the output caption, following the predictions of a tagger.

Inspired by this last line of research, we combine the ability to constrain the predictions from a language model with the usage of object regions and of fully-attentive architectures, which is dominant in traditional image captioning. Precisely, we devise a model with a specific ability to select objects in the scene to be described, with a class-independent module that can work on both in-domain and out-of-domain objects. Further, we combine this with a variant of the Beam Search algorithm which can include constraints produced by the region selector, while assuring end-to-end differentiability. We provide extensive experiments to validate the proposed approach: when tested on the *held-out* portion of the COCO dataset, our model provides state-of-the-art results in terms of caption quality and adaptability to describe objects unseen in the training set. Given its simplicity and effectiveness, our approach can also be thought of as a powerful new baseline for NOC, which can foster future works in the same area.

2.4.2 Proposed method

Our NOC approach can be conceptually divided into two modules: an *image captioner* and a *region selector*. While the image captioning model is conditioned on the input image and is in charge of modeling a sequence of output words, the region selector is in charge of choosing the most relevant objects which need to be described, regardless of their adherence to the training set. The objects picked by the selector are used as constraints during the generation process, so that the output caption is forced to contain their labels as predicted by an object detector. All the components of our architecture are based on fully-attentive structure, and end-to-end training is allowed also when adding constraints to the language model. Fig. 2.18 shows an outline of the approach.

Class-Independent Region Selector

The role of the region selector is to identify objects which must be described in the output sentence. Since the object selector will need to work on classes that are unseen in the training set, we adopt a class-independent strategy in which no information about the object class is employed in the feature extraction process. Instead, we model intra-class relationships between objects of the same class, to handle the case in which multiple objects of the same class are present on the scene.

Given a set of regions $\boldsymbol{X} = \{x_i\}_i$ extracted from the input image, along with their classes $\{c_i\}_i$, we extract central coordinates, width, height and, additionally, we compute the object area. We also consider as an extra feature the confidence score s_i of the object, to obtain a class-independent feature vector:

$$x_i = \left[\left(\frac{x_c}{W} \right), \left(\frac{y_c}{H} \right), \left(\frac{w_i}{W} \right), \left(\frac{h_i}{H} \right), \left(\frac{w_i \cdot h_i}{W \cdot H} \right), s_i \right] \tag{2.22}$$

where x_c and y_c are the coordinates of the center of the region, w_i and h_i its width and height, and W and H the image dimensions.

The set of feature vectors obtained for an image is then fed to a sequence of Transformer-like [306] layers, each of them composed by an *inner-attention* operator and a *self-attention* operator. The inner-attention operator is devised to connect together regions belonging to the same class, while the self-attention operator provides complete connectivity between elements in $\boldsymbol{X}$. The combination of these two operators allows the region selector to independently focus on specific clusters of objects, in order to exchange semantically related information and learn intra-class dependencies, and then, to model long-range and diverse dependencies.

Table 2.14: Evaluation on the *held-out* COCO test set, when using different constraint selection approaches.

	Cross-Entropy Loss							CIDEr Optimization							CIDEr Optimization with DGBS						
	In-Domain			Out-Domain				In-Domain			Out-Domain				In-Domain			Out-Domain			
	M	C	S	M	C	S	F1	M	C	S	M	C	S	F1	M	C	S	M	C	S	F1
No Constraints	27.2	108.9	20.2	22.4	68.5	14.7	0.0	28.4	122.3	22.3	23.5	76.8	16.3	0.0	28.1	120.9	21.9	23.4	76.5	16.1	0.0
Top-1	26.2	97.4	19.2	24.1	75.9	17.6	60.1	27.6	110.5	21.3	25.4	84.6	18.8	60.2	27.9	115.9	21.0	25.3	84.7	18.7	60.2
Top-2	24.4	81.9	16.4	23.8	68.7	16.1	68.1	26.2	95.4	18.4	25.1	77.6	17.3	68.1	27.1	102.9	18.4	25.6	80.0	17.2	68.1
Top-3	22.7	69.9	14.4	22.4	56.9	14.5	66.0	25.1	83.3	16.5	24.4	67.1	15.4	66.0	26.6	92.3	17.0	25.2	70.8	15.6	66.0
Region Selector (*w/o Inner*)	25.2	70.6	17.7	24.1	70.6	16.8	70.2	26.8	101.5	19.6	25.6	80.5	18.0	70.2	27.4	108.0	19.7	25.8	82.2	18.2	70.2
Region Selector	26.2	97.0	19.2	**24.9**	**78.2**	**18.3**	**75.0**	27.6	109.2	21.1	**26.1**	**87.7**	**19.4**	**75.0**	27.9	115.3	21.0	**26.3**	**88.5**	**19.4**	**75.1**
Oracle Constraints	*27.3*	*107.0*	*20.6*	*25.6*	*84.0*	*19.0*	*76.0*	*28.5*	*118.9*	*22.5*	*26.6*	*91.7*	*20.2*	*76.0*	*28.6*	*122.9*	*22.3*	*26.6*	*92.3*	*20.2*	*76.0*

Given a partition of $\boldsymbol{X}$ computed according to the class each region belongs to, *i.e.* $\{\boldsymbol{r}_c \subseteq \boldsymbol{X} \mid \forall x_i, x_j \in \boldsymbol{r}_c, c_i = c_j\}_c$, the result of the inner-attention operator applied over an element of the partition is a new set of elements $\mathcal{I}(\boldsymbol{r}_c)$, with the same cardinality as $\boldsymbol{r}_c$, in which each element is replaced with a weighted sum of values computed from regions of the same class. Formally, it can be defined as:

$$\mathcal{I}(\boldsymbol{r}_c) = \mathsf{Attention}(W_q\boldsymbol{r}_c, W_k\boldsymbol{r}_c, W_v\boldsymbol{r}_c), \tag{2.23}$$

where $\boldsymbol{r}_c$ is the set of all elements of $\boldsymbol{X}$ belonging to class c, W_* are learnable projection matrices, and Attention indicates the standard dot-product attention [306].

The inner attention layer is applied independently over each element of the above-defined partition so that the overall encoding of $\boldsymbol{X}$ is a new sequence of elements defined as follows:

$$\mathcal{I}(\boldsymbol{X}) = \left(\mathcal{I}(\boldsymbol{r}_1), \mathcal{I}(\boldsymbol{r}_2), ..., \mathcal{I}(\boldsymbol{r}_C)\right), \tag{2.24}$$

where C indicates the number of classes. After each inner-attention layer, a self-attention layer is employed to connect elements of different classes together. Formally, it is defined as:

$$\mathcal{S}(\boldsymbol{X}) = \mathsf{Attention}(W_q\boldsymbol{X}, W_k\boldsymbol{X}, W_v\boldsymbol{X}), \tag{2.25}$$

where W_* are, again, learnable projection matrices.

After a sequence of inner- and self-attention layers, in which each pair of operators is followed by a position-wise feed-forward network [306], the region selector outputs a selection score Y_i for each object proposal. To do so, we apply an affine transformation and a non-linear activation to the output of the last layer:

$$Y_i = \sigma\left(\mathsf{RegionSelector}(X_i)W_o\right), \tag{2.26}$$

where $W_o \in \mathbf{R}^{d\times 1}$ are learnable weights and σ is a sigmoid.

Training. The region selector is trained using a binary cross-entropy loss. To build ground-truth data, for each image we collect the object classes identified by the object detector and construct a binary ground-truth vector indicating whether a class name is contained in at least one of the ground-truth captions associated with the image. We also consider as positives synonyms and plural forms of the object class names. At inference time, we extract the selected objects for each image adopting 0.5 as threshold.

Image Captioner

After object selection, our image captioning model is responsible for generating a caption using the chosen class names as constraints. Inspired by recent works which employ fully-attentive models in image captioning [124, 65, 217], we create a captioning model with an encoder-decoder structure, where the encoder refines image region features and the decoder generates captions auto-regressively.

Encoder. Recent captioning literature has shown that object regions are the leading solution to encode visual inputs [7, 357, 347], followed by self-attentive layers to model region relationships [124, 137, 65, 237, 286, 217]. However, as self-attention can only encode pairwise similarities, it exhibits a significant limitation on encoding knowledge learned from data. To overcome this restraint, we enrich our encoder with memory slots [58, 65]. Specifically, we extend the set of keys and values of self-attention layers with additional learnable vectors, which are independent of the input sequence and can encode a priori information retrieved through attention.

Decoder. The decoder is the actual language model, conditioned on both previously generated words and image region encodings. As in the standard Transformer [306], our language model is composed of a stack of decoder layers, each performing a masked self-attention and a cross-attention followed by a position-wise feed-forward network. Specifically, for each cross-attention, keys and values are inferred from the encoder output, while for the masked self-attention, queries, keys, and values are exclusively extracted from the input sequence of the decoder. This self-attention is right-masked so that each query can only attend to keys obtained from previous words.

Including Lexical Constraints

To include the lexical constraints produced by the region selector when decoding from the language model, we devise a variant of the Beam Search algorithm [234, 130] which supports the adoption of single-word constraints. Given a number of word constraints $\boldsymbol{W} = \{w_0, w_1, ..., w_n\}$ and a maximum decoding length T, we frame the decoding process in a matrix $\boldsymbol{G}$ with n rows and T columns, where the horizontal axis covers the time steps in the output sequence, and the vertical axis indicates the constraints coverage. Each cell of the matrix can contain a beam of partially decoded sequences.

At iteration t, each row i of $\boldsymbol{G}[:, t]$ can be filled in two ways: either by continuing the beam contained in $\boldsymbol{G}[i, t-1]$ by sampling from the probability

Algorithm 2: Grid Beam Search

```
$\boldsymbol{G} \leftarrow$ initGrid($n, T, k$)
for $t = 1; t < T; t++$ do
    for $c = \max(0, n + t - T); c < \min(t, n); c++$ do
        $g, s = \emptyset$ forall *hyp in* $\boldsymbol{G}[c, t-1]$ do
            $g \leftarrow g \cup$ model.step($hyp$)
        end
        if $c > 0$ then
            forall *hyp in* $\boldsymbol{G}[c-1, t-1]$ do
                $s \leftarrow s \cup$ model.add_constr($hyp, \{w_0, ..., w_n\}$)
            end
        end
        $\boldsymbol{G}[c, t] \leftarrow \underset{h \in g \cup s}{}$ (model.score($h$))
    end
end
$topHyp \leftarrow$ hasEOS($\boldsymbol{G}[n, :]$)Remove sequences w/o EOS
return $\underset{h \in topHyp}{\arg\max}$ (model.score($h$))
```

distribution of the language model, or by forcing the inclusion of a constraint from $\boldsymbol{W}$. In the former case, the resulting updated beam of sequences is stored in $\boldsymbol{G}[i, t]$, while in the latter case it is stored in $\boldsymbol{G}[i + 1, t]$. At the end of the generation process, the last row of $\boldsymbol{G}$ will contain sequences that satisfy all constraints.

Algorithm 2 reports the pseudo-code of our constrained beam search procedure. There, k indicates the number of elements in each bin, model.step indicates sampling from the language model probability distribution to continue the generation of a partially-decoded sequence, while model.add_constr indicates a function which continues a beam by adding all the available constraints, excluding those which have already been generated for a sequence. Because all the operations required to include constraints are differentiable, we call our constraint inclusion approach *Differentiable Grid Beam Search* (DGBS), and employ it to fine-tune the image captioner also when using a CIDEr-D optimization strategy.

Table 2.15: Comparison with the state of the art on the *held-out* COCO test set.

	F1 Scores									Captioning Metrics				
	$F1_{\text{bottle}}$	$F1_{\text{bus}}$	$F1_{\text{couch}}$	$F1_{\text{microwave}}$	$F1_{\text{pizza}}$	$F1_{\text{racket}}$	$F1_{\text{suitcase}}$	$F1_{\text{zebra}}$	$F1_{\text{average}}$	B-4	M	R	C	S
DCC [123]	4.6	29.8	45.9	28.1	64.6	52.2	13.2	79.9	39.8	-	21.0	-	59.1	13.4
NOC [309]	17.8	68.8	25.6	24.7	69.3	55.3	39.9	89.0	48.8	-	21.3	-	-	-
NBT [215]	14.0	74.8	42.8	63.7	74.4	19.0	44.5	92.0	53.2	-	23.9	-	84.0	16.6
CBS [6]	16.3	67.8	48.2	29.7	77.2	57.1	49.9	85.7	54.0	-	23.6	-	77.6	15.9
LSTM-C [356]	29.7	74.4	38.8	27.8	68.2	70.3	44.8	91.4	55.7	-	23.0	-	-	-
DNOC [336]	33.0	76.9	54.0	46.6	75.8	33.0	59.5	84.6	57.9	-	21.6	-	-	-
LSTM-P [197]	28.7	75.5	47.1	51.5	81.9	47.1	62.6	93.0	60.9	-	23.4	-	88.3	16.6
NBT + CBS [215]	38.3	80.0	54.0	**70.3**	81.1	74.8	67.8	**96.6**	70.3	-	24.1	-	86.0	17.4
Top-2	29.6	77.4	44.7	62.6	83.3	**81.2**	70.7	95.1	68.1	28.1	25.6	52.7	80.0	17.2
Region Selector (*w/o Inner*)	42.3	78.3	54.4	59.4	85.3	79.1	67.2	95.6	70.2	28.4	25.8	52.8	82.2	18.2
Region Selector	**43.9**	**83.7**	**66.8**	64.7	**88.0**	81.0	**76.9**	95.4	**75.1**	**30.3**	**26.3**	**53.8**	**88.5**	**19.4**

Table 2.16: Region selector performance evaluation using different loss weights for zero and one values.

			In-Domain			Out-Domain			
	λ_0	λ_1	M	C	S	M	C	S	F1
Region Selector (*w/o Inner*)	0.4	0.6	27.4	111.9	20.3	25.8	85.6	18.7	68.5
Region Selector	0.4	0.6	**28.1**	**119.2**	**21.3**	**26.0**	**89.0**	**19.4**	**70.4**
Region Selector (*w/o Inner*)	0.3	0.7	27.2	108.7	19.9	25.9	84.9	18.6	69.9
Region Selector	0.3	0.7	**28.0**	**116.4**	**21.2**	**26.2**	**88.7**	**19.4**	**74.2**
Region Selector (*w/o Inner*)	0.2	0.8	27.4	108.0	19.7	25.8	82.1	18.2	70.2
Region Selector	0.2	0.8	**27.9**	**115.3**	**21.0**	**26.3**	**88.5**	**19.4**	**75.1**
Region Selector (*w/o Inner*)	0.1	0.9	26.9	97.8	18.2	25.6	73.2	16.7	67.1
Region Selector	0.1	0.9	**27.9**	**114.3**	**20.8**	**26.2**	**87.5**	**19.2**	**75.6**

2.4.3 Experimental evaluation

Evaluation Protocol

Dataset. We conduct experiments on the *held-out* COCO dataset [123], which consists of a subset of the COCO dataset [201] for standard image captioning, where the training set excludes all image-caption pairs that mention at least one of the following eight objects: *bottle*, *bus*, *couch*, *microwave*, *pizza*, *racket*, *suitcase*, and *zebra*. We follow the splits defined in [123] and take half of COCO validation set for validation and the other half for testing.

Metrics. To evaluate caption quality, we use standard captioning metrics (*i.e.* BLEU-4 [238], METEOR [19], ROUGE [199], CIDEr [307], and SPICE [5]), while we employ F1-scores [123] to measure the model ability to incorporate new objects in generated captions.

Implementation details. To extract geometric features and confidence scores for our region selector, we employ Faster R-CNN [260] with ResNet-50-FPN backbone, trained on COCO [201]. For both training and inference, we discard the detections of the *person* and *background* classes. During training, we use different loss weights (*i.e.*, $\lambda_0 = 0.2$ and $\lambda_1 = 0.8$) to balance the importance of zero and one ground-truth values, and we limit the number of object proposals for each image to 10 according to their confidence scores. Region selector features are projected to a 128-dimensional embedding space and passed through $N = 2$ identical layers, each composed of inner-attention, self-attention, and feed-forward.

For our image captioning model, we extract object features from Faster R-

CNN [260] with ResNet-101 finetuned on Visual Genome [175, 7]. Following [65], we use three layers for both encoder and decoder and employ 40 memory vectors for each encoder layer. We represent words with GloVe word embeddings [242], using two fully-connected layers to convert between the GloVe dimensionality (*i.e.*, 300) and the captioning model dimensionality (*i.e.*, 512) before the first decoding layer and after the last decoding layer. Before the final softmax, we multiply with the transpose of the word embeddings. We pre-train our captioning model using cross-entropy and finetune it using RL with CIDEr-D reward. During this phase, we use the classes detected by Faster R-CNN, trained on COCO, that are mentioned in the ground-truth captions as constraints for our DGBS algorithm. We limit the number of possible constraints to 5.

All experiments are performed with a batch size equal to 50. For training the region selector and pre-training the captioning model, we use the learning rate scheduling strategy of [306] with a warmup equal to $10,000$ iterations and Adam [170] as optimizer. CIDEr-D optimization is done with a learning rate equal to 5×10^{-6}.

Experimental Results

Table 2.14 shows the results of our model in terms of captioning metrics and F1-score averaged over the eight held-out classes, using different strategies to train the captioning model. We compare with a variant of our region selector without inner-attention (*i.e.*, *w/o Inner*) and using the top-k detections, with $k = 1, 2, 3$, instead of our selection strategy. For reference, we also report the performance when using oracle constraints coming from ground-truth captions. As it can be seen, our solution achieves the best results in terms of both caption quality and F1-score, demonstrating the effectiveness of our region selector for choosing constraints for the captioning model and the importance of the inner-attention operator. Furthermore, by comparing the results with standard CIDEr optimization and those obtained using our DGBS algorithm during training, we can see improved results on both in-domain and out-domain captions, thus confirming the usefulness of our training strategy.

In Table 2.16, we show the results when using different weights to balance the importance of zero and one ground-truth values. As it can be seen, our complete region selector achieves better performance than the variant without inner-attention, thus further demonstrating the effectiveness of the proposed attention operator. Additionally, employing $\lambda_0 = 0.2$ and $\lambda_1 = 0.8$ provides the best balance in terms of captioning metrics and F1-score.

Finally, in Table 2.15, we compare our model with NOC state-of-the-art approaches. As it can be noticed, our region selector obtains the best results in terms of both F1-scores and captioning metrics, achieving a new state of the art on the *held-out* COCO dataset.

Chapter 3

Cross-modal retrieval

As shown in the previous chapters, recent progress in computer vision and natural language processing has enabled the blooming of neural networks able to narrow the gap between vision and language generating new solutions not only for image and video captioning, but also for cross-modal retrieval [172, 85, 186, 191, 227], visual question answering [333, 7, 268], and vision-and-language navigation [8, 324, 93]. In this chapter, we aim our attention on cross-modal retrieval and on the development of deep learning architectures capable of retrieving visual items given textual queries and vice versa.

Contributions

The leading design of many cross-modal retrieval methods has been that of learning a joint multimodal embedding space in which text and images could be projected and compared. In the first part of this chapter, we propose an attention-based aggregation function that aggregates elements of a sequence or a set in order to obtain a single response, like a classification or a similarity score, and we use these responses as projections for the two modalities. We also demonstrate the effectiveness of this solution for visual question answering, applying these outputs for classification purposes. Experimentally, we show that our approach increases performances on both tasks.

This chapter is related to publications [2, 3, 5, 9] reported in Appendix A, by the author of the thesis. See Appendix A for details.

While most of the state-of-the-art solutions have shown impressive results using large-scale models trained with a vast amount of training data, their application to more real-world scenarios has been rarely investigated due to their inference time and model size. Therefore, in the second part of this chapter, we go beyond these limitations and tackle the design of visual-semantic architecture that fills the gap between effectiveness and efficiency. Our model, called ALADIN, first generates scores by aligning at fine-grained level images and texts, and then it learns a shared embedding space by distilling these scores, allowing for a more efficient kNN search. We demonstrate that our approach can compete with state-of-the-art large models while being almost 90 times faster.

Finally, while all of these solutions as proved impressive achievements on fully-supervised settings in which a large amount of training data is available, their application to more challenging scenarios has been rarely investigated. In the last part of this chapter, we go beyond these limitations and tackle the design of visual-semantic algorithms in the domain of the digital humanities and cultural heritage. To this end, we collect and annotate the Artpedia dataset that contains paintings and textual sentences describing both the visual content of the paintings and other contextual information, and we devise a model that matches images and texts but also identifies if a sentence is visual or contextual. Moreover, since this setting also features a significant lack of training data, making the use of fully-supervised approaches infeasible, we propose cross-modal retrieval solutions that can automatically align illustrations and textual elements without paired supervision. Our approach transfers the knowledge learned on ordinary visual-semantic datasets to the artistic domain. Experimentally, we validate the proposed strategies and quantify the domain shift between natural images and artworks on two datasets specifically designed for this domain.

3.1 Related work

The key issue of cross-modal retrieval methods is to measure the visual-semantic similarity between images and textual sentences. Typically, this is achieved by learning a common embedding space where visual and textual data can be projected and compared. One of the first attempt in this direction has been made by Kiros *et al.* [172] in which a triplet ranking loss is used to maximize the distance between mismatching items and minimize that between matching pairs.

Following this line of work, Faghri *et al.* [85] introduced a simple modification of standard loss functions, based on the use of hard negatives during training, that has been demonstrated to be effective in improving the final performance and widely adopted by several subsequent methods [83, 105, 186, 285]. Among them, Gu *et al.* [105] further improved the visual-semantic embedding representations by incorporating generative processes of images and text. Differently, Engilberge *et al.* [83] proposed a novel approach in which spatial pooling mechanisms are used to embed visual features and localize new concepts from the embedding space. Later, strong improvements have been obtained with the stacked cross-attention mechanism proposed by Lee *et al.* [186] in which a latent correspondence between image regions and words of the caption is learned to match images and textual sentences. Wang *et al.* [325] extended this paradigm by adding the relative position of image regions in the visual encoder, demonstrating better performance. On a similar line, Li *et al.* [191] introduced a visual-semantic reasoning model based on graph convolutional networks that can generate better visual representations and capture key objects and semantic concepts present on a scene.

Many works followed [227, 229, 191, 286, 249, 330], trying out BERT [78] as a text extractor other than a simple GRU and showing the effectiveness of region-based features [7] as visual representation. After the success of BERT-like models in Natural Language Processing [78, 188, 210], many works tried to employ the Transformer Encoder to jointly process images and text, like VilBERT [212], OSCAR [196], VL-BERT [288], or VinVL [371]. These methods tackle image-text matching as a binary classification problem, where an (image, sentence) pair is input to the complex Transformer architecture which is trained to predict the probability that the sentence relates to the image.

3.2 Attention-based aggregation function

As both images and text can be encoded as sets or sequences of elements – like regions and words – proper reduction functions are needed to transform a set of encoded elements into a single response, like a classification or similarity score. In this section, we propose a novel fully-attentive reduction method for vision and language. Specifically, our approach computes a set of scores for each element of each modality employing a novel variant of cross-attention, and performs a learnable and cross-modal reduction, which can be used for both classification and ranking. We test our approach on image-text matching and visual question answering, building fair comparisons with other reduction choices, on both COCO and VQA 2.0 datasets. Experimentally, we demonstrate that our approach leads to a performance increase on both tasks. Further, we conduct ablation studies to validate the role of each component of the approach.

3.2.1 Introduction

As humans we learn to combine vision and language early in life, building connections between visual stimuli and our ability to communicate in a common natural language. The abundance and diversity of daily-created data pose new unparalleled opportunities in the attempt to artificially reproduce this joint visual-semantic understanding. Recent progress at the intersection of Computer Vision and Natural Language Processing has led to new architectures capable of automatically combining the two modalities, improving the performance of different vision-and-language tasks, such as image captioning [7, 314, 58, 65], cross-modal retrieval [172, 20, 85, 186, 64], and visual question answering [9, 7, 293]. All these settings have usually been addressed by using recurrent neural networks that can naturally model the sequential nature of textual data. However, the recent advent of fully attentive mechanisms, in which the recurrent relation is abandoned in favor of the use of self- and cross-attention, has consistently changed the way to deal with visual and textual data, as testified by the success and performance improvements obtained with the Transformer [306] and BERT [78] models.

Nonetheless, the difficulty in tackling these problems is still given by the huge discrepancy between visual-semantic modalities. In this context, recent research efforts have mainly focused on treating images and text as sets or sequences of building elements, such as image regions and sentence words, leading to a better content understanding of both modalities [7, 186]. While this approach has allowed more fine-grained alignment and richer representation capabilities of visual-semantic concepts, it has also caused a large increase of features that need to be combined together without loosing inter- and intra-modality interactions.

As such, aggregating features represents one of the crucial steps in visual-semantic tasks in which different information are fused together to obtain a compact and comprehensive representation of both modalities. In this section, we tackle the problem of aggregating visual-semantic features in an effective and learnable way, and propose a novel aggregation function based on attentive mechanisms that can be successfully applied to different vision-and-language tasks. Our method can be seen as a variant of the cross-attention schema in which a set of scores are learned to aggregate feature vectors coming from image regions and textual words, thus taking into account the cross-modality interactions between elements (Fig. 3.1).

We apply our attention-based aggregation function to cross-modal retrieval

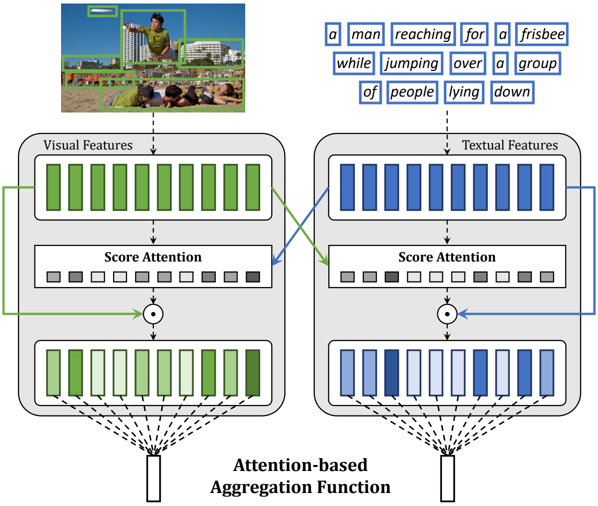


Figure 3.1: We propose a novel aggregation function for vision-and-language tasks. Given sets of visual and textual inputs, our approach computes a set of *scores* for each modality, using a novel operator based on cross-attention which ensures a learnable reduction based on cross-modal information flow.

and visual question answering: in the first case, the compact representation of visual-semantic data is used to measure the similarity between the input image and the textual sentence, while, in the visual question answering task, it is used to compute a classification score over a set of possible answers for each image-question pair. Experimentally, we test our approach on the COCO dataset for cross-modal retrieval and on the VQA 2.0 dataset for visual question answering, and we demonstrate its effectiveness in both settings with respect to different commonly used aggregation functions.

To summarize, our main contributions are as follows:

- We introduce a new aggregation method based on attentive mechanisms that learns a compact representation of sets or sequences of feature vectors.
- We tailor our method to combine vision-and-language data in order to obtain a cross-modal reduction for both classification and ranking objectives. Also, our method can be easily adapted to other tasks requiring an aggregation of elements with minimum changes in the architecture design.

- We show the effectiveness of our solution when compared to other common reduction operators, demonstrating superior performance in aggregating multi-modal features.

3.2.2 Related research efforts

Visual Question Answering

Many different solutions have been proposed to address the VQA task, ranging from Bayesian [220] and compositional [9, 133] approaches to spatial attention-based methods [354, 7] and bilinear pooling schemes [169]. In the last few years, the use of attention mechanisms has become the leading choice for this task, resulting in new models in which relevance scores over visual and textual features are computed to process only relevant information. Among them, Anderson *et al.* [7] re-visited the standard attention over a spatial grid of features and proposed to encode images with multiple feature vectors coming from a pre-trained object detector.

After this work, several methods with attention over image regions have been presented [169, 31, 98, 293, 192]. While Cadene *et al.* [31] proposed a reasoning module to encode the semantic interaction between each visual region and the question, Gao *et al.* [98] introduced a dynamic fusion framework that integrates inter- and intra-modality information. Differently, Li *et al.* [192] presented a novel solution based on graph attention networks that considers spatial and semantic relations to enrich image representations.

Following the advent of fully-attentive mechanisms for sequence modeling tasks like machine translation and language understanding [306, 78], different Transformer-based solutions have also been proposed to address multimodal settings [293, 99, 65]. In the context of visual question answering, Yu *et al.* [365] presented a co-attention module made of a stack of attentive layers based on self-attention, keeping the textual encoder based on recurrent neural networks. Gao *et al.* [99], instead, introduced a novel architecture entirely based on fully-attentive mechanisms that learns cross-modality relationships between latent summarizations of visual regions and questions. On a similar line, Tan *et al.* [293] proposed a Transformer-based model that has demonstrated improved performance thanks to a pre-training phase on large amounts of image-sentence pairs.

Feature Aggregation Methods

The aggregation of spatial and temporal features is one of the key challenges in deep learning architectures. Different solutions have been proposed and heavily depend on the domain in which applying the aggregation functions (*i.e.* images or text). While fusing and pooling operations applied over depths, scales, and resolutions constitute fundamental components in visual recognition architectures, the sequential nature of textual data requires different strategies to reduce feature dimensionality.

Regarding the visual domain, with the first strategies adopted in early popular deep learning models [176, 280, 292], the architecture design has moved in last few years to deeper and wider networks [119, 339, 136] incorporating bottlenecks and connectivity novelties like skipping, gating, and aggregating mechanisms. While going deeper, *i.e.* aggregating across channels and depths, improves the semantic recognition accuracy, spatial fusion, *i.e.* aggregating across scales and resolutions, is needed to achieve a better localization capability. In this context, feature pyramid networks [200] are the predominant approach, making use of top-down and lateral connections between feature hierarchical levels.

On a different note, data with a sequential nature such as textual sentences require different solutions to take into account the temporal dependencies between elements. In this setting, the use of recurrent neural networks has remained the most commonly used strategy, where hidden representations, learned through memory and gating mechanisms, are adopted as global encoding of a sequence of feature vectors.

Recently, with the advent of fully attentive architectures [306] that overcame limitations of recurrent networks, novel solutions based their global understanding of sequences through the addition of a special CLS token at the beginning of each sequence [78, 293]. Thanks to the use of attention that models inter- and intra-modality connections, this CLS token can learn a compact representation of an input sequence for general classification purposes. Additionally, similar efforts have been made on the encoding of textual sentences, where again mean and max pooling or CLS token have remained the predominant aggregation approaches [142, 258].

Differently from previous works, we propose a novel aggregation method based on attentive mechanisms that can reduce in a learnable way a set or a sequence of features coming from either the visual or textual domain.

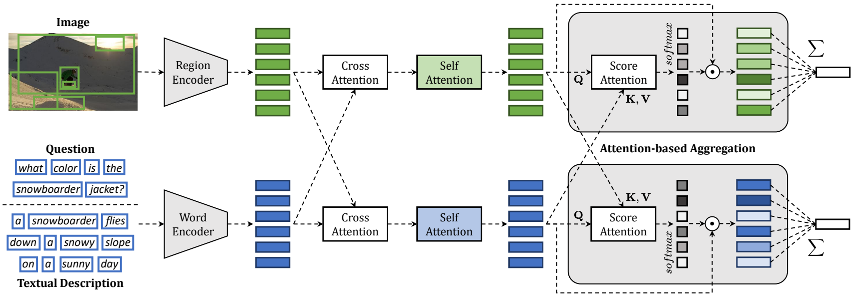


Image
Region Encoder
Cross Attention
Self Attention
Score Attention
Q
K, V
softmax
Σ
Attention-based Aggregation
Question
what color is the snowboarder jacket?
a snowboarder flies down a snowy slope on a sunny day
Textual Description
Word Encoder

Figure 3.2: Our architecture for cross-modal feature extraction and matching. After a cross-modal feature extraction stage, the proposed attention-based aggregation function aligns and reduces vectors from both modalities into compact and cross-modal representations.

3.2.3 Proposed method

The popular scaled dot-product attention mechanism operates on a set of input vectors and generates a new set of vectors, where each one has been updated with relevant information coming from the others. This has been proved largely effective in sequence to sequence tasks such as natural language understanding and machine translation [306, 78]. However, visual-semantic tasks such as visual question answering and cross-modal retrieval deal with multi-modal input sequences and require alignment between modalities and global perspectives in order to reach a classification or similarity output.

To this end, we propose a novel attention-based aggregation function that learns to align and combine two sets of features into a global and compact representation based on the cross-domain connections between modalities. In the simplest case, the two sets of features will be regions from an input image and word features from a natural language sentence.

In a nutshell, our approach leverages dot-product attention to compute cross-modal scores for each element of the two feature sets. The weights are then used to take a weighted sum of the input feature vectors, reducing the two sets into a pair of vectors which can be used for classification or ranking.

In the following, we firstly present our attention-based reduction method. With the aim of testing the operator on both image-text matching and visual question answering, we then introduce a general architecture for both tasks, where features from multi-modal inputs are extracted and combined. In the last section, we discuss the final stages of the architecture and the training choices.

Attention-based Aggregation Function

Motivated by the need of leveraging the information contained in sequence of vectors and at the same time to compare multi-modal information, our aggregation function is based on the scaled dot-product attention mechanism [306].

To recall what attention is, given three sets of vectors, *i.e.* queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$, scaled dot-product attention computes a weighted sum of the value vectors according to a similarity distribution between query and key vectors. This is usually done in a multi-head fashion, so that for each head h the attention operator is defined as

$$\mathsf{Attention}_h(\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d}}\right) \boldsymbol{V}_h, \tag{3.1}$$

where $\boldsymbol{Q}_h$ is a matrix of n_q query vectors, $\boldsymbol{K}_h$ and $\boldsymbol{V}_h$ both contain n_k keys and values, and d is the dimensionality of queries and keys, used as a scaling factor.

In the case of self-attention, queries, keys and values are obtained for each head as linear projections of the same input vectors belonging to a single modality, while in cross-attention, queries are a projection of one modality vectors and keys and values are projections of the other modality vectors. Inspired by cross-attention, we define a *Score Attention* operator which computes a relevance score for each element of the query sequence, considering its relationships with keys and values coming from the other modality.

Formally, given the set of query, key and value vectors from all heads, our Score dot-product attention can be formulated as

$$\mathsf{ScoreAttn}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \mathrm{fc}\left(\left[\mathrm{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^T}{\sqrt{d}}\right) \boldsymbol{V}_h\right]_h\right), \tag{3.2}$$

where $\mathcal{Q}$, $\mathcal{K}$, $\mathcal{V}$ indicate the set of queries, keys and values for the different heads, $[...]_h$ indicates the concatenation of the outputs of all heads and fc is a linear projection that outputs a single scalar score for each input query.

In order to learn complex interactions between modalities and therefore to guide the reduction process based on the cross-domain relations, Score Attention is applied on queries from one modality and keys and values from the other modality. Therefore, given a set of input vectors $\boldsymbol{X}$ coming from one modality (*e.g.* regions of an image) and a set of input vectors $\boldsymbol{Z}$ coming from the other modality (*e.g.* words of a text), we obtain a final condensed representation for $\boldsymbol{X}$ conditioned on $\boldsymbol{Z}$ as a weighted sum of its vectors using the scores provided by the Score Attention operator, *i.e.*

$$\boldsymbol{Y}(\boldsymbol{X}, \boldsymbol{Z}) = \sum_{i=0}^{n_q} \boldsymbol{S}_i(\boldsymbol{X}, \boldsymbol{Z}) \cdot \boldsymbol{X}_i \tag{3.3}$$

$$\boldsymbol{S}(\boldsymbol{X}, \boldsymbol{Z}) = \mathrm{softmax}\left(\mathsf{ScoreAttn}(\mathcal{Q}, \mathcal{K}, \mathcal{V})\right), \tag{3.4}$$

where queries $\mathcal{Q}$ are obtained as projections of $\boldsymbol{X}$, while keys and values $\mathcal{K}$, $\mathcal{V}$ are obtained from $\boldsymbol{Z}$. The softmax is applied over the n_q scores returned by the Score Attention operator. Conversely, the same applies to the reduction of the other modality $\boldsymbol{Z}$ by considering $\boldsymbol{Z}$ as query sequence and $\boldsymbol{X}$ as key and value ones.

As it can be seen from Eq. (3.2) and (3.4), our Score Attention operator can be thought as a cross-attention that, instead of yielding a sequence of vectors, computes a sequence of scores conditioned on keys and values from the other modality. Therefore, the final compressed representation for each modality can capture a global perspective of the input, focusing on elements that show higher importance with respect to the cross-domain interactions.

Noticeably, this aggregation function can be executed multiple times in parallel with different query, key and value projections, thus yielding more than one output vector. This in principle can foster a more disentangled representation, in which different output vectors refer to different global aspects of the same input features. We therefore test our method with different number of compressed vectors, and we refer later to this hyper-parameter with k. Whenever the number of vectors is more than one, we average their contributions with a non-learnable reduction operator. More details on this can be found in the Implementation Details section.

Visual-Semantic Model

To test our aggregation operator, we devise a general architecture for cross-modal feature extraction and matching, with the aim of tackling different tasks with the same common pipeline. Specifically, the architecture is tested on both image-text retrieval and visual question answering. Given input regions from an image, and words from a textual description, we adopt a bi-directional GRU as text encoder, retaining for each word the average embedding between the forward hidden state and the backward hidden state. On the visual side, instead, we apply a linear projection to the features of image regions.

Following recent progress in fully-attentive models and cross-modality interaction [306, 293], after this encoding stage we propagate visual and textual features with a cross-attention operation, followed by a self-attention for each modality. On top of this, two instances of the aggregation operator are applied, one for each modality, thus obtaining one global vector for each modality. A summary of the overall architecture is reported in Fig. 3.2.

Training

The last stage of the model and the training objectives depend on the specific task. In the following we report the main differences.

Visual Question Answering. After applying the aggregation operator, the two vector representations are concatenated and fed to a fully connected layer which is in charge of predicting the final answer class. Additionally, in the case of VQA, we add a position-wise feed-forward layer between the reduction operator and the final concatenation for class prediction.

During the training phase, we employ the binary cross-entropy loss in a multi-label fashion, *i.e.* applying it independently for all classes. For fairness of comparison, we do not make use of any data augmentation strategy and do not employ any external data source like part of the VQA literature does.

Cross-modal Retrieval. In the case of image-text matching, instead, the compressed vectors given by the application of the aggregation operator are compared with a cosine similarity to measure their similarity score. During training, we adopt an hinge-based triplet ranking loss, which is the most common ranking objective in the retrieval literature. Following Faghri *et al.* [85], we only backpropagate the loss obtained on the hardest negatives found in the mini-batch. Given image and sentence pairs (I, T), our final loss with margin α is thus defined as

$$\begin{aligned} L_{hard}(I,T) = &\max_{\hat{T}} \left[\alpha - S(I,T) + S(I,\hat{T})\right]_{+} \\ &+ \max_{\hat{I}} \left[\alpha - S(I,T) + S(\hat{I},T)\right]_{+}, \end{aligned}$$

where S indicates the cosine similarity, $[x]_{+} = \max(x, 0)$, $\hat{T}$ is the hardest negative sentence and $\hat{I}$ is the hardest negative image.

3.2.4 Experimental evaluation

In this section, we report the results on the two considered visual-semantic tasks (*i.e.* visual question answering and cross-modal retrieval) by comparing our attention-based aggregation function with respect to different baselines. First, we provide implementation details and introduce the datasets used in our experiments.

Datasets

To validate the effectiveness of our solution, we employ two of the most widely used datasets containing visual-semantic data. In particular, we carry out the experiments on the VQA 2.0 [104] and COCO [201] datasets to address visual question answering and cross-modal retrieval, respectively.

COCO. The dataset contains more than 120 000 images, each of them annotated with 5 different textual descriptions. We follow the splits provided by Karpathy *et al.* [159], where 5 000 images are used for validation, 5 000 for testing and the rest for training. Following the standard evaluation protocol [85], retrieval results on this dataset are reported by averaging over 5 folds of 1 000 test images each.

VQA 2.0. The dataset is composed of images coming from the COCO dataset and are divided in training, validation, and test according to the official splits. For each image, three questions are provided on average. These questions are divided into three different categories: `Yes/No`, `Number`, and `Others`. Each image-question pair is annotated with 10 answers collected by human annotators, and the most frequent answer is selected as the correct one. We report experimental results on the validation and test-dev sets of this dataset, only using the training split to train our model. Differently from standard literature that uses additional training data coming from different datasets, we only focus on image-question-answer triplets from this dataset.

Implementation Details

To encode image regions, we employ the Faster R-CNN model finetuned on the Visual Genome dataset [175, 7], obtaining a 2048-dimensional feature vector for each region. We reduce the dimensionality of region feature vectors by feeding them to a fully connected layer with a size of 512. For each image, we select the top 36 regions with the highest class detection confidence score. As mentioned, to encode word vectors, we use a bi-directional GRU with a single layer using either

Table 3.1: Accuracy results on VQA 2.0 dataset. The results are reported on the validation and test-dev splits. All models are trained only on the VQA 2.0 training split.

	Validation				**Test-Dev**			
Aggregation Function	All	Yes/No	Number	Others	All	Yes/No	Number	Others
Mean Pooling	54.87	71.50	37.93	46.69	56.05	71.00	38.88	47.19
Max Pooling	56.73	75.68	37.64	47.37	57.95	75.14	38.48	47.69
LogSumExp Pooling	54.61	70.94	38.27	46.53	55.68	70.36	38.72	47.00
1D Convolution	56.87	72.35	39.18	49.79	57.79	71.71	39.97	49.96
CLS Token	58.31	74.29	39.89	51.03	59.40	74.26	40.31	51.07
Ours ($k = 1$)	60.73	77.68	41.86	**52.84**	62.05	77.84	42.47	**53.03**
Ours ($k = 2$)	60.76	78.06	42.32	52.48	62.06	78.26	42.62	52.66
Ours ($k = 3$)	60.50	77.82	41.56	52.33	61.80	78.22	41.69	52.35
Ours ($k = 5$)	**60.99**	**78.62**	42.53	52.46	62.17	78.52	42.27	52.74
Ours ($k = 7$)	60.95	78.40	**42.65**	52.53	**62.43**	**78.75**	**43.33**	52.83
Ours ($k = 10$)	59.94	77.30	40.82	51.80	61.16	77.39	40.69	51.97

learned or pre-trained word embeddings to represent words of the sentence. We set the hidden size of the GRU layer to 512.

Following the standard implementation [306], each scaled dot-product attention also includes a dropout, a residual connection, and a layer normalization. We set the dimensionality d of each layer to 512, the number of heads in both scaled dot-product and score attention to 8, and the dropout keep probability to 0.9. In all our experiments, we use Adam [170] as optimizer and a batch size equal to 64.

Visual Question Answering. For VQA models, we set the initial learning rate to 0.0005 decreased by a factor of 10 every 10 epochs. To represent words, we use and finetune the pre-trained GloVe word embeddings [242] with a word dimensionality equal to 300. We set the maximum length of input questions to 14, padding the shorter ones. For the additional position-wise feed forward layer used in VQA models, we set the hidden size to the same dimensionality d of attention layers. When we use a number of compressed vectors k larger than 1, we average the k vectors of each modality to obtain a single compact representation for both image regions and words.

Following a common practice in the VQA task [7], the set of candidate answers is limited to correct answers in the training set that appear more than 8 times, resulting in an output vocabulary size equal to $3\,129$.

Cross-modal Retrieval. We set the initial learning rate to 0.00007 decayed by a factor of 10 every 10 epochs, and the margin α of the triplet loss function to 0.2. Also, we clip the 2-norm of vectorized gradients to 2.0. To encode words, we use one-hot vectors and linearly project them with a learnable embedding matrix to the word dimensionality of 300. To create the word vocabulary, we take into account only the words that appear at least 5 times in the training and validation sets.

In our attention-based aggregation function, when the number of compressed vectors k is larger than 1, we compute a pair-wise cosine similarity between each pair of compressed vectors coming from the two modalities, and we average the resulting k similarity scores. Intuitively, each aggregation module learns to extract and compare different relevant information, specializing each vector to distinct semantic meaning.

Baselines

To evaluate the proposed method, we compare our results with respect to five different aggregation functions, namely mean pooling, max pooling, log-sum-exp pooling, 1D convolution, and CLS token. For all baselines, we employ the pipeline defined in Sec. 3.2.3, and the same hyper-parameters and implementation choices used for our architecture.

Mean Pooling. The mean aggregation function is one of the most common approaches for feature reduction and refers to the global average pooling between each vector of the input sequence. Since input sequences may have different lengths, in our experiments the mean pooling operation is computed using only the valid elements of the sequence.

Max Pooling. Similarly to the mean operation, the max pooling is another commonly used strategy to reduce feature dimensionality and selects the maximum activation in the feature maps. In our setting, we apply max pooling to the sequence dimension, thus obtaining a single summarized vector for each input sequence.

LogSumExp Pooling. It can be considered as a smooth approximation of the maximum function and is defined as the logarithm of the sum of the argument exponentials. We apply this operation along the feature dimension thus condensing the most important features for each vector of the sequence.

1D Convolution. Convolution is the fundamental operation of CNNs and works well for identifying patterns in data. We test 1D convolutions applied to the sequence dimension to obtain a compact and aggregated representation of the whole set of vectors. In our experiments, we set the kernel size equal to the input sequence length.

CLS Token. Following the recent progress of pre-training strategies and cross-modality matching [78], we also consider the integration of a special CLS token at the beginning of each input sequence. Thanks to the cross- and self-attention

Table 3.2: Comparison between different word embedding strategies on VQA 2.0 validation set.

Aggregation Func.	**Word Emb.**	All	Yes/No	Number	Others
Ours $(k = 5)$	Learned	59.29	77.24	40.29	50.66
Ours $(k = 5)$	GloVe	60.98	78.51	42.20	**52.61**
Ours $(k = 5)$	GloVe Finetuned	**60.99**	**78.62**	**42.53**	52.46
Ours $(k = 7)$	Learned	59.23	76.98	40.02	50.80
Ours $(k = 7)$	GloVe	**61.13**	**79.13**	42.13	52.47
Ours $(k = 7)$	GloVe Finetuned	60.95	78.40	**42.65**	**52.53**

operations, the CLS token can be used as a final compact representation of the entire sequence. We add a CLS token for each modality and use them in last stage of the pipeline according to the specific task.

Visual Question Answering Results

Experimental results for the VQA task are shown in Table 3.1 by comparing our aggregation function with respect to the aforementioned baselines. For each method, we report the accuracy on all image-question pairs of the considered splits and the accuracy values on the three question categories of the VQA 2.0 dataset (*i.e.* `Yes/No`, `Number`, and `Others`).

As it can be seen, our method surpasses all other aggregation functions by a significant margin on both validation and test-dev splits. With respect to the CLS token, which is the top performing baseline in this task, our solution achieves an improvement of 2.68% and 3.03% in terms of overall accuracy on the validation and test-dev splits, respectively.

Additionally, we test our attention-based aggregation method by using a different number of k compressed vectors and different word embedding strategies. In the bottom section of Table 3.1, we report the accuracy results by varying the number of compressed vectors. As it can be noticed, the model with 1 vector reaches good results surpassing all other baselines. Nevertheless, higher performances can be achieved with 5 and 7 compressed vectors suggesting that a correct answer can be positively influenced by capturing different aspects of the input features. Above a certain numbers of k vectors, we instead observe a degradation of the performance, as demonstrated by the results with 10 vectors. This can be explained by the greater complexity of the model that undermines the benefits of learning different global vectors.

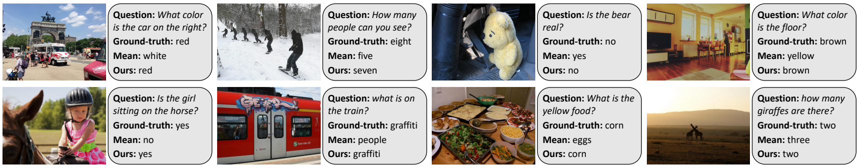


Figure 3.3: Qualitative results on VQA 2.0 validation set. For each image, we report a sample question, the ground-truth answer, and the corresponding answers predicted by our aggregation function and by the mean pooling operation.

In Table 3.2, we show the performance on the VQA 2.0 validation set when using different word embedding strategies. In particular, we compare the results by employing learnable word embeddings and pre-trained GloVe vectors, either fixed or finetuned during training. In our experiments, the GloVe word embeddings lead to an improvement of the final accuracy results using both 5 and 7 compressed vectors. The performance gap between fixed and finetuned GloVe vectors is not very large, but a slight improvement is given when using the finetuned version. For this reason, all experiments are carried out by using the GloVe vectors finetuned during training. On the contrary, learning word embeddings from scratch brings to lower performances in all settings.

Qualitative Results. Some sample results on the VQA 2.0 validation set are reported in Fig. 3.3. For each image, we show the corresponding question, the ground-truth correct answer and the answers predicted by our attention-based aggregation function and the mean pooling operation. The results demonstrate the effectiveness of our strategy also from a qualitative point of view and confirm better performance than one of the most widely used solution to aggregate features. Our method is able to correctly identify the color of the objects contained in the question and count the number of instances of a given entity. Also, it can accurately answer either simple (*e.g.* `Yes/No`) or more complex questions that require a complete understanding of the scene.

Cross-modal Retrieval Results

Table 3.3 shows the results for the cross-modal retrieval task on the COCO test set. For both text and image retrieval, we report the results in terms of recall@K (with $K = 1, 5, 10$) which measures the portion of query images or query captions for which at least one correct result is found among the top-K retrieved elements.

Table 3.3: Cross-modal retrieval results on Microsoft COCO 1K test set.

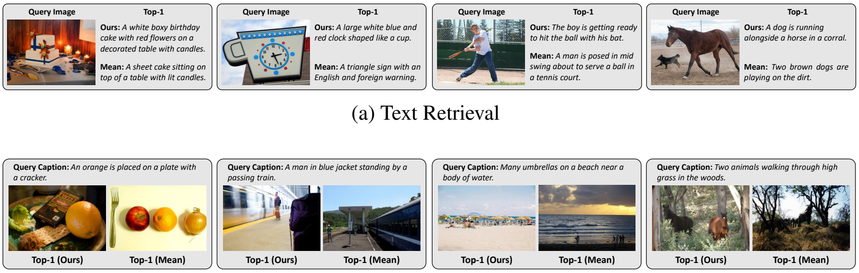
	Text Retrieval			**Image Retrieval**		
Aggregation Function	R@1	R@5	R@10	R@1	R@5	R@10
Mean Pooling	69.66	93.12	97.64	50.42	82.27	90.83
Max Pooling	69.04	92.68	96.98	51.20	83.27	91.52
LogSumExp Pooling	64.20	91.52	96.84	47.22	82.26	91.23
1D Convolution	65.66	91.86	96.58	49.25	81.43	90.42
CLS Token	70.30	93.38	97.24	51.05	83.28	**91.80**
Ours $(k = 1)$	**70.80**	93.16	97.24	50.77	82.76	91.31
Ours $(k = 2)$	70.36	**93.46**	97.20	**51.31**	**83.38**	91.69
Ours $(k = 3)$	70.42	93.34	97.22	50.98	83.17	91.65
Ours $(k = 4)$	70.14	93.42	**97.76**	50.82	82.66	91.14

Also in this setting, we compare our aggregation function with respect to the previously defined baselines and we analyze the performance by varying the number of compressed vectors used to aggregate input sequences.

As it can be seen, our attention-based aggregation achieves the best results among all considered aggregation functions on both text and image retrieval. Also in this case, the CLS token results to be the top performing baseline according to all evaluation metrics, confirming the importance of using inter- and intra-modality interactions to reduce feature dimensionality.

Differently from the VQA task in which the best results are obtained with 5 and 7 compressed vectors, the best performances are instead achieved with a lower number of vectors (*i.e.* 2 and 3), as shown in the bottom section of Table 3.3. In this setting, we do not find beneficial the use of GloVe word vectors and all results are thus obtained by learning word embeddings during training. This suggests that the large amount of textual data contained in the COCO dataset compared to that available for the VQA task can lead to specific and more suited word embedding representations.

Qualitative Results. Finally, we show some sample results for text and image retrieval in Fig. 3.4a and 3.4b, respectively. Also in this case, we compare our results with those obtained by using the mean pooling aggregation function. As it can be seen, these qualitative results further confirm the effectiveness of our solution leading to increased and more accurate performance on both settings.


Query Image
Top-1
Ours: A white boxy birthday cake with red flowers on a decorated table with candles.
Mean: A sheet cake sitting on top of a table with lit candles.
Query Image
Top-1
Ours: A large white blue and red clock shaped like a cup.
Mean: A triangle sign with an English and foreign warning.
Query Image
Top-1
Ours: The boy is getting ready to hit the ball with his bat.
Mean: A man is posed in mid swing about to serve a ball in a tennis court.
Query Image
Top-1
Ours: A dog is running alongside a horse in a corral.
Mean: Two brown dogs are playing on the dirt.
Query Caption: An orange is placed on a plate with a cracker.
Top-1 (Ours)
Top-1 (Mean)
Query Caption: A man in blue jacket standing by a passing train.
Top-1 (Ours)
Top-1 (Mean)
Query Caption: Many umbrellas on a beach near a body of water.
Top-1 (Ours)
Top-1 (Mean)
Query Caption: Two animals walking through high grass in the woods.
Top-1 (Ours)
Top-1 (Mean)

(a) Text Retrieval

(b) Image Retrieval

Figure 3.4: Qualitative results for text and image retrieval. For each sample, we report the top-1 result retrieved by our aggregation function and by the mean pooling operation.

3.3 ALADIN: efficient image-text matching

In literature, image-text matching is often used as a pre-training objective to forge architectures able to jointly deal with images and texts. Nonetheless, solving this task is of critical importance in cross-modal search engines, where finding images related to a given query text or vice-versa is a well-known application. Many recent methods proposed effective solutions to the image-text matching problem, mostly using recent large vision-language (VL) Transformer networks. However, these models are often computationally expensive, especially at inference time. This prevents their adoption in large-scale cross-modal retrieval scenarios, where results should be provided to the user almost instantaneously. In this section, we propose to fill in the gap between effectiveness and efficiency by proposing an ALign And DIstill Network (ALADIN). ALADIN first produces high-effective scores by aligning at fine-grained level images and texts. Then, it learns a shared embedding space – where an efficient kNN search can be performed – by distilling the relevance scores obtained from the fine-grained alignments. We obtained remarkable results on MS-COCO, showing that our method can compete with state-of-the-art VL Transformers while being almost 90 times faster. The code for reproducing our results is available at `https://github.com/mesnico/ALADIN`.

3.3.1 Introduction

As already mentioned, by understanding the hidden semantic connections between a text and an image, many works in literature solved challenging multi-modal problems, such as image captioning [7, 65, 284] or visual question answering [7, 381, 18]. Among these tasks, *image-text matching* has crucial importance [172, 85, 63, 227, 228] and it consists of outputting a relevance score for each given (image, text) pair, where the score is high if the image is relevant to the text and low otherwise. Although this task is usually employed as a vision-language pre-training objective, it is crucial for cross-modal retrieval, which usually consists of two sub-tasks: *image retrieval*, where we want images relevant to a given text, and *text retrieval*, where we ask for sentences better describing an input image. Efficiently and effectively solving these retrieval tasks is strategically important in modern cross-modal search engines.

Many state-of-the-art models for image-text matching, like Oscar [196] or UNITER [49], comprise large and deep multi-modal vision-language (VL) Transformers with early fusion, which are computationally expensive, especially during the inference phase. In fact, during inference, all the (image, text) pairs from the test set should be forwarded through the multi-modal Transformer to obtain the relevance scores. This is clearly unfeasible in large datasets and unusable in large-scale retrieval scenarios, where the system latency should be as small as possible.

For achieving such a performance objective, many approaches in the literature project image and text embeddings in a common space where similarity is measured through simple dot products. This allows the introduction of an *offline* phase, in which all the dataset items are encoded and stored, and an *online* phase in which only the query is forwarded through the network and compared with all the offline-stored elements. Although these approaches are very efficient, they are usually not sufficiently effective as the ones relying on early modality fusion using large VL Transformers.

In the light of these observations, in this section we propose an ALign And DIstill Network model (*ALADIN*), which exploits the knowledge acquired by large VL Transformers to craft an efficient yet effective model for image-text retrieval. In particular, we employ late fusion approaches so that the two visual and textual pipelines are kept separated until the final matching phase. The first objective consists of *aligning* image regions with sentence words, using a simple yet effective alignment head. Then, a common visual-textual embedding space is learned by distilling the scores from the alignment head using a learning-to-rank objective. In this case, we use the learned alignment scores as ground-truth (teacher) scores.

Our approach is inspired by the recent success of knowledge distillation [12, 21, 35, 338, 382], used to transfer knowledge from a large model to a smaller and more efficient one. We propose to use scores distillation to learn a visual-textual common space, employing the knowledge acquired by a pre-trained VL Transformer. In this case, the knowledge distillation is framed as a learning-to-rank problem [34, 245, 30], widely used in literature but, as far as we know, never used for distilling cross-modal scores.

We show that, on the widely used MS-COCO dataset, the alignment scores can reach results comparable with large joint vision-language models such as UNITER and OSCAR, while being far more efficient, especially during inference. On the other hand, the distilled scores used to learn the common space can defeat previous common space methods on the same dataset, opening the way toward metric-based indexing for large-scale retrieval.

To sum up, in this section, we propose the following contributions:

- We employ two instances of a pre-trained VL Transformer as a backbone for extracting separate visual and textual features.

- We adopt a simple yet effective alignment method for producing high-quality scores instead of the poorly-scalable output of large joint VL Transformers.

- We create an informative embedding space by framing the problem as a learning-to-rank task and distilling the final scores using the scores in output from the alignment head.

3.3.2 Proposed method

The proposed architecture is composed of two different stages. The first stage, which we refer to as *backbone*, is composed of the layers of a pre-trained large vision-language transformer – VinVL [371], an extension to the powerful OSCAR model [196]. In the backbone, the language and the visual paths do not interact through cross-attention mechanisms so that the features from the two modalities can be extracted independently at inference time.

The second stage, instead, is composed of two separate *heads*: the *alignment* head and *matching* head. The alignment head is used to pre-train the network to efficiently align the visual and the textual concepts in a fine-grained manner, as done in TERAN [227]. Differently, the matching head is used to construct an

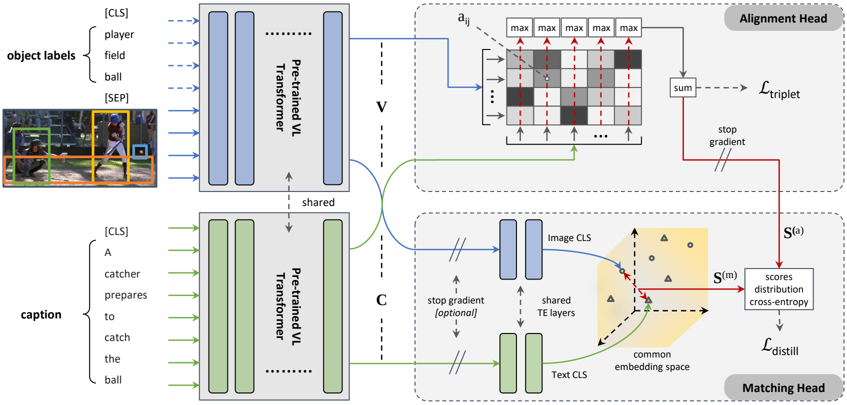


Figure 3.5: Overview of our architecture. The backbone extracts visual and textual features that are used in both the matching and alignment heads. The matching head is trained by distilling the scores using the ones coming from the alignment head.

informative cross-modal common space, that can be used to efficiently represent images and text as fixed-length vectors for use in large-scale retrieval. The scores from the matching head are distilled using the scores from the alignment head as guidance. The overall architecture is shown in Figure 3.5.

In the following, we dive into the building blocks of the architecture – *i.e.*, the backbone, the alignment head, and the matching head.

Vision-Language Backbone

As the backbone for feature extraction, we use the pre-trained layers from VinVL [371], an extension to the large-scale vision-language OSCAR model [196]. Our goal is to obtain suitable vectorial representations for the image $\mathcal{V}$ and the text $\mathcal{C}$ in input. In particular, we employ the model pre-trained on the image-text retrieval task. The authors used a binary classification head on top of the CLS token of the output sequence, and the model is trained to predict if the input images and textual sentences are related or not.

In our use case, the visual and textual pipelines should be separated, so that they can be forwarded independently at inference time. For this reason, we use two instances of the VinVL architecture, in a shared-weights configuration to forward the two modalities independently, as shown in Figure 3.5.

As in [371], we use as visual tokens both the visual features extracted from object regions[1] and their labels, and the two sub-sequences are separated by a SEP token. In the end, the outputs from the last layers of the disentangled VinVL architecture are two sequences, $\boldsymbol{V} = \{\boldsymbol{v}_{\text{cls}}, \boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_N\}$, representing the image $\mathcal{V}$, and $\boldsymbol{C} = \{\boldsymbol{c}_{\text{cls}}, \boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_M\}$, representing the text $\mathcal{C}$. Note that, in both sequences, the first element is the CLS token, used to collect representative information for the whole image or text.

Alignment Head

The alignment head comprises a similarity matrix that computes the fine-grained relevances between the visual tokens $\boldsymbol{V}$ and textual tokens $\boldsymbol{C}$. The fine-grained similarities are then pooled to obtain the final global relevance between the image and the text. In particular, we use a formulation similar to the one used in TERAN [227]. Specifically, the features in output from the backbone are used to compute a visual-textual tokens alignment matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, built as follows:

$$\boldsymbol{A} = a_{ij}^{kl} = \text{cosine}(\boldsymbol{v}_i, \boldsymbol{c}_j) = \frac{\boldsymbol{v}_i^T \boldsymbol{c}_j}{\|\boldsymbol{v}_i\| \|\boldsymbol{c}_j\|} \qquad i \in g_k, j \in g_l, \tag{3.5}$$

where g_k is the set of indexes of the region features from the k-th image and g_l is the set of indexes of the words from the l-th sentence. At this point, the similarities s_{kl} between the image k and the caption l are computed by pooling the similarity matrix $\boldsymbol{A}$ along dimensions (i, j) through an appropriate pooling function. Guided by [227], we use the *max-over-regions sum-over-words* policy, which computes the following final similarity score:

$$\boldsymbol{S}^{(\text{a})} = s_{kl}^{(\text{a})} = \sum_{j \in g_l} \max_{i \in g_k} A_{ij}. \tag{3.6}$$

The dot-product similarity used to compute $\boldsymbol{A}$ in Eq. 3.5 resembles the computation of the cross-attention between visual and textual tokens. The difference boils down to the interaction between the visual and textual pipelines, which happens only at the very end of the whole architecture. This *late cross-attention* makes the sequences $\boldsymbol{V}$ and $\boldsymbol{C}$ cacheable, eliminating the need to forward the whole architecture whenever a new query – either visual or textual – is issued to the system. The computation of $\boldsymbol{S}^{(\text{a})}$, involving only simple non-parametric

[1] https://github.com/microsoft/scene_graph_benchmark

operations, is very efficient and can be easily implemented on GPU to obtain high inference speeds.

The loss function used to force this network to produce suitable similarities s for each (image, text) pair is the *hinge-based triplet ranking loss*, used in previous works [85, 191, 227]. Formally,

$$\mathcal{L}_{\text{triplet}} = \sum_{k,l} \max_{l'}[\alpha + s_{kl'} - s_{kl}]_+ + \max_{k'}[\alpha + s_{k'l} - s_{kl}]_+, \tag{3.7}$$

where s_{kl} is the similarity estimated between image k and caption l, and $[x]_+ \equiv \max(0, x)$; the values k', l' are the indexes of the image and caption hard negatives found in the mini-batch as done in [85], and α is a margin that defines the minimum separation that should hold between positive and negative pairs.

Given that the alignment head is directly connected to the backbone, we fine-tuned the backbone on this new alignment objective. More details on the training procedure are reported in Section 3.3.2.

Matching Head

The matching head uses the same sequences $\boldsymbol{V}$ and $\boldsymbol{C}$ given from the backbone and employs them to produce the features $\tilde{\boldsymbol{v}} \in \mathbb{R}^d$ for the image $\mathcal{V}$ and $\tilde{\boldsymbol{c}} \in \mathbb{R}^d$ for the caption $\mathcal{C}$. These representations are forced to lay in the same d-dimensional embedding space. In this space, k-neirest-neighbor search can be efficiently computed — using metric space approaches or inverted files — to quickly retrieve images given a textual query or vice-versa. Specifically, we forward $\boldsymbol{V}$ and $\boldsymbol{C}$ through a 2-layer Transformer Encoder (TE):

$$\bar{\boldsymbol{V}} = \text{TE}(\boldsymbol{V}); \qquad \bar{\boldsymbol{C}} = \text{TE}(\boldsymbol{C}). \tag{3.8}$$

As in [229], the TE shares the weights among the two modalities, and the final vectors encoding the whole image and caption are the CLS tokens in output from the TE layers: $\tilde{\boldsymbol{v}} = \bar{\boldsymbol{V}}[0] = \bar{\boldsymbol{v}}_{\text{cls}}$ and $\tilde{\boldsymbol{c}} = \bar{\boldsymbol{C}}[0] = \bar{\boldsymbol{c}}_{\text{cls}}$. The final relevances are simply computed as the cosine similarities between the the vector $\tilde{\boldsymbol{v}}_k$ from the k-th image and $\tilde{\boldsymbol{s}}_l$ from the l-th sentence: $\boldsymbol{S}^{(\text{m})} = s^{(\text{m})}_{kl} = \text{cosine}(\tilde{\boldsymbol{v}}_k, \tilde{\boldsymbol{s}}_l)$.

In principle, we could optimize the common space using the same hinge-based triplet ranking loss in Eq. 3.7 already used to train the alignment head. Instead, in the light of the good effectiveness-efficiency trade-off of the alignment head, we propose to learn a distribution for $\boldsymbol{S}^{(\text{m})}$ using the previously-learned $\boldsymbol{S}^{(\text{a})}$ as teachers.

Specifically, we frame the problem of distilling the distribution of $\boldsymbol{S}^{(\mathrm{m})}$ from $\boldsymbol{S}^{(\mathrm{a})}$ as a *learning-to-rank* problem. We employ the mathematical framework developed in the ListNet approach [34], which models the probability of an object being ranked at the top, given the scores of all the objects. Differently from this framework, here we need to optimize for two different entangled distributions: the distribution of text-image similarities when sentences are used as queries, and the distribution of image-text similarities when instead images are used as queries. In particular, given a textual query k and an image query l, the probabilities of the image i and text j to be the top-one elements respectively with respect to $\boldsymbol{S}^{(\mathrm{a})}$ are:

$$P_{\boldsymbol{S}^{(\mathrm{a})}}(i) = \frac{\exp(s_{ik}^{(\mathrm{a})})}{\sum_{t=1}^{B} \exp(s_{tk}^{(\mathrm{a})})}; P_{\boldsymbol{S}^{(\mathrm{a})}}(j) = \frac{\exp(s_{lj}^{(\mathrm{a})})}{\sum_{t=1}^{B} \exp(s_{tj}^{(\mathrm{a})})} \tag{3.9}$$

where B is the batch size, as the learning procedure is confined to the images and sentences in the current batch. Therefore, during training, only B images are retrieved using the query k, and B textual elements are retrieved using the query l. Similarly, an analogous probability can be defined over $\boldsymbol{S}^{(\mathrm{m})}$:

$$P_{\boldsymbol{S}^{(\mathrm{m})}}(i) = \frac{\exp(\tau s_{ik}^{(\mathrm{m})})}{\sum_{t=1}^{B} \exp(\tau s_{tk}^{(\mathrm{m})})}; P_{\boldsymbol{S}^{(\mathrm{m})}}(j) = \frac{\exp(\tau s_{lj}^{(\mathrm{m})})}{\sum_{t=1}^{B} \exp(\tau s_{tj}^{(\mathrm{m})})} \tag{3.10}$$

where τ is a temperature hyper-parameter which compensates for the fact that $\boldsymbol{S}^{(\mathrm{m})}$ ranges in [0, 1]. We empirically found that $\tau = 6.0$ works well in practice. The final matching loss can be formulated as the cross-entropy between the $P_{\boldsymbol{S}^{(\mathrm{a})}}$ and $P_{\boldsymbol{S}^{(\mathrm{m})}}$ probabilities, for both the image-to-text and text-to-image cases.

$$\mathcal{L}_{\text{distill}} = -\sum_{i=1}^{B} P_{\boldsymbol{s}^{(\mathrm{a})}(i)} \log(P_{\boldsymbol{s}^{(\mathrm{m})}}(i)) - \sum_{j=1}^{B} P_{\boldsymbol{s}^{(\mathrm{a})}(j)} \log(P_{\boldsymbol{s}^{(\mathrm{m})}}(j)) \tag{3.11}$$

Notice that accurate and dense teacher scores are needed to obtain a good estimate of the teacher distributions $P_{\boldsymbol{s}^{(\mathrm{a})}}(i)$ and $P_{\boldsymbol{s}^{(\mathrm{a})}}(j)$. This partly motivates our choice of first researching an effective and efficient alignment head that could output the scores to be used as ground-truth for the matching head.

Training

During the training phase, we initially respect the following constraints: (a) the backbone is finetuned only when training the alignment head, and (b) the gradients do not flow backward through $\boldsymbol{S}^{(\mathrm{a})}$ when training the matching head (as depicted in Figure 3.5 through the *stop-gradient* indication). The constraint (b) comes from the fact that the scores $\boldsymbol{S}^{(\mathrm{a})}$ are used as teacher scores. Therefore, they should not modify the weights of the backbone, because it is assumed that the backbone is already trained with the alignment head. Given these constraints, we train the network in two steps. First, we train the alignment head by updating the backbone weights using $\mathcal{L}_{\text{triplet}}$ (**ALADIN A/ft.** in the experiments). Then, we freeze the backbone and we learn the matching head by updating the weights of the 2-layer Transformer Encoder using $\mathcal{L}_{\text{distill}}$ (**ALADIN D** in the experiments). Note that the formalism *X/ft.* signifies that the gradients coming from that head loss X are used to finetune the backbone. Possible head losses are X=$\{$T, D, A$\}$ for T=triplet, D=distillation, and A=alignment, where T and D come from the matching head, while A from the alignment head. When */ft.* is omitted, it means that the backbone remains frozen.

We explore also the joint training of the two heads. Specifically, we relax constraint (a), so that gradients coming from the two heads can update the backbone. Sticking to the previous formalism, we refer to this experiment as **ALADIN A/ft. + D/ft.**. Nevertheless, when directly applying this training schema, we experienced some instabilities. If the alignment head — working as a teacher for the matching head — is not warmed-up, it can not initially provide good teacher scores. The consequence is that noisy gradients backpropagate through the matching head and interfere with the finetuning of the backbone. For this reason, we warmup the backbone by pre-training it with the alignment loss $\mathcal{L}_{\text{triplet}}$ (as in the *ALADIN A/ft.* setup).

3.3.3 Experimental evaluation

In this section, we report detailed results for validating our approach. In addition to the training setups described in 3.3.2, we consider two more schemes as baselines: **ALADIN T** trains the matching head using the standard hinge-based triplet ranking loss without distillation, starting from a pre-trained backbone (*i.e.* ALADIN A/ft.) and leaving it fixed; similarly, **ALADIN T/ft.** lacks the alignment head and the backbone is finetuned only with the gradients from the matching head.

Dataset and Metrics

We perform our experiments on the widely-used MS-COCO dataset, which contains a large corpus of images scraped from the web. Each image is annotated with 5 textual descriptions. We follow the splits introduced by [159], which reserves 113,287 images for training, 5,000 for validating, and 5,000 for testing. In literature, a smaller test set comprising only 1,000 images is often used. For a fair comparison, we report the results on both 5K and 1K test sets. In the case of 1K images, the results are computed by performing a 5-fold cross-validation and averaging the results.

As commonly done to evaluate cross-modal retrieval models [85, 191, 247, 212, 187], we use the recall@k metric for evaluating the ability of our model to correctly retrieve relevant texts or images. Specifically, the recall@k measures the percentage of queries able to retrieve the correct item among the first k results.

Alignment Head Results

We first compare the results obtained with our alignment head against some recent methods comprising large-scale pre-trained Transformer models (Table 3.4). We consider only the *Base* versions and not the *Large* ones, for hardware limitations. For a fair comparison, we initialize our backbone with the weights of VinVL Base [371]. Notice that, at test time, all the reported models except ours need to compute a number of network forward steps in the order of $O(n^2r)$, where n is the number of images and r is the number of sentences associated to each image ($r = 5$ in case of MS-COCO). In fact, due to cross-attention links between visual and textual pipelines, intermediate representations cannot be cached for being reused with a different query. Instead, given the disentangled pipelines, our model enables caching of the image and text features in output from the backbone for speeding up the retrieval with never seen queries, with a number of network forward steps in the order of $O(n + nr)$. As we can notice from Table 3.4, this disentanglement comes at the cost of a slight reduction of the overall effectiveness, as we can notice by comparing our approach to the VinVL model. Nevertheless, our model ALADIN A/ft. can perfectly compete, and partially overtake, all the previous entangled visual-textual Transformer models on both image and sentence retrieval tasks. From the results on the ALADIN A/ft. + D/ft. model, we can notice that when the distillation loss is also active the alignment scores are pretty comparable to ALADIN A/ft. In particular, on the 5K test set, we observe slight improvements in both image and sentence retrieval. This evidence suggests that the distillation loss has the collateral effect of regularizing its own teacher scores, as done in recent works on self-distillation [370, 35].

Table 3.4: Experiment results using scores from the alignment head. The comparison is performed with entangled visual-textual Transformer models.

		1K Test Set						5K Test Set					
		Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
Model	Training Data	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
12-in-1 [213]	4.4M	-	-	-	65.2	91.0	96.2	-	-	-	-	-	-
VilBERT [212]	3.1M	-	-	-	58.2	84.9	91.5	-	-	-	-	-	-
Unicoder-VL [189]	3.8M	84.3	97.3	99.3	69.7	93.5	97.2	62.3	87.1	92.8	46.7	76.0	85.3
UNITER (Base) [49]	5.6M	-	-	-	-	-	-	63.3	87.0	93.1	48.4	76.7	85.9
OSCAR (Base) [196]	6.5M	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
VinVL (Base) [371]	8.9M	-	-	-	-	-	-	**74.6**	**92.6**	**96.3**	**58.1**	**83.2**	**90.1**
ALADIN A/ft.	8.9M	**88.1**	**99.1**	**99.7**	**75.4**	**95.2**	97.9	70.0	90.7	95.6	54.4	81.0	88.6
ALADIN A/ft. + D/ft.	8.9M	87.6	98.5	**99.7**	75.0	**95.2**	**98.0**	69.9	91.3	95.7	54.7	81.0	88.7

Table 3.5: Experimental results using scores from the matching head. The comparison is performed with methods using disentangled visual-textual pipelines.

		1K Test Set						5K Test Set					
		Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
Model	Training Data	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
TERN [229]	0.6M	65.5	91.0	96.5	54.5	86.9	94.2	40.2	71.1	81.9	31.4	62.5	75.3
SAEM (ens.) [335]	0.6M	71.2	94.1	97.7	57.8	88.6	94.9	-	-	-	-	-	-
CAMERA (ens.) [249]	0.6M	77.5	96.3	98.8	63.4	90.9	95.8	55.1	82.9	91.2	40.5	71.7	82.5
TERAN (ens.) [227]	0.6M	80.2	96.6	99.0	67.0	92.2	96.9	59.3	85.8	92.4	45.1	74.6	84.4
DSRAN (w. BERT) [330]	0.6M	80.6	96.7	98.7	64.5	90.8	95.8	57.9	85.3	92.0	41.7	72.7	82.8
ALADIN T	8.9M	79.2	96.7	99.1	68.9	92.8	96.6	57.9	84.8	91.8	46.0	74.8	84.1
ALADIN D	8.9M	83.1	97.4	99.3	70.5	93.6	97.3	62.7	87.5	93.5	47.4	76.2	85.4
ALADIN T/ft.	8.9M	**84.9**	**98.5**	99.6	71.9	93.8	97.0	63.6	87.4	93.5	49.7	77.7	86.3
ALADIN A/ft. + D/ft.	8.9M	84.7	98.0	**99.8**	**72.7**	**94.5**	**97.5**	**64.9**	**88.6**	**94.5**	**51.3**	**79.2**	**87.5**
CLIP (0-shot) [250]	*0.4B*	-	-	-	-	-	-	*58.4*	*81.5*	*88.1*	*37.8*	*62.4*	*72.2*
ALIGN [149]	*1.8B*	-	-	-	-	-	-	*77.0*	*93.5*	*96.9*	*59.9*	*83.3*	*89.8*

Matching Head Results

We compare the common space created from our matching head with other disentangled methods using similar approaches. The results are shown in Table 3.5. As explained above, for comparison we report also the matching head directly trained using the hinge-based triplet loss (ALADIN T and ALADIN T/ft.) without distilling the scores from the alignment head. Furthermore, for completeness, we report also the results from the recent methods CLIP (0-shot) [250] and ALIGN [149]. Although the comparison with CLIP (0-shot) may result unfair, we decided to stick with the results obtained by the authors of the original paper, to avoid all the intricacies deriving from the hyper-parameter tuning phase needed for a satisfactory fine-tuning stage. However, these models use from 100× to 1000× more training data, so we exclude them from the analysis.

All of our methods outperform the previous models, notably surpassing TERAN [227], the method that introduced the alignment matrix used in the alignment head. Concerning the experiments that non-finetune the backbones (ALADIN T and ALADIN D), we argue that scores distillation helps, especially in the recall@1, where we observe an improvement of about 8% and 2% on sentence and image retrieval respectively for the 5K test set. We obtain the best results by using our model ALADIN A/ft. + D/ft., which jointly trains the alignment and distillation heads by also finetuning the backbone with the respective gradients. The alignment scores from this setup already proved to be effective in Table 3.4. The distilled scores in output from the matching head follow the same trend, obtaining the best results on the 5K test set.

Effectiveness vs Efficiency

To better show the advantage of our model in terms of computing times, in Figure 3.6 we plot the effectiveness vs the efficiency of our approach compared with other methods. We address image-retrieval on the 1K test set, and we report the sum of the recall values (rsum) versus the average time needed to solve a textual query. These experiments are run on a system equipped with an RTX 2080Ti and an AMD Ryzen 7 1700 Eight-Core Processor. As we can notice, the scores from the alignment head (*ALADIN A/ft.*) can directly compete with VL Transformer models, although being almost 20 times faster. Notably, the scores computed on the distilled space from *ALADIN A/ft. + D/ft.* obtain a speedup of almost 90×, with a rsum loss of only about 7% with respect to VinVL. Therefore, the proposed models help fill the gap between efficiency and effectiveness – *i.e.*,

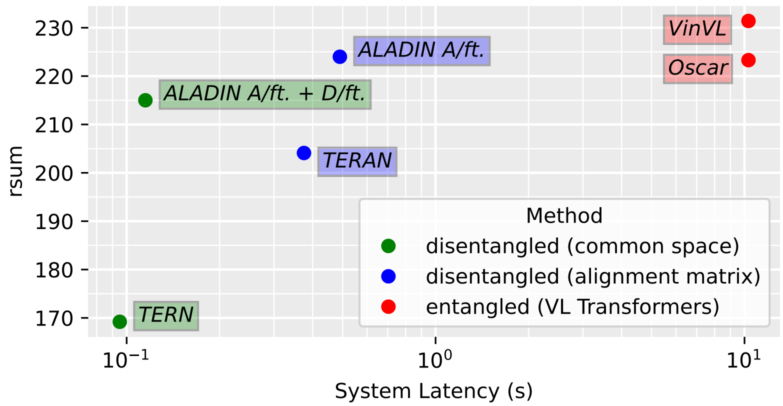


Figure 3.6: Effectiveness vs efficiency. We report effectiveness as the sum of recall values on the image retrieval (rsum), and efficiency as the time needed to search the 5K test images.

the top left zone of the diagram.

Considering the efficiency-effectiveness trade-offs of both the alignment and matching heads, the whole architecture could be deployed in real application scenarios in a two-stage configuration: first, the faster matching head proposes relevant candidates using k-NN search on the common space; then, the candidates are re-ranked using the scores from the alignment head. This pipeline would enable the alignment head, which is slower but more effective, to contribute to the final ranking while keeping the whole system highly scalable.

3.4 Artpedia

As vision and language techniques are widely applied to realistic images, there is a growing interest in designing visual-semantic models suitable for more complex and challenging scenarios. In this section, we address the problem of cross-modal retrieval of images and sentences coming from the artistic domain. To this aim, we collect and manually annotate the *Artpedia* dataset that contains paintings and textual sentences describing both the visual content of the paintings and other contextual information. Thus, the problem is not only to match images and sentences, but also to identify which sentences actually describe the visual content of a given image. To this end, we devise a visual-semantic model that jointly addresses these two challenges by exploiting the latent alignment between visual and textual chunks. Experimental evaluations, obtained by comparing our model to different baselines, demonstrate the effectiveness of our solution and highlight the challenges of the proposed dataset. The Artpedia dataset is publicly available at: `http://aimagelab.ing.unimore.it/artpedia`.

3.4.1 Introduction

As humans, we can seamlessly connect what we visually see or imagine and what we hear or say, therefore building effective bridges between our ability to see and our ability to express ourselves in a common language. In the effort of artificially replicating these connections, new algorithms and architectures have recently emerged for image and video captioning [7, 215, 57] and for visual-semantic retrieval [172, 85, 186]. The former architectures combine vision and language in a generative flavour on the textual side, and in the latter common spaces are built to integrate the two domains and retrieve textual elements given visual queries, and vice versa.

While the standard objective in visual-semantic retrieval is that of associating images and *visual sentences* (*i.e.* sentences that visually describe something), the variety of sentences which can be found in textual corpora is definitely larger, and also contains sentences which do not describe the visual content of a scene. Here, we go a step beyond and extend the task of visual-semantic retrieval to a setting in which the textual domain does not exclusively contain visual sentences, and explore the task of identifying relevant visual sentences given image queries. As such, the task establishes two challenges, the first one being that of understanding whether the sentence has a visually relevant content, and the second being that of associating elements between the two domains.

Further, we also address a second shortcoming of most visual-semantic works, *i.e.* that of dealing with photo-realistic images and simple texts. As there is a growing need of extending these algorithms to less general semantic and visual domains, we both increase the complexity on the visual and on the semantic side. To create an environment where all the aforementioned challenges live together, we focus on the case of artistic data — which surely advertise more complex and unusual visual and semantic features, and propose a new dataset with *visual* and *contextual* sentences for each visual item. In short, visual sentences deal with the visual appearance of the item, contextual ones describe either the item or its context without dealing with its visual appearance.

Table 3.6: Overview of the most relevant datasets containing artistic images.

Dataset	**# Images**	**# Sentences**	**Manually Annotated**	**Task**
Wikipaintings [158]	85,000	-	✗	Classification
Art500k [221]	554,198	-	✗	Classification and retrieval
Brueghel [274]	1,587	-	✓	Near duplicate detection
SemArt [100]	21,383	21,383	✗	Visual-semantic retrieval
EsteArtworks [36]	553	1,278	✓	Visual-semantic retrieval
BibleVSA [20]	2,282	2,271	✓	Visual-semantic retrieval
Artpedia	2,930	28,212	✓	Visual-semantic retrieval (with contextual texts)

We also design and evaluate a model for jointly associating visual and textual elements, and identifying visual textual samples as opposed to contextual ones. Taking inspiration from state of the art models for visual-semantic retrieval, we test both traditional approaches, based on global feature vectors, and approaches that model the latent alignment between visual and textual chunks.

3.4.2 Related work for cultural heritage

Deep Learning techniques often require significant efforts to be applied to the domain of Digital Humanities and Cultural Heritage, due to the presence of specific challenges. The research efforts of the past few years have resulted in various works and applications spanning from generative models to classification and retrieval solutions. On the generative and synthesis side, promising results have been obtained for transferring the style of a painting to a real photograph [101, 265, 156] and inversely, to create a realistic representation of a given painting [383, 298, 299, 300]. On the analysis and feature extraction side, instead, several efforts have been made on the collection and annotation of large scale datasets containing artistic images, mainly focusing on style and genre classification [158, 221, 287], visual patterns detection [274], and artwork instance recognition [73].

For a comprehensive analysis, Table 3.6 shows a summary of the most relevant dataset related to the cultural heritage domain. To the best of our knowledge, there is a limited bunch of works that address the problem of retrieving artistic images from textual descriptions, and vice versa [20, 36, 100]. While [20, 36] take the problem in a semi-supervised way by exploiting the knowledge from large-scale datasets containing realistic images, [100] uses additional metadata such as title, author, genre, and period of the paintings to match images and text. In this section, we instead propose a visual-semantic model capable of discriminating *visual* and *contextual* sentences for each considered painting and, at the same time, associating the corresponding visual and textual elements.

Moreover, in the following Sec. 3.5, after briefly reviewing the most important works related to visual-semantic retrieval, we focus on image-text matching approaches applied to the artistic domain, and subdividing them between supervised and semi-supervised methods.

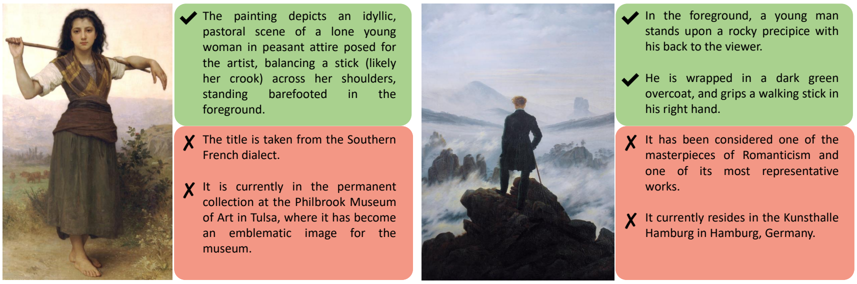
3.4.3 The Artpedia dataset

To foster the research on the development of visual-semantic algorithms which deal with contextual sentences, we propose a novel dataset with visual and contextual sentences describing real paintings. *Artpedia* contains a collection of $2,930$ painting images, each associated to a variable number of textual descriptions. Each sentence is labelled either as a *visual* sentence or as a *contextual* sentence, if does not describe the visual content of the artwork. Contextual sentences can describe the historical context of the artwork, its author, the artistic influence or the place where the painting is exhibited. As in standard cross-modal datasets, the association between sentences and painting is also provided. A sample of the dataset and its annotations is shown in Figure 3.7.

As the name suggests, the dataset has been collected by crawling Wikipedia pages. To this aim, our crawling strategy followed the Wikipedia category hierarchy by navigating all categories containing paintings between the 13th and the 21th century. We then extracted the textual descriptions taking into account all the summaries of each Wikipedia page and the description section whenever present. Finally, we split the text into sentences using the spaCy NLP toolbox[2] and manually annotated each sentence either as visual or contextual. As an additional product of the crawling procedure, we also release the title and the year of each painting, together with the URL of each image.

Overall, Artpedia contains a total of $28,212$ sentences, $9,173$ labelled as visual sentences and the remaining $19,039$ as contextual sentences. On average, each painting is associated with 3.1 visual and 6.5 contextual sentences. The mean length of the textual items is 21.5 words, considerably longer than those of standard image captioning datasets. For a comprehensive analysis of the visual and semantic content of our Artpedia dataset, we report in Figure 3.8 the distribution of paintings over the given range of centuries, the distribution of sentence lengths, and the most common object classes obtained by running a pre-trained object detector [259, 175].

[2] `https://spacy.io/`



✓ The painting depicts an idyllic, pastoral scene of a lone young woman in peasant attire posed for the artist, balancing a stick (likely her crook) across her shoulders, standing barefooted in the foreground.

✗ The title is taken from the Southern French dialect.

✗ It is currently in the permanent collection at the Philbrook Museum of Art in Tulsa, where it has become an emblematic image for the museum.

✓ In the foreground, a young man stands upon a rocky precipice with his back to the viewer.

✓ He is wrapped in a dark green overcoat, and grips a walking stick in his right hand.

✗ It has been considered one of the masterpieces of Romanticism and one of its most representative works.

✗ It currently resides in the Kunsthalle Hamburg in Hamburg, Germany.

Figure 3.7: Sample paintings from our Artpedia dataset with corresponding visual (green boxes) and contextual (red boxes) sentences.

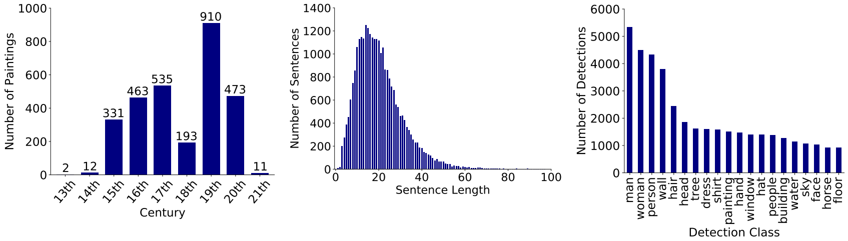


Figure 3.8: Analyses on our Artpedia dataset. From left to right, we report the painting distribution over centuries, the sentence lengths distribution, and the most common detection classes.

With respect to other visual-semantic datasets containing artistic images (reported in Table 3.6), Artpedia provides a larger number of sentences, divided into visual and contextual through a manual annotation procedure. Moreover, to the best of our knowledge, this is the only dataset that contains two types of artistic sentences describing both the visual content of the paintings and other contextual information. For this reason, we devise a visual-semantic model capable of jointly discriminating between visual and contextual sentences of the same painting, and identifying which visual descriptions from a subset of textual elements (*i.e.* a subset of visual descriptions from different paintings) are associated to a specific painting.

To allow the training of our model and foster researches on this domain, we also provide training, validation and test splits obtained by proportionally dividing

Table 3.7: Number of paintings, visual and contextual sentences for each Artpedia split.

	Training	**Validation**	**Test**
Paintings	2,252	339	339
Visual sentences	7,109	1,036	1,028
Contextual sentences	14,822	2,134	2,083

the number of paintings. Splits have been obtained with the constraint of balancing the distributions over centuries and the number of visual sentences to maintain relevant statistics across the subsets. Table 3.7 reports the number of paintings for each split along with the corresponding number of visual and contextual sentences.

3.4.4 Aligning visual and contextual sentences with images

Cross-modal retrieval is characterized by two main tasks: when the query is a textual sentence, the objective is to retrieve the most relevant images, while with an image as a query, the objective is to retrieve the most relevant sentences. The goal is to maximize recall at K, the fraction of queries for which the most relevant item is ranked among the top K retrieved ones. Besides, our setting leverages the presence of visual and contextual sentences, and takes into account this difference when computing the latent alignment within a single page. In the following, we refer to a page as an element of our Artpedia dataset comprising an image and its visual and contextual sentences. Our goal is not only to maximize recall, but also to distinguish the two types of sentences associated to a painting.

In a nutshell, our model firstly maps image regions and sentence words into a joint embedding space. Then, it computes a cross-attention mechanism divided in two branches, where one attends to words with respect to each image region, while the other attends to image regions with respect to each word. This mechanism computes a similarity score for each branch between an image and a sentence. During training, the similarity score is used to minimize two loss functions: our intra-page loss, which strives to rank the sentences associated to a single image, bringing near its visual sentences and pushing away its contextual ones, and the inter-page triplet ranking loss that takes into account all images and their visual sentences as in standard cross-modal retrieval settings.

Similarity function

As mentioned before, the similarity is computed with a cross-attention mechanism that comprises two distinct branches: image-to-text and text-to-image attention, inspired by [186, 342]. Since the two branches are similar, diversified only by the input order, we only describe the first one.

Firstly, given an image I, we extract salient regions such that each of them encodes an object or other entities, and project them into the joint embedding space, obtaining a final set of regions $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}, \boldsymbol{v}_i \in \mathbb{R}^D$. Also, given a sentence T composed of n words, encoded with a word embedding strategy, we project each word into the joint embedding space thus obtaining a vector $\boldsymbol{e}_j \in \mathbb{R}^D$ for each word j. Therefore, given an image I with k detected regions and a sentence T with n words, we compute the similarity matrix for all possible region-word pairs:

$$s_{ij} = \boldsymbol{v}_i^\top \boldsymbol{e}_j \quad i \in [1, k], j \in [1, n] \tag{3.12}$$

where s_{ij} represents the similarity between the region i and the word j. Since region and word features are ℓ_2 normalized, this product corresponds to a cosine similarity.

To attend words with respect to each image region, we compute a sentence-context vector for each region. The sentence-context vector $\boldsymbol{a}_i$ is a weighted representation of the sentence with respect to the region i of the image, where the similarities between the region i and the sentence words are used to weight each word as follows:

$$\boldsymbol{a}_i = \sum_{j=1}^{n} \alpha_{ij} \boldsymbol{e}_j \tag{3.13}$$

where

$$\alpha_{ij} = \frac{\exp\left(\lambda_s s_{ij}\right)}{\sum_{j=1}^{n} \exp\left(\lambda_s s_{ij}\right)} \tag{3.14}$$

and λ_s is a temperature parameter [53].

Finally, to evaluate the similarity of each image region given the sentence-context, we compute the cosine similarity between the attended sentence vector $\boldsymbol{a}_i$ and each image region feature $\boldsymbol{v}_i$:

$$R\left(\boldsymbol{v}_i, \boldsymbol{a}_i\right) = \frac{\boldsymbol{v}_i^\top \boldsymbol{a}_i}{\|\boldsymbol{a}_i\|} \tag{3.15}$$

To summarize the similarity between an image I and a sentence T, we employ average pooling between all image regions and the sentence-context vector:

$$R_{AVG}(I,T) = \frac{\sum_{i=1}^{k} R\left(\boldsymbol{v}_i, \boldsymbol{a}_i\right)}{k} \tag{3.16}$$

Likewise, the other branch follows the same procedure but swapping image regions and sentence words, computing a region-context vector for each sentence word, evaluating their cosine similarities and summarizing the final branch score in the same way. Finally, by averaging the similarity scores of the two branches, we obtain the final similarity score $S(I,T)$ between an image I and a sentence T.

Training

Intra-page loss. With the objective of correctly ranking visual and contextual sentences of a given image, we propose an intra-page loss function that learns the latent alignment between an image and its corresponding visual sentences within a single page of the dataset. Given an image I, a visual sentence T_V and a contextual sentence T_C, our intra-page loss is computed by taking into account the similarity score $S(I,T_V)$ between the image and the visual sentence and the similarity score $S(I,T_C)$ between the image and the contextual one:

$$L_{intra}(I,T_V,T_C) = [\alpha - S(I,T_V) + S(I,T_C)]_+ \tag{3.17}$$

where $[x]_+ = max(x,0)$ and α is the margin. Note that, since this loss function is computed within a single page, both considered visual and contextual sentence are taken within the sentences of the given image I.

Inter-page triplet ranking loss. Since our final objective is not only to identify visual and contextual sentences of the same image, but also to associate matching image-visual sentence pairs within the entire dataset, we define an inter-page triplet ranking loss, which is typical of cross-modal retrieval methods.

As proposed in [85], we focus solely on the hardest negatives in the mini-batch. So that, our final inter-page triplet ranking loss with margin α is defined as follows:

$$L_{inter}(I,T) = \max_{\hat{T}} \left[\alpha - S(I,T) + S(I,\hat{T})\right]_+ + \max_{\hat{I}} \left[\alpha - S(I,T) + S(\hat{I},T)\right]_+ \tag{3.18}$$

where only the hardest negative sentences $\hat{T}$ or hardest negative images $\hat{I}$ for each positive pair $S(I,T)$ are taken into account. In our case, a negative sentence $\hat{T}$ is a visual sentence of another image. Since this loss function aims to associate images and visual sentences of the entire dataset, contextual sentences are only used by our intra-page loss.

Final training objective. The final training loss is obtained by a linear combination of the two loss functions, *i.e.* $L = \lambda_w L_{inter} + (1 - \lambda_w) L_{intra}$, where $\lambda_w \in [0, 1]$ is a parameter that weights the contribution of the two losses. When λ_w is equal to 0, the training procedure only minimizes our intra-page loss, whilst when λ_w is equal to 1, all the attention is given to the inter-page triplet ranking loss.

3.4.5 Experimental evaluation

We experimentally evaluate the effectiveness of our approach by comparing it with different baselines. First, we provide all implementation details used in our experiments.

Implementation details

To encode image regions, we use Faster R-CNN [259] trained on Visual Genome [175, 7], thus obtaining 2048-dimensional feature vectors. For each image, we exploit the top 20 detected regions with the highest class confidence scores. To project regions into the visual-semantic embedding space, we use a fully connected layer with a size of 512.

For the textual counterpart, we compare GloVe [242] with word embeddings learned from scratch. In both cases, the word embedding size is set to 300. Then, with the aim of capturing the semantic context of the sentence, we employ a bi-directional GRU with a size of 512, so that given a sentence with n words, the bi-directional GRU captures the context reading forward from word 1 to n and reading backwards from word n to 1, averaging the two hidden states to obtain the final embedding vector for each word.

To train our model, we use the Adam optimizer with an initial learning rate of 10^{-6} decreased by a factor of 10 after 15 epochs. In all our experiments, we use a batch size of 128 and clip the gradients at 2. Finally, the margin α and the temperature parameter λ_s are respectively set to 0.2 and 6.

Baselines

To evaluate our solution, we build different baselines to quantify both the effectiveness of using a cross-attention model and that of our intra-page loss. To this aim, we first exploit global features to encode images and sentences in place of multiple feature vectors for each image or sentence. In particular, to encode images, we extract 2048-dimensional feature vectors from the average pooling layer of a ResNet-152, while, to encode sentences, we feed word embeddings through a bi-directional GRU network and average the outputs of the last hidden state in both directions. After projecting both images and sentences into a common embedding space, the final similarity score between an image and a sentence is given by the cosine similarity between the two ℓ_2-normalized embedding vectors.

Furthermore, we compare the proposed intra-page loss function with respect to binary cross-entropy. Therefore, visual and contextual sentences are not projected into the same embedding space, but fed through a binary classification branch. In practice, each sentence is classified either as visual or contextual by concatenating the image and sentence embeddings and feeding them through two fully connected layers of size 512 and 1, respectively. For the cross-attention model, the image embedding is obtained by averaging the image region embedding vectors, while the sentence embedding is obtained by averaging the last hidden states of the bi-directional GRU in the two directions.

For both baselines, all other hyper-parameters and training details are the same as those used in our complete model.

Cross-modal retrieval results

We first evaluate the effectiveness of our model to identify and distinguish visual sentences with respect to contextual ones. Table 3.8 shows the results on the Artpedia test set in terms of average precision (AP). In particular, the results are obtained by training the models with λ_w equal to 0 (*i.e.* by only minimizing the intra-page loss or binary cross-entropy). As it can be seen, our intra-page loss function always obtains better performance with respect to the binary cross-entropy baseline either when exploiting global features to embed images and sentences or when using the cross-attention approach described in Section 3.5.2. Regarding the word embedding strategy, GloVe vectors achieve better results with respect to word embeddings learned from scratch, probably due to the presence of peculiar words, typical of the artistic domain.

In Table 3.9, we show the performance of our complete model trained with various λ_w weights to differently balance the contribution of the two loss functions. In this case, the goal is not only to correctly distinguish between visual and contextual sentences of a given image, but also to find the corresponding visual sentences from a subset of other textual elements (*i.e.* visual sentences of different

Table 3.8: Intra-page results in terms of Average Precision (AP).

Model	Word Embedding	AP
Global features with BCE loss	Learned	39.3
Global features with BCE loss	GloVe	40.8
Global features with intra-page loss	Learned	52.8
Global features with intra-page loss	GloVe	**55.3**
Cross-attention with BCE loss	Learned	42.6
Cross-attention with BCE loss	GloVe	41.7
Cross-attention with intra-page loss	Learned	86.3
Cross-attention with intra-page loss	GloVe	**88.5**

images). Results are reported in terms of recall@K ($K = 1, 5$) using a different number N of items from which perform retrieval. In details, given an image as a query, the retrieval of a textual element is performed from a subset of visual sentences of N different images (*i.e.* the visual sentences of the query and those of other $N - 1$ randomly selected images). Instead, given a textual query, the retrieval of an image is performed from a subset of N different images (*i.e.* the image linked to the query and other $N - 1$ randomly selected images from the Artpedia test set). We also report the results of identifying visual sentences with respect to contextual ones in terms of average precision. As it can be noticed, by increasing the λ_w weight, we obtain an increment of recall metrics with a slight drop of average precision values, in almost all considered combinations of features and word embeddings. Also in this case, the cross-attention mechanism and the GloVe word embeddings achieve better results than global features and learned word embeddings.

Finally, Figure 3.11 shows learned embedding spaces using the best model (*i.e.* cross-attention with GloVe word embeddings) using different λ_w weights. Since in this case images and sentences are composed of an embedding vector for each image region and word of the sentence, we represent each image or sentence by summing the ℓ_2-normalized embedding vectors of its image regions or words, and ℓ_2-normalized again the result. This strategy has been largely used in image and video retrieval works, and is known for preserving the information of the original vectors into a compact representation with fixed dimensionality [297]. To get a suitable two-dimensional representation out of a 512-dimensional space, we run the t-SNE algorithm [218], which iteratively finds a non-linear projection which preserves pairwise distances from the original space. As it can be observed, the higher the λ_w weight, the greater the distance between images and visual sentences in the embedding space, thus confirming the drop of average precision

Table 3.9: Cross-modal retrieval results with a different number N of retrievable items and with respect to different λ_w weights.

				$N = 10$				$N = 50$				$N = 100$			
Model	**Word Emb.**	λ_w	**AP**	Img-to-Text		Text-to-Img		Img-to-Text		Text-to-Img		Img-to-Text		Text-to-Img	
				R@1	**R@5**	**R@1**	**R@5**	**R@1**	**R@5**	**R@1**	**R@5**	**R@1**	**R@5**	**R@1**	**R@5**
Global	Learned	0.25	44.9	9.4	36.3	7.6	50.0	2.4	8.3	1.3	8.9	0.6	5.0	0.5	3.6
Global	GloVe		43.1	12.4	35.4	8.5	48.7	2.7	9.1	2.2	11.1	0.6	5.3	1.8	5.5
X-Attn	Learned		85.9	15.3	40.4	17.5	61.9	2.7	13.3	4.2	16.7	2.1	8.0	2.2	9.0
X-Attn	GloVe		**88.2**	**19.8**	**44.0**	**22.7**	**69.6**	**8.6**	**22.1**	**6.1**	**23.6**	**4.4**	**15.9**	**4.0**	**14.8**
Global	Learned	0.50	50.2	9.4	38.1	9.9	50.7	1.8	10.0	2.0	10.5	0.6	6.2	1.1	5.1
Global	GloVe		46.0	8.8	37.2	9.8	48.9	1.2	10.0	2.0	10.1	1.8	4.1	1.0	4.4
X-Attn	Learned		85.2	11.5	40.1	17.4	61.0	3.2	13.6	3.8	18.7	1.2	7.7	2.4	9.9
X-Attn	GloVe		**87.5**	**26.3**	**54.3**	**21.2**	**69.7**	**8.8**	**27.7**	**7.5**	**22.9**	**6.2**	**18.6**	**4.1**	**14.1**
Global	Learned	0.75	53.4	10.6	38.3	10.4	50.0	2.4	10.6	2.3	11.6	1.5	5.6	1.4	6.2
Global	GloVe		44.9	10.9	34.2	8.9	47.7	1.8	8.6	1.8	9.3	0.9	4.4	0.7	4.6
X-Attn	Learned		84.6	10.9	37.5	18.5	64.3	2.7	10.0	5.1	20.1	1.2	7.1	2.9	11.4
X-Attn	GloVe		**86.5**	**29.5**	**57.2**	**23.7**	**71.2**	**13.6**	**31.9**	**5.8**	**23.1**	**8.6**	**22.7**	**4.1**	**13.6**

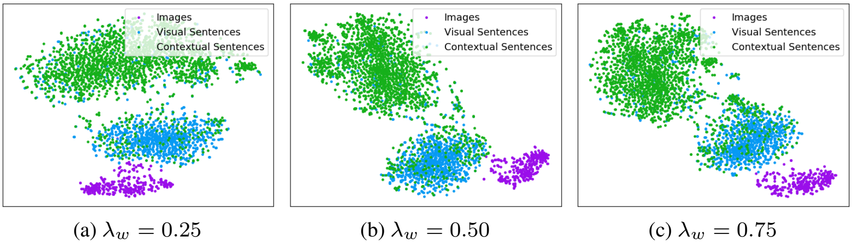
values when decreasing the importance of our intra-page loss during training.

3.5 Aligning Digital Humanities

As shown in the previous section, research efforts have resulted in algorithms that can retrieve images from textual descriptions and vice versa, when paired training data is provided. However, the domain of the Digital Humanities features complex visual and semantic structures and also a significant lack of training data, which makes the use of fully-supervised approaches often infeasible. In this section, we go beyond these limitations and tackle the design of visual-semantic algorithms proposing a joint visual-semantic embedding that can automatically align illustrations and textual elements without paired supervision. This is achieved by transferring the knowledge learned on ordinary visual-semantic datasets to the artistic domain. Experiments, performed on two datasets specifically designed for this domain, validate the proposed strategies and quantify the domain shift between natural images and artworks.

3.5.1 Introduction

As humans, we can easily link our ability to see and understand the surrounding environment with the ability to express ourselves in natural language. In the effort of artificially replicating these connections, new models have emerged for image and video captioning [7, 215, 57] and for visual-semantic retrieval [172, 85, 186]. The former architectures combine vision and language in a generative flavor on



(a) $\lambda_w = 0.25$ (b) $\lambda_w = 0.50$ (c) $\lambda_w = 0.75$

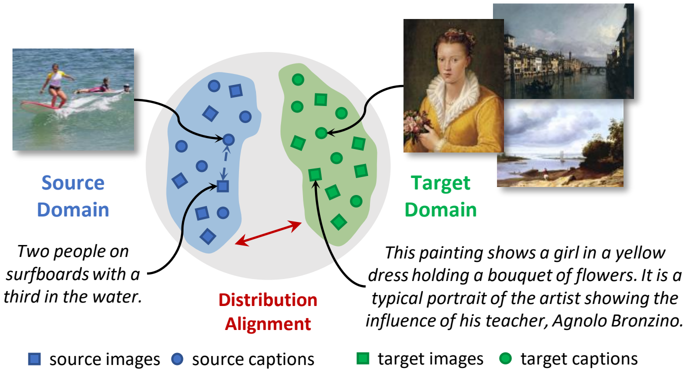
Figure 3.9: Comparison between visual-semantic embedding spaces obtained by training the model with different λ_w weights. Visualizations are obtained by running the t-SNE algorithm [218] on top of embedding vectors representing images and sentences (both visual and contextual).

the textual side, the latter build common spaces to integrate the two domains and retrieve textual elements given visual queries, and vice versa.

The leading solutions for visual-semantic retrieval have so far relied on fully supervised settings in which paired training samples are available and have been applied to general-purpose datasets where the state of the art of concept recognition methods is useful and well assessed. In the domain of arts and culture, however, both visual and textual elements are far from those of ordinary datasets. On one side, textual descriptions often contain technical language with symbolic reminds, metaphors and artistic or historical connections; on the other side, artworks and illustrations are characterized by visual features different from those of natural images. Beyond this domain-shift issue, the supervised training of a common visual-semantic embedding requires sufficiently large datasets. Instead, the artistic domain is often characterized by small scale datasets in which the pairing between visual and textual elements is not available or expensive to obtain.

Tackling the aforementioned setting, in this section we propose a semi-supervised visual-semantic embedding model (`SS-VSE`) for cross-modal retrieval in the artistic domain. Our approach relies on the construction of a common semantic embedding, in which the knowledge learned on a supervised and ordinary visual-semantic dataset is transferred to an artistic dataset in which the pairing between images and sentences is not available. After using global feature vectors, we also investigate the use of auto-encoders (`SS-VSE-AE`) to obtain more compact representations of input images and sentences. Experiments are conducted on two datasets specifically designed for the artistic domain. In particular, we use the

Figure 3.10: Visual and textual data from the artistic domain are different from those addressed by ordinary visual-semantic datasets, posing significant challenges in the automatic understanding of arts and culture. Our approach can align illustrations and textual elements by transferring the knowledge learned on standard datasets to match images and captions coming from a target domain.


Source Domain
Two people on surfboards with a third in the water.
Distribution Alignment
Target Domain
This painting shows a girl in a yellow dress holding a bouquet of flowers. It is a typical portrait of the artist showing the influence of his teacher, Agnolo Bronzino.
🟦 source images
🔵 source captions
🟩 target images
🟢 target captions

BibleVSA dataset [20] which contains illustrations and textual sentences extracted from the commentaries of a historical manuscript, and the SemArt dataset [100] that is composed of artwork images and textual comments. Extensive experiments are presented to validate the proposed solution and to visualize the effect of the knowledge transfer between source and target datasets.

Motivation

Only a few works have applied image-text matching strategies to artistic data. Among them, [100] used additional metadata such as title, author, genre, and period of the paintings to find corresponding image-text pairs. [285] introduced a new dataset and a visual-semantic model to discriminate visual and contextual sentences associated to artistic images and, at the same time, to align the corresponding visual and textual elements. While [100, 285] matched images and textual descriptions in a supervised way, [20, 36] addressed the problem in a *semi-supervised* setting, adapting the knowledge learned on a given source domain to align images and text belonging to a different target domain and without directly training the model on the target domain. This solution, which is known as *domain adaptation*, has been used in a wide variety of applications such as image classification [211], semantic segmentation [129, 51], object detection [143, 50], and image captioning [46, 345]. Typically, it is addressed by minimizing the distance between feature space statistics of the source and target, or by using domain adversarial objectives where a domain classifier is trained to distinguish between the source and target representations.

3.5.2 Semi-supervised cross-modal retrieval

In the following, we describe our strategy for cross-modal retrieval in the artistic domain. Our model has a two-fold role: retrieving relevant images given textual sentences as queries, and retrieve relevant sentences when given images as queries. Parameters of the model are learned with the objective of maximizing recall at K – *i.e.* the fraction of queries for which the most relevant item is ranked among the top K retrieved ones. As training data in the artistic domain is often scarce, we build a proposal that does not need a paired training set in which the associations between images and sentences are known in advance. Rather, our model transfers the knowledge learned on a source annotated dataset to a target dataset in which the pairing between the two modalities is unknown at training time.

In a nutshell, the paradigm of the common embedding space is exploited to learn similarities between images and sentences. In addition to using global feature vectors to encode data from both modalities, we also investigate the use of auto-encoders to learn more compact representations of images and sentences. To transfer knowledge to the artistic domain without leveraging annotated pairs, we devise a distribution alignment strategy based on the Maximum Mean Discrepancy measure, which aims at uncovering suitable cross-modal representation of cultural heritage data without supervision.

Visual-semantic embeddings

Aligning works of arts and their corresponding textual descriptions requires the ability to compare visual and textual data in this particular domain. To this end, we adopt the strategy of creating a shared multi-modal embedding space, in which both textual and visual elements can be projected and compared using a similarity function.

Formally, we denote $\phi(I, \mathbf{w}_\phi) \in \mathbb{R}^{D_\phi}$ as the feature representation computed from an image I of the dataset (such as the representation coming from a CNN), and $\psi(T, \mathbf{w}_\psi) \in \mathbb{R}^{D_\psi}$ as the representation of a textual element T, computed, for example, using a text encoder on one-hot vectors, or as a function of pre-trained word embeddings. Here, $\mathbf{w}_\phi$ and $\mathbf{w}_\psi$ indicate, respectively, the learnable weights of the visual and textual encoders.

To project those representations into a common semantic space, we perform a linear projection followed by a ℓ_2-normalization step, so that the resulting embedding space lies on the ℓ_2 unit ball:

$$f(I, \mathbf{w}_f, \mathbf{w}_\phi) = \ell_{2,norm}(\mathbf{w}_i^\intercal \phi(I, \mathbf{w}_\phi)) \tag{3.19}$$

$$g(T, \mathbf{w}_g, \mathbf{w}_\psi) = \ell_{2,norm}(\mathbf{w}_c^\intercal \psi(T, \mathbf{w}_\psi)), \tag{3.20}$$

where $\ell_{2,norm}$ is the ℓ_2 normalization function. Being D the dimensionality of the joint embedding space, $\mathbf{w}_f$ is a $D_\phi \times D$ matrix, and $\mathbf{w}_g$ is a $D_\psi \times D$ matrix.

Visual and textual elements can be compared in the joint multi-modal embedding space by computing the cosine similarity (equivalent, in this case, to a dot product) between their projections, so that the similarity between an image I and a caption T becomes

$$s(I, T) = f(I, \mathbf{w}_f, \mathbf{w}_\phi) \cdot g(T, \mathbf{w}_g, \mathbf{w}_\psi). \tag{3.21}$$

Clearly, the utility of the joint embedding space is maximized when it exhibits suitable cross-modality matching properties, *i.e.* when similarities in the embedding space correspond to meaningful similarities in both modalities. In this case, the embedding space acts as a bridge between the two modalities and makes it possible to retrieve textual pieces describing a query image, and images described by a query caption by identifying the closest neighbors in both modalities.

Given a dataset annotated with matching visual-semantic pairs, a good proxy of this property is to verify that corresponding pairs are neighbours in the embedding space. As a matter of fact, classical approaches have relied on the availability of paired datasets, and have learned the joint embedding for a specific domain in a completely supervised way, *e.g.* training the parameters of the model according to a Hinge triplet ranking loss with margin, which imposes suitable similarities between matching and non-matching elements. Formally, it is defined as:

$$\begin{aligned}\ell(I, T) = &\sum_{\hat{T}} \left[\alpha - s(I, T) + s(I, \hat{T})\right]_+ + \\ &+ \sum_{\hat{I}} \left[\alpha - s(I, T) + s(\hat{I}, T)\right]_+\end{aligned} \tag{3.22}$$

where $[x]_+ = \max(0, x)$ and α is a margin. In the equation above, (I, T) is a matching image-text pair (*i.e.*, such that T describes the content of I, and I

represents the content of T), while $\hat{T}$ is a negative text with respect to I (such that $\hat{T}$ does not describe I), and $\hat{I}$ is a negative image with respect to T (such that T does not describe $\hat{I}$). The terms contained in both sums require that the difference in similarity between the matching and the non-matching pair is higher than a margin α: in the first sum, this is done by considering an image anchor and matching or non-matching captions; in the latter, instead, a caption is used as anchor.

As reported by a recent work by [85], in a completely supervised setting it is often beneficial to replace the sums in Eq. 3.22 with maximum operations, so to consider only the most violating non-matching pair.

Auto-encoding images and sentences

In addition to the use of plain global feature vectors, we also investigate an alternative projection strategy in which images and sentences are fed to an auto-encoder to learn a more compact yet powerful representation of the input, which can in turn be used as the input of the projection function defined in Eq. 3.19.

To this end, we design a textual auto-encoder which can convert variable-length captions to fixed-length representations from which input sentences can be reconstructed. In particular, our model exploits Gated Recurrent Networks (GRUs) [52] for both encoding and decoding. Formally, given a sentence $T = (w_1, w_2, ..., w_N)$ with length N, we firstly encode it word by word through a single-layer GRU and take the last hidden state of the Recurrent layer as the encoding of the sentence. Given the recurrent relation defined by the GRU cell and the t-h word, *i.e.*

$$\mathbf{h}_t = \text{GRU}_e(w_t, \mathbf{h}_{t-1}), \tag{3.23}$$

the encoding of the input sentence is defined as:

$$\mathbf{h}_N = \text{GRU}_e(w_N, \mathbf{h}_{N-1}). \tag{3.24}$$

In the decoding stage, the input sentence is reconstructed by feeding $\mathbf{h}_N$ to a second GRU layer which is in charge of generating the reconstructed sentence. During training, at the t-th iteration the Recurrent layer is fed with $\mathbf{h}_N$ and the previous ground-truth words, and it is trained to predict the t-h word. Formally, the training objective is thus:

$$\max_{\mathbf{w}} \sum_{t=1}^{T} \log \Pr(w_t | w_{t-1}, w_{t-2}, ..., w_1, \mathbf{h}_N). \tag{3.25}$$

The probability of a word is modeled via a softmax layer applied to the output of the decoder. To reduce the dimensionality of the decoder, a linear embedding transformation is used to project one-hot word vectors into the input space of the decoder and, vice-versa, to project the output of the decoder to the dictionary space.

Given the auto-encoder for the textual part, we build an encoder-decoder model that can take an image feature vector as input and reconstruct it starting from an intermediate and more compact representation. In practice, the encoder model is composed of a single fully connected layer. We indeed notice that a single layer leads to have a fairly informative representation of the image feature vector. Formally, we define the output of the encoder model $\mathbf{z}$ (*i.e.* the intermediate representation of the input image) as

$$\mathbf{z} = \tanh(\mathbf{W_e}\phi(I) + b_e), \tag{3.26}$$

where $\mathbf{W_e}$ and b_e are, respectively, the weight matrix and the bias vector of the encoder. Notice that the output of the encoder layer is fed through a $\tanh$ non-linearity activation function.

The decoder model has a symmetric structure. Therefore, starting from the intermediate vector $\mathbf{z}$, the decoder applies a single fully connected layer that transforms $\mathbf{z}$ to the size of the input image feature vector. Formally, the reconstructed image feature vector $\hat{\phi}(I)$ is defined as

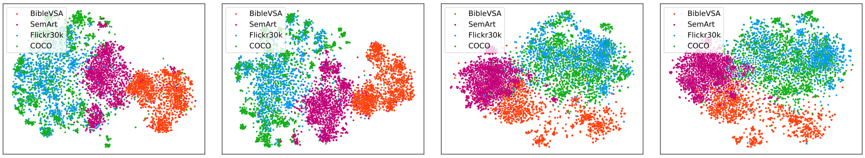
$$\hat{\phi}(I) = \mathbf{W_d}\mathbf{z}_i + b_d, \tag{3.27}$$

where $\mathbf{W_d}$ and b_d are the weight matrix and the bias vector of the decoder. Overall, the image auto-encoder is trained to minimize the reconstruction error for each input image. We define the decoder loss function as the mean square error between the original image feature vector $\phi(I)$ and the corresponding reconstruction $\hat{\phi}(I)$.

Aligning distributions

While the knowledge of matching and non-matching pairs on a source dataset can be exploited to train the embedding space, as discussed in Sec. 3.5.2, the two reconstruction losses can be applied to both the source and the target dataset, thus building encoded representations which are suitable for both datasets. However, this is not enough to transfer knowledge from the source domain to the target domain, as there is no guarantee that encoded words and sentences from the target dataset will lie together in the embedding space.

Figure 3.11: Comparison between the visual and textual features of ordinary visual-semantic datasets (Flickr30k, COCO) and those of BibleVSA and SemArt dataset. Visualization is obtained by running the t-SNE algorithm on top of the features. Best seen in color.



(a) VGG-19 (b) ResNet-152 (c) GloVe (d) FastText

To this end, we match the distributions of textual and visual data in the target domain, while learning from pairs sampled from the source domain. Following recent works in the field [141, 303, 343], we use the Maximum Mean Discrepancy (MMD) to compare distributions. This, basically, computes the distance between the expectations of the two distributions in a reproducing kernel Hilbert space $\mathcal{H}_\kappa$ endowed with a kernel κ, and can be used as an additional loss term:

$$\mathcal{L}_{mmd} = \|\mathbf{E}_{I\sim\mathcal{I}}\left[f(I)\right] - \mathbf{E}_{T\sim\mathcal{T}}\left[g(T)\right]\|^2_{\mathcal{H}_\kappa}, \tag{3.28}$$

where $\mathcal{I}$ is the distribution of the illustrations, and $\mathcal{T}$ is the distribution of captions. The kernel in the MMD criterion must be a universal kernel, and thus we empirically choose a Gaussian kernel:

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\sigma\|\mathbf{x} - \mathbf{y}\|^2\right). \tag{3.29}$$

At training time, we sample two mini-batches of samples, one from the supervised set and a second one from the unsupervised dataset. The back-propagated loss is then the sum of the supervised loss (Eq. 3.22) on the supervised set, plus the MMD loss $\mathcal{L}_{mmd}$ approximated over the batch from the unsupervised set. Additionally, the two loss terms of the auto-encoders are evaluated over both the supervised and the unsupervised batches.

3.5.3 Experimental evaluation

Datasets

We perform experiments on two different visual-semantic datasets containing artistic images and corresponding textual descriptions (described below). As source domains, we use Flickr30k and COCO which are composed of natural images and are commonly used to train cross-modal retrieval methods. For these two datasets, we use the splits provided by [159].

BibleVSA [20]. The dataset consists of $2,282$ illustrations taken from the digitized version of the Borso d'Este Holy Bible, one of the most significant illustrated manuscripts of Renaissance. Each image is associated with a single textual phrase extracted from a textual commentary which describes the content of each page of the manuscript. In our experiments, we use the original training, validation, and test split, respectively composed of $1,671$, 293, and 307 image-caption pairs.

SemArt [100]. This dataset is composed of $21,384$ paintings extracted from the Web Gallery of Art, which contains European fine-art reproductions between the 8th and the 19th century. Each image is associated to an artistic comment and to a set of 7 different attributes comprising the title, the author, and the type of the painting. Overall, the dataset is divided in training, validation and test split with $19,244$, $1,069$ and $1,069$ elements, respectively. The average length of each artistic comment is more than 80, with a maximum number of words equal to 830. This highlights the difference between SemArt and ordinary visual-semantic datasets (*i.e.* COCO has an average caption length lower than 11) and accentuates the challenges of this set of data. To first validate our solution in a less complex scenario, we limit the validation and test set to 300 randomly selected image-text pairs. Then, we evaluate our model using a different number of retrievable items.

Implementation details

To encode input images, we use two different convolutional networks: the VGG-19 [280] and ResNet-152 [119]. We extract image features from the *fc7* layer of the VGG-19 and from the average pooling layer of the ResNet-152 thus obtaining an input image embedding dimensionality D_ϕ of 4096 and 2048, respectively.

For encoding image descriptions, we use a GRU network [52]. We set the dimensionality of the GRU and of the joint embedding space D to 512, while the input size of word embeddings D_ψ is set to 300. We use either a text encoder on one-hot vectors or different pre-trained word embeddings (such as GloVe [242] and FastText [28]) as input of the GRU.

The model with textual and visual auto-encoders is trained using the same input and output sizes. For the training with pre-trained word embeddings, instead of using the loss function defined in Eq. 3.25, we compute the cosine distance between original and reconstructed embeddings of each word.

Table 3.10: Semi-supervised cross-modal retrieval results using different visual features. Results are reported on BibleVSA and SemArt test set.

Method	CNN Feat.	Text Retrieval			Image Retrieval		
		R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
		COCO → BibleVSA					
SS-VSE	VGG-19	**13.1**	29.5	36.1	3.9	16.7	27.5
SS-VSE	ResNet-152	9.8	**31.1**	**50.8**	**6.2**	**22.3**	**30.8**
SS-VSE-AE	VGG-19	**9.8**	**27.9**	34.4	**3.6**	15.7	25.9
SS-VSE-AE	ResNet-152	6.6	23.0	**36.1**	**3.6**	**19.7**	**29.8**
		COCO → SemArt					
SS-VSE	VGG-19	3.7	11.7	19.0	2.3	10.0	19.3
SS-VSE	ResNet-152	**6.7**	**19.3**	**27.0**	**5.0**	**17.3**	**29.3**
SS-VSE-AE	VGG-19	**5.0**	**14.3**	**22.7**	1.7	9.0	15.3
SS-VSE-AE	ResNet-152	4.7	12.7	21.0	**3.7**	**11.0**	**18.0**

All experiments are performed by using Adam optimizer with a learning rate of 0.0002 for 15 epochs and then decreased by a factor of 10. We set the margin α to 0.2, the σ parameter of the Gaussian kernel to 1 and the size of the mini-batch to 128.

Analysis of artistic visual-semantic data

To get an insight of characteristics of the BibleVSA and SemArt datasets, we analyze the distribution of image and textual features respectively obtained from CNNs and sentence embeddings and compare them with those extracted from classical visual-semantic datasets.

For the visual part, we extract the activation from the VGG-19 and ResNet-152 networks, while, for textual elements, we embed each word of a caption with a word embedding strategy (either GloVe or FastText). To get a feature vector for a sentence, we sum the ℓ_2 normalized embeddings of the words, and we apply the ℓ_2-norm also to the results. This strategy is largely used in image and video retrieval literature and is known for preserving the information of the original vectors into a compact representation with fixed dimensionality [297] .

Fig. 3.11 shows the distributions of visual and textual features of both datasets. To get a suitable two-dimensional representation, we run the t-SNE algorithm [218], which iteratively finds a non-linear projection that preserves the statistical distribution of the pairwise distances from the original space. As it can be observed, the features of ordinary visual-semantic datasets share almost the

Table 3.11: Semi-supervised cross-modal retrieval results using different word embeddings. Results are reported on BibleVSA and SemArt test set.

Method	Word Emb.	Text Retrieval			Image Retrieval		
		R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
		COCO → BibleVSA					
SS-VSE	FastText	8.2	19.7	34.4	2.6	16.7	26.6
SS-VSE	GloVe	6.6	23.0	39.3	3.6	16.7	27.2
SS-VSE	-	**9.8**	**31.1**	**50.8**	**6.2**	**22.3**	**30.8**
SS-VSE-AE	FastText	**6.6**	**27.9**	34.4	3.3	14.4	25.2
SS-VSE-AE	GloVe	4.9	19.7	**41.0**	**3.9**	13.8	27.5
SS-VSE-AE	-	**6.6**	23.0	36.1	3.6	**19.7**	**29.8**
		COCO → SemArt					
SS-VSE	FastText	1.7	5.0	7.7	0.7	2.3	7.3
SS-VSE	GloVe	3.3	11.3	16.0	2.0	11.0	17.7
SS-VSE	-	**6.7**	**19.3**	**27.0**	**5.0**	**17.3**	**29.3**
SS-VSE-AE	FastText	3.7	10.0	17.0	3.0	9.3	11.7
SS-VSE-AE	GloVe	2.7	12.0	17.0	1.7	7.0	12.3
SS-VSE-AE	-	**4.7**	**12.7**	**21.0**	**3.7**	**11.0**	**18.0**

same visual and textual distributions. BibleVSA and SemArt, on the contrary, feature a completely different distribution, according to both modalities and all feature extractors. This underlines, on the one hand, that artistic datasets define a completely new domain. On the other hand, instead, this motivates the low performance of existing models when tested on these datasets.

Table 3.12: Semi-supervised cross-modal retrieval results on SemArt test set using a different number N of retrievable items.

Method	$N = 100$						$N = 300$						$N = 500$						$N = 1000$					
	Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval			Text Retrieval			Image Retrieval		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
	Flickr30k → SemArt																							
VSE	2.0	10.0	14.0	3.0	11.0	17.0	1.7	5.7	8.7	1.0	5.3	7.3	0.8	2.6	6.0	0.4	3.6	5.6	0.5	1.6	2.8	0.1	1.2	2.8
SS-VSE	**7.0**	**23.0**	**40.0**	**10.0**	**23.0**	**37.0**	**5.0**	**15.3**	**22.0**	**3.7**	**13.3**	**17.7**	**3.6**	**9.6**	**14.6**	**1.8**	**7.6**	**12.0**	**1.5**	**6.2**	**10.0**	**1.2**	**3.5**	**7.4**
VSE-AE	3.0	9.0	15.0	5.0	12.0	19.0	2.3	6.0	7.3	0.7	6.0	9.0	1.2	4.2	6.4	0.8	3.0	5.4	0.5	2.2	4.1	0.5	1.6	3.1
SS-VSE-AE	**6.0**	**28.0**	**42.0**	**6.0**	**18.0**	**30.0**	**4.0**	**12.7**	**20.0**	**2.3**	**10.0**	**16.3**	**1.8**	**9.2**	**14.8**	**1.6**	**6.0**	**11.4**	**1.0**	**5.6**	**9.4**	**0.6**	**3.4**	**6.8**
	COCO → SemArt																							
VSE	5.0	13.0	21.0	3.0	8.0	19.0	1.7	8.7	15.3	1.0	8.0	12.3	1.2	3.6	6.4	1.6	3.4	6.0	1.0	2.7	3.6	0.5	2.3	3.6
SS-VSE	**16.0**	**34.0**	**52.0**	**12.0**	**32.0**	**48.0**	**6.7**	**19.3**	**27.0**	**5.0**	**17.3**	**29.3**	**3.8**	**12.2**	**19.8**	**3.4**	**11.6**	**19.4**	**2.7**	**8.9**	**14.0**	**2.3**	**6.9**	**12.9**
VSE-AE	6.0	15.0	20.0	3.0	11.0	22.0	3.0	7.3	11.7	0.3	3.7	6.7	1.6	4.0	6.2	1.2	2.8	4.0	0.8	2.6	4.0	0.8	1.6	2.3
SS-VSE-AE	**7.0**	**24.0**	**39.0**	**6.0**	**17.0**	**26.0**	**4.7**	**12.7**	**21.0**	**3.7**	**11.0**	**18.0**	**2.0**	**10.0**	**15.8**	**2.2**	**5.0**	**10.8**	**0.9**	**6.1**	**10.0**	**1.0**	**3.8**	**5.8**

Table 3.13: Semi-supervised retrieval results on BibleVSA test set.

Method	Text Retrieval			Image Retrieval		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
	Flickr30k → BibleVSA					
`VSE`	3.3	8.2	16.4	1.6	12.1	19.7
`SS-VSE`	**9.8**	**23.0**	**39.3**	**4.6**	**16.1**	**26.6**
`VSE-AE`	1.6	4.9	13.1	3.0	9.8	17.0
`SS-VSE-AE`	**3.3**	**23.0**	**29.5**	**3.3**	**13.1**	**23.0**
	COCO → BibleVSA					
`VSE`	1.6	9.8	16.4	2.6	10.5	20.0
`SS-VSE`	**9.8**	**31.1**	**50.8**	**6.2**	**22.3**	**30.8**
`VSE-AE`	3.3	6.6	14.8	1.6	9.8	19.7
`SS-VSE-AE`	**6.6**	**23.0**	**36.1**	**3.6**	**19.7**	**29.8**

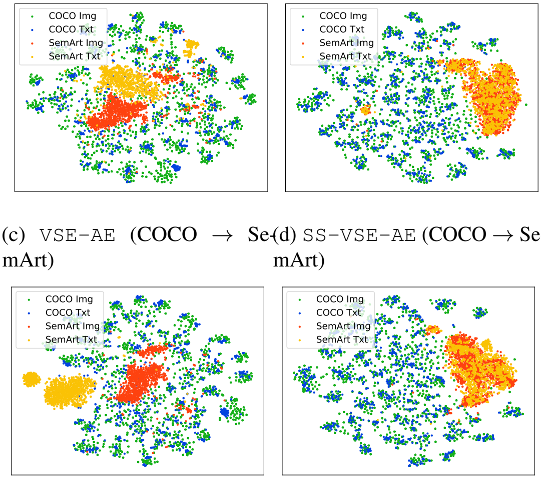
Cross-modal retrieval results

To evaluate the effectiveness of the visual-semantic embeddings, we report rank-based performance metrics $R@K$ ($K = 1, 5, 10$) for image and caption retrieval. In particular, $R@K$ computes the percentage of test images or test sentences for which at least one correct result is found among the top-K retrieved sentences, in the case of caption retrieval, or the top-K retrieved images, in the case of image retrieval.

Firstly, we assess the performance of our full model when using different CNN features or different word embeddings, to get an insight of the role of different global feature vectors. In Table 3.10, we show the performance of the proposed approach on the test sets of BibleVSA and SemArt when using image features extracted, respectively, from VGG-19 and ResNet-152. Table 3.11 compares the use of FastText and GloVe embeddings versus a learned word embedding matrix. In this case, the results on SemArt test set are obtained by using 300 randomly selected retrievable items.

For space reasons, we limit this analysis to a single source dataset (namely, COCO), as we have observed similar behaviours on Flickr30k. The two variants of our approach are denoted as `SS-VSE` and `SS-VSE-AE`, where the first refers to the model with global feature vectors and linear projection, and the latter refers to the model with the visual and textual auto-encoder. As it can be observed, the global descriptor extracted from ResNet-152 outperforms the one extracted from VGG-19 in almost all settings. Noticeably, learned word embeddings outperform pre-trained solutions. We speculate that this performance drop is due to the the

Figure 3.12: Comparison between t-SNE projections of the embedding spaces learned with (b-d) and without (a-c) the MMD loss. Best seen in color.



(a) VSE (COCO → SemArt)

(b) SS-VSE (COCO → SemArt)

(c) VSE-AE (COCO → SemArt)

(d) SS-VSE-AE (COCO → SemArt)

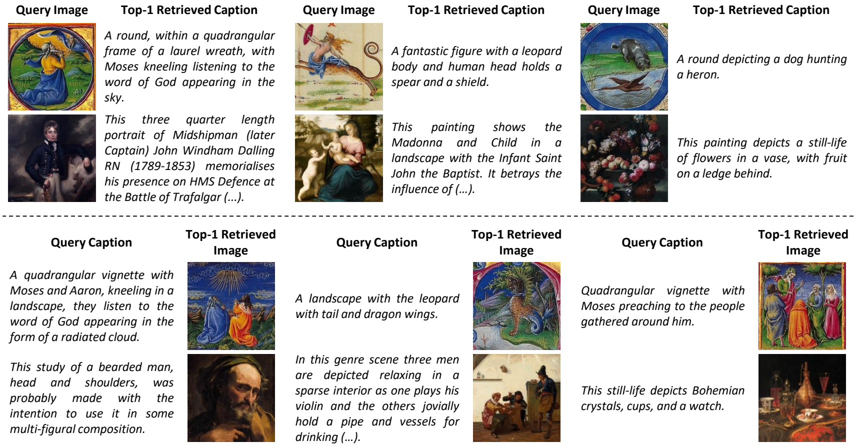
highly specialized nature of the target datasets. In this regards, word embeddings seem to offer a poor initialization point with respect to a from-scratch learning of the word embedding matrix.

Another interesting consideration is that the use of hard negatives in the triples loss function is typically beneficial in a supervised setting [85]. Instead, in our semi-supervised setting, we do not report the same advantages in improving the alignment of the target domain.

Evaluation of semi-supervised embeddings

In Tables 3.12 and 3.13, we compare the performances of the two proposed semi-supervised approaches (SS-VSE and SS-VSE-AE) on SemArt and BibleVSA test set with respect to the two models trained without the distribution alignment (VSE and VSE-AE). For these experiments, we use global feature vectors extracted from ResNet-152 and learned word embeddings. Given the significant size of SemArt dataset, we report retrieval results when using different sets of database

Figure 3.13: Qualitative image-to-text (upper) and text-to-image (lower) results on BibleVSA (first and third rows) and SemArt (second and fourth rows) dataset, using the proposed semi-supervised strategy.



A round, within a quadrangular frame of a laurel wreath, with Moses kneeling listening to the word of God appearing in the sky.

A fantastic figure with a leopard body and human head holds a spear and a shield.

A round depicting a dog hunting a heron.

This three quarter length portrait of Midshipman (later Captain) John Windham Dalling RN (1789-1853) memorialises his presence on HMS Defence at the Battle of Trafalgar (...).

This painting shows the Madonna and Child in a landscape with the Infant Saint John the Baptist. It betrays the influence of (...).

This painting depicts a still-life of flowers in a vase, with fruit on a ledge behind.

A quadrangular vignette with Moses and Aaron, kneeling in a landscape, they listen to the word of God appearing in the form of a radiated cloud.

A landscape with the leopard with tail and dragon wings.

Quadrangular vignette with Moses preaching to the people gathered around him.

This study of a bearded man, head and shoulders, was probably made with the intention to use it in some multi-figural composition.

In this genre scene three men are depicted relaxing in a sparse interior as one plays his violin and the others jovially hold a pipe and vessels for drinking (...).

This still-life depicts Bohemian crystals, cups, and a watch.

items (*i.e.* 100, 300, 500, 1000). We notice that, when using a medium-scale source dataset like Flickr30k, the use of the auto-encoder is competitive with the use of a linear projection of the global feature vector. Instead, when transferring from a large-scale dataset like COCO, the reconstruction term is not needed and the reduced size of the representation degrades the performance. In all settings, the MMD loss gives a significant contribution to the final performance thus confirming the effectiveness of our distribution alignment strategy.

To get a better understanding of the role of the MMD loss, we also show the learned multi-modal embedding space by using t-SNE visualizations. Figure 3.12 shows the embedding spaces when transferring from COCO to SemArt, with and without the MMD loss. As it can be noticed, without the MMD loss the distribution of textual and visual elements on the target domain remains almost separate, as the learning signal from the source domain is not general enough on the target domain. On the contrary, when applying the MMD loss the distribution of the learned image embeddings matches that of the textual counterpart on the target domain, thus confirming the effectiveness of the proposed semi-supervised strategy. Noticeably, the distributions of the source and target domain still remain separate in the embedding space, thus underlying the diverse nature of the two sets.

Finally, Fig. 3.13 reports sample qualitative results on BibleVSA and SemArt dataset. As it can be noticed, our method can retrieve significant elements without employing any paired supervision from the artistic dataset.

Chapter 4

Gene and Protein expression

In the previous chapters, we have shown how it is possible to apply attention mechanism and Transformer architectures to Vision and Language, bridging the gap between different modalities. Thanks to its capabilities, the Transformer model have firstly led a revolution in Natural Language Processing followed by Computer Vision and almost every field of artificial intelligence research. The key ability resides in capturing better long-range interactions among distal elements in data, excelling in dealing with sequences.

Following these premises, in this chapter we explore the application of the attention paradigm to the languages of life: the genetic code and the protein sequences. We propose a new class of deep learning models based on the Perceiver model, built upon Transformer, which exploit asymmetric attention and is able scale to longer sequences. We present a method able to predict the gene expression (mRNA level) given its DNA sequence, and a method predicting the protein expression given its amino-acid sequence. We demonstrate the effectiveness of our methods and promising future opportunities.

4.1 Perceiver for gene and protein expression

The functions of an organism and its biological processes result from the expression of genes and proteins. Therefore quantifying and predicting mRNA and protein

This chapter is related to publications [10] reported in Appendix A, by the author of the thesis. See Appendix A for details.

levels is a crucial aspect of scientific research. Concerning the prediction of mRNA levels, the available approaches use the sequence straddling the Transcription Start Site (TSS) as input to neural networks. The State-of-the-art models (e.g., Xpresso and Basenjii) predict mRNA levels exploiting Convolutional (CNN) or Long Short Term Memory (LSTM) Networks. However, CNN prediction depends on convolutional kernel size, and LSTM suffers from capturing long-range dependencies in the sequence. Concerning the prediction of protein levels, as far as we know, there is no model for predicting protein levels by exploiting the gene or protein sequences.

In the following, we present a new class of models for mRNA and protein level prediction that exploit the Perceiver architecture, which is built upon the Transformer and can attends to long-range interactions in data sequences and, in addition, overcomes the quadratic complexity of the standard Transformer architectures. Specifically, we present: 1. DNAPerceiver model to predict mRNA levels from the sequence straddling the TSS; 2. ProteinPerceiver model to predict protein levels from the protein sequence; 3. Protein&DNAPerceiver model to predict protein levels from TSS-straddling and protein sequences.

We evaluate our models on cell lines, mice, glioblastoma, and lung cancer tissues. The results show the effectiveness of the Perceiver-type models in predicting mRNA and protein levels. In the future, inserting regulatory and epigenetic information into the model could improve mRNA and protein level predictions. The source code is freely available at `https://github.com/MatteoStefanini/DNAPerceiver`

4.1.1 Introduction

Most of the biological processes that regulate the functions of an organism are due to the activity of proteins [67, 66, 313]. In recent decades, the incredible development of sequencing techniques and proteomics quantifications have enabled a systematic analysis of the activity level of thousands of genes and proteins [367, 243]. In addition, it is known that many regulatory and epigenomic processes regulate the expression of mRNAs and proteins [144, 26, 84], and the sequence straddling the transcription start site (TSS) has long been investigated to predict the mRNA levels in various tissues. However, the protein level prediction from sequences has yet to be addressed to the best of our knowledge.

In recent years, deep learning techniques spread in health applications[224, 4, 29, 33] [164, 373] and previous works focused on mRNA level prediction from TSS-straddling sequences [163, 380, 1]. In particular, Convolutional Neural Networks have been adopted to deal with the sequential nature of the DNA[163, 380, 1, 14, 244]. Specifically, Basenjii[163] applies convolutional layers followed by dilated convolutions to share information across large distances in the gene sequences. Dilated convolutions have a wider filter created by inserting spaces in the filter elements. Those gaps exponentially increase the receptive field width, thus taking into account longer dependencies in the sequences. Similarly, Expecto[380] applies convolutional layers to extract features from the sequences using a predefined window's size. Each window yields a set of features stacked together in a high-dimensionality feature vector. Spatial transformations are then applied to reduce feature vector dimensionality to output mRNA levels. On the same line, Xpresso[1] introduced a deep convolutional model composed of two sequential convolutional and max-pooling layers followed by two fully connected layers, demonstrating that a localized region around the transcription start site captures the most relevant information for mRNA level prediction.

Although convolutions represent an effective way to deal with gene sequences, they have some significant limitations that hinder their representational power. Above all, the locality nature of convolutions limits the information propagation in the network among distal elements, requiring many successive layers to expand the receptive field and thus not allowing to capture of long-range relationships and dependencies in sequence elements[306]. In 2017 the attention mechanism revolutionized sequence processing, achieving outstanding performance in capturing long-range dependencies due to each token's global interaction in the input sequence (so-called self-attention), extracting global information directly from the first layer [306]. However, the self-attention operator has a quadratic complexity $O(n^2)$, making the prediction unfeasible for long sequences. The Enformer model[14] firstly applies self-attention to genomic data, capturing wide-ranging relationships and improving mRNA level prediction. However, to keep the computation feasible, the model is composed of a first convolutional step that extracts local features that are then applied to self-attention layers to capture long-range interactions.

Our method, instead, is based on the Perceiver architecture[146], which allows for asymmetric attention between inputs and learnable query vectors, therefore expanding its capabilities to attend longer sequences directly on the raw data without an initial convolutional step. The advantage of the Perceiver architecture is not limited to the computational aspects. The regulatory parts of a gene (e.g., enhancer and silencer) can be at a considerable distance from the gene region on which they act. Unlike CNN and LSTM, these long-range interactions are modeled in the Perceiver architecture, allowing a better mRNA level prediction.

In this chapter, we present three models, all based on the Perceiver architecture: DNAPerceiver, ProteinPerceiver, and DNA&ProteinPerceiver. DNAPerceiver predicts the mRNA and protein levels from the TSS-straddling sequence, and its performances are directly compared with competitor models on various datasets. ProteinPerceiver and DNA&ProteinPerceiver instead predict the protein levels from the protein sequence (ProteinPerceiver) and the combination of TSS-straddling and protein sequences (DNA&ProteinPerceiver), respectively. The latter two models were evaluated under different experimental conditions. However, due to the task's novelty, it is impossible to report comparisons with models in the literature. Below, the Materials and Method paragraph contains the technical details of the models developed and the data used. Subsequently, the Results and Discussion paragraphs report the results obtained. Finally, the Conclusion paragraph summarizes the main contributions of this work.

4.1.2 Proposed method

In order to predict mRNA and protein levels, human protein-coding genes were selected, and their TSS-straddling and protein sequences were obtained (see details in the Dataset paragraph). Then, three models based on a Perceiver architecture were implemented: DNAPerceiver, ProteinPerceiver, and DNA&ProteinPerceiver. DNAPerceiver predicts mRNA levels from the TSS-straddling gene sequence. ProteinPerceiver and DNA&ProteinPerceiver instead predict protein levels from the protein sequence and the combination of the TSS-straddling and protein sequences, respectively.

Although each model differs in the prediction task, the general structure is very similar. First, each model receives a sequence representing the TSS-straddling or the protein sequence as input. Then, the Perceiver model encodes and processes the sequence, and a discrete number is outputted for each sequence. This number represents the samples' average amount of mRNA or protein levels. The greater the number, the greater the amount of molecule (mRNA or protein) circulating. Therefore the main difference between the three Perceiver architectures consists of the input data: TSS-straddling sequences for DNAperceiver; protein sequences for ProteinPerceiver, and TSS-straddling and protein sequences for DNA&ProteinPerceiver architecture.

Datasets

We evaluate our models adopting different settings, depending on the task and thus the desired combination of input sequences and predicted output. Overall, there are two input types: inputDNA and inputProt. InputDNA consists of the sequence of human protein-coding genes upstream and downstream of the transcription start site (TSS). The sequence upstream of the TSS contains the gene's promoter, while the sequence downstream of the TSS contains the exons and introns of the gene. InputDNA sequences are taken from the Xpresso publication [1] due to its particular data curation. Indeed, in this dataset, the TSS positions were accurately revised by Xpresso's authors exploiting Cap Analysis Gene Expression (CAGE) experiments, a method to measure the actual TSS location. Specifically, it comprises 18377 genes split into 16377 genes for training, 1000 for validation, and 1000 for the test. The maximum length of the TSS sequence of a gene is set to 20000 base pairs. Xpresso DNA input also comes with half-life features, which contain general information about the gene (e.g., gene length, number of introns). Therefore, whenever we use InputDNA sequences, we also include half-life features as additional input to our models at different network points, as explained in the architecture section.

InputProt, on the other hand, consists of protein sequences. Therefore, the promoter region and all non-coding parts of a gene are not included in the inputProt sequence. All protein sequences were obtained from Uniprot database [56], processed with Biopython library [55], and intersected with Xpresso's list of protein-coding genes.

As for the labels, we used four typologies for predicting mRNA levels (labelGeneMouse, labelGeneHuman, labelGeneGlio, labelGeneLung) and two typologies for predicting protein levels (labelProtGlio and labelProtLung). labelGeneMouse and labelGeneHuman come from the Xpresso publication, containing the mean mRNA levels of mouse and human samples, respectively. These labels were obtained in the biologically controlled context of cell lines, and therefore the prediction task is limited. To evaluate the predictive capabilities of the models on high throughput multi-omics human data from clinical studies, we selected mRNA and protein levels on patients with glioblastoma [319] and lung cancer [266]. LabelGeneGlio and labelGeneLung contain the labels of the mediated mRNA values for glioblastoma and lung cancer, respectively. The same procedure has been applied to obtain the mediated protein levels for the same patients, named labelProtGlio and labelProtLung, respectively [319, 266].

Given the scarcity of data, except for Xpresso comparisons, we adopt the K-Fold validation setting and average the results across the folds. We set the number of folds K to 10.

Metric

To measure the effectiveness of our methods, we compute the variance explained r^2, also known as the coefficient of determination: $r^2 = 1 - \frac{SSR}{SST}$, where SSR stands for Sum Squared Regression (the sum of the residuals -actual values minus predicted value- squared) and SST for Total Sum of Squares (the sum of the distance the data is away from the mean all squared). This coefficient is the most widely adopted metric for mRNA level prediction, ranging from 0 to 1. When it is 0, the model makes a prediction no better than random, while when it is 1 the model perfectly predicts the actual labels.

DNAPerceiver architecture

As stated above, various models in the literature focused on predicting mRNA levels from the TSS-straddling sequence. This work aims to reveal if mRNA levels can be explained by the TSS-straddling sequence alone. All predictive models do not use the whole gene sequence as input but only the portion straddling TSS, which involves numerous regulatory and transcriptional processes. In particular, the region preceding the TSS contains the promoter, a region targeted explicitly by transcription factors, elements responsible for the final quantity of mRNA produced. The data used in this model are inputDNA as input and labelExprMouse, labelExprHuman, labelGeneGlio, and labelGeneLung as output.

Figure 4.1 shows the architecture of the DNAPerceiver. The model is composed of two distinct flows: one with asymmetric attention as in the original Perceiver model [146], and another with a convolutional step inspired by the Enformer model [14]. The asymmetric attention reduces the complexity of the attention from $O(n^2)$ to $O(n \times m)$ where n is the length of the input sequence, and m is a hyperparameter defying the latent space dimensionality. The model can attend to long sequences and condense their semantic information within a tight latent space. The convolutional step extracts another representation of the same DNA sequence and is then used to query the latent space in the final decoding stage.

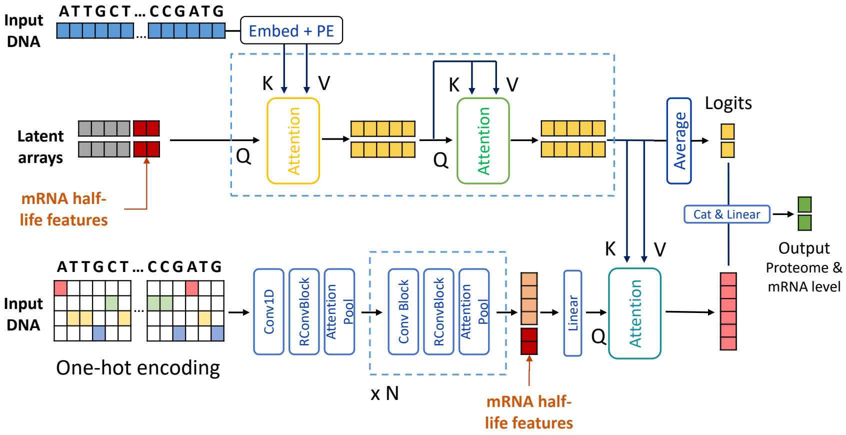


Figure 4.1: DNAPerceiver architecture. It is based on the Perceiver IO model [145]. The upper flow represents the asymmetric attention that distills the sequence in a smaller latent space, where learnable arrays attend to all the input sequences and refine their representations with self-attention and feed-forward networks. The lower flow depicts the decoding stage of the Perceiver IO, where instead of using learnable vectors like in the original model, we use, as the final query, the same sequence processed by a convolutional pipeline inspired by the Enformer model [14]. In this figure, Q,K,V stands for Query, Keys and Values as in typical Transformer architecture, PE is the Positional Encoding, Conv1D is a 1-dimensional Convolution and RConvBlock is a 1D Convolution with a residual connection. The first Convolutional layer is applied to the one-hot encoded version of the sequence, as all previous model of literature, while the upper part of the model embeds the one-hot vectors into learnable embedding vectors through linear projections, as typical Transformer architecture requires.

Therefore, while our model still leverages a convolutional step, it takes more advantage of the recent advancements of attentive architectures, *i.e.* the Perceiver, that have originated from the original Transformer model. Transformers are a class of deep learning models, first introduced by Vaswani *et al.* [306], that attained substantial breakthroughs in natural language processing and computer vision. Specifically, they consist of attention blocks that aggregate information from the entire input sequence by computing a weighted sum across the representations of all other tokens for each sequence token. Since each token directly attends to all other positions in the sequence, they allow for a much better information flow between distal elements, in contrast with convolutional layers, which may require many successive layers to increase the receptive field [306].

These methods were recently applied to model mRNA sequences. However, given the quadratic complexity of attention $O(n^2)$, the length of the input can explode quadratically, rendering it infeasible to encode sequences of more than a few thousand letters. For this reason, our method is based on the Perceiver [146], a model that builds upon Transformers but scales to hundreds of thousands of inputs, as it leverages an asymmetric attention mechanism to distill inputs into a tight latent bottleneck iteratively. Then, the latent arrays go through self-attention blocks to refine their representation and potentially other asymmetric attention layers before getting averaged to obtain the logits for the task at hand.

Specifically, we use the Perceiver IO [145], which improved the decoder capabilities of the model by adding a final decoding stage. This stage acts as a query on the latent arrays, allowing the model to produce outputs of arbitrary size and semantics, and deal with diverse domains without sacrificing the benefits of deep, domain-agnostic processing.

In our implementation, however, we introduce substantial modifications concerning the Perceiver IO architecture. Firstly, instead of learning a different set of output arrays for the decoding stage, we use the same InputDNA sequence after being processed by a Convolutional step. This step consists of multiple Conv layers, Residual connections, and Attention Pooling layers inspired by the Enformer model [14]. Secondly, another difference is that our model in the decoding stage also considers the processed latent arrays by applying a final head that computes their average and uses them as final logits. The processing is similar to that of the original Perceiver model. However, in our case, it is fused with the final decoding mechanism proposed by the Perceiver IO.

Hence, in our architecture, the TSS-straddling or the protein sequence given in input is processed twofold: as learnable vectors for the perceiver flow, where the asymmetric attention is applied with the latent arrays, and as one-hot encoding vectors fed to the convolutional step. After embedding the input letters, we also add a learnable Positional Encoding, initialized with a sinusoidal function as in the original Transformer model to deal with positions in the asymmetric attention.

Latent arrays are initialized with random numbers from a normal distribution with mean 0 and variance 1, while the inputDNA is represented with one-hot encoding vectors, applied to the Convolutional step, and linearly projected into embedding vectors for the attention path. Moreover, mRNA half-life features are injected in both flows: they are appended to the latent arrays and the convolutional step in the final feature representations.

In the DNAPerceiver configuration, we predict the mRNA level for both human cell-line and mouse data, and we evaluate our model on the Xpresso dataset, comparing it with other similar methods and Xpresso itself. Further, we applied the DNAPerceiver model to predict mRNA and protein levels in human high throughput sequencing data. As discussed in the Dataset paragraph, we take the protein labels from two real-human datasets, ending with 10529 pairs of labels for Lung cancer (labelProtLung) and 10280 pairs of labels for glioblastoma (labelProtGlio). In this configuration, we output two predictions for labelGene and labelProt, mRNA and protein levels, respectively, for both datasets.

Implementation details

To represent the A, T, C, and G letters, we use one-hot vectors, and for the perceiver flow, we linearly project them to learnable vectors of dimensionality 32. In addition, we add the letter P as padding. To represent letter positions, we employ learnable positional encodings initialized in a standard sinusoidal fashion[306]. We use 128 latent learnable arrays with a dimensionality of 128 each, constituting the dimensionality of the following self-attention layers. The number of heads in asynchronous attention is set to 1, while self-attention is set to 8. The attention over the input is computed by considering only valid letters and masking the rest. Feed-forward layers have a dimensionality of 256 and GELU nonlinearity. The depth of the Perceiver, the number of layers of asynchronous attention followed by self-attention, is set to 1. For the convolutional query flow, we adopt a similar strategy as Enformer[14] using the first layer of Conv1D with a kernel size of 15, channel dimensionality of 64 and Attention Pooling with a pooling size of 12. Subsequent convolutional layers, forming the Conv tower, have a kernel size of 5 and attention pooling size of 6. Each Conv layer applies GELU nonlinearity and is followed by a residual connection.

The length of the InputDNA sequence is set to 10500, taking the majority part from the promoter side and less from the actual gene, specifically considering 7000 base pairs before the TSS and 3500 after the TSS. We apply dropout throughout the model before each linear projection and attention layer, with a keep probability of 0.8. We train our model using ADAM[368], a batch size of 128, and we follow the learning rate scheduling strategy of [306] with a warmup equal to 8000 iterations. We apply a weight decay of 0.2 and an early stopping strategy to avoid overfitting.

We found it helpful to use the Tanh activation for our final predicted scores only in this configuration and when applied to Xpresso mRNA levels. In the end, we weighted the loss contribution using a weight of 10 for the mRNA.

ProteinPerceiver architecture

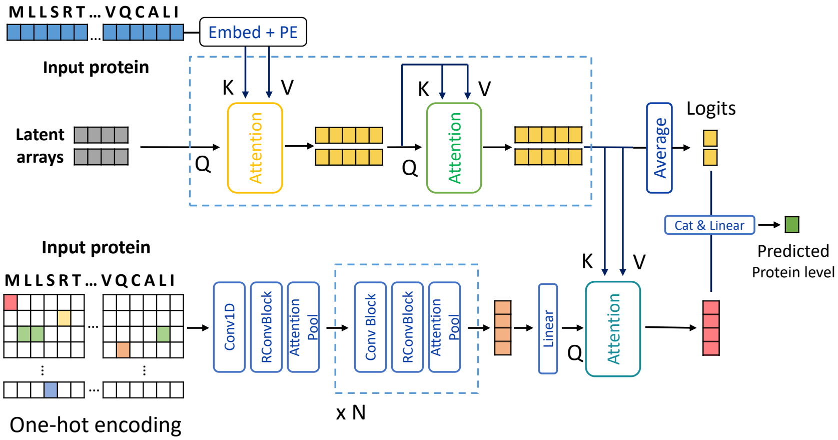


Figure 4.2: ProteinPerceiver architecture, where input is the protein sequence and output is the protein level.

Figure 4.2 shows the ProteinPerceiver model, which aims to measure how much protein levels depend on the protein sequence. The main differences with respect to the DNAPerceiver are the input and output configuration. Here the input is the protein sequence and the output is the protein level. Although the mRNA level prediction task is debated within the scientific community, to the best of our knowledge, there are no publicly available models for protein level prediction using protein sequences. In the last decade, the quantification of mRNA levels has been available in large quantities. Instead, extracting and quantifying proteins is more recent and less mature than mRNA extraction and quantification techniques. Protein quantification is what scientists are most interested in biologically. However, these techniques are currently more expensive, limiting data availability. Moreover, the mRNA level quantification can evaluate more than 20000 protein-coding genes versus approximately 2 to 8 thousand proteins for protein quantification. Unfortunately, given the experiments' novelty, no comparison model in the literature is available. The input consists of inputProtein and labelProtGlio and labelProtLung as output.

We match protein sequences and proteomics labels available, assembling a total of 10430 protein sequences for Glioblastoma and 10699 for Lung Carcinoma with corresponding proteomic labels.

Differently from the DNAPerceiver setting, here we only adjust the dropout keep probability to 0.7 and the attention pooling size in the convolutional query to 10 in the first layer and 5 in the following ones. Moreover, we set the maximum length of the protein sequence to 6000 and the final weight of the MSE loss to 100 for Lung data and 3000 for Glioblastoma data. We optimize the model using Lamb[363], a learning rate of 0.0005, and a Cosine Annealing schedule strategy with 8000 steps of warmup.

Protein&DNAPerceiver architecture

The ProteinPerceiver model receives the protein sequence as input to predict protein levels. However, the protein level is determined by the protein sequence and by regulatory, transcriptional, and epigenetic factors. Although considering all regulatory processes is not straightforward, in this paragraph, we have evaluated the combined effect of the protein and the TSS-straddling sequence to predict the protein levels. The model simultaneously uses inputDNA and inputProtein and outputs labelProtGlio and labelProtLung.

TSS-straddling and protein sequences are matched together when both are available from the Xpresso dataset[1] and the protein sequence dataset, ending up with a total of 9815 triplets gene-protein-labels for Lung Carcinoma and 9534 triplets for Glioblastoma.

Explicitly, our model deals with two different input sequences, one for the Perceiver flow and one for the Convolutional query. In addition, we investigated the use of the input in an alternate manner: when the TSS-straddling sequence is in input to the Perceiver, we use the protein sequence as a query, and vice versa, with protein as the perceiver input, we use the TSS-straddling for the query computation. The Results paragraph shows that the best version differs depending on the data and the prediction. The maximum length of the protein sequence is set to 6000, while the DNA sequence length is set to 8000. If not specified, we kept the same hyperparameters of the DNAPerceiver configuration.

A summary of the architecture names, prediction tasks, input, and outputs is reported in Table 4.1.

Model	Tasks	Input	Output
DNAPerceiver	mRNA levels	InputDNA	labelGene
DNAPerceiver	mRNA&protein levels	InputDNA	labelGene&labelProt
ProteinPerceiver	protein levels	InputProt	labelProt
DNA&ProteinPerceiver	protein levels	InputDNA&InputProt	labelProt

Table 4.1: Summary of the different configurations of our model depending on the prediction task and the input-output setting.

4.1.3 Experimental evaluation

This paragraph discusses the results obtained and the comparison with the state-of-the-art approaches.

Results on mRNA level prediction using Xpresso's labels

In this setting, DNAPerceiver was trained on the Xpresso sequences and their labels, aiming to predict the mRNA level, both from mouse and human organisms (labelExprMouse and labelExprHuman). We follow the split of the original dataset[1], thus obtaining 16377 genes for training and 1000 genes for both validation and test set. As shown in Table 4.2, DNAPerceiver performs better than the Xpresso method in terms of r^2 in human and mouse data. In human cell-line data, it reaches an r^2 of 0.62, which, compared to the 0.59 of the Xpresso model, gains 0.03 points of r^2.

The basenji method has a similar mRNA level prediction task to the one presented in this work. However, a direct comparison cannot be made as Basenji uses Cap Analysis Gene Expression (CAGE) input data which are not available for our dataset (Xpresso's dataset released the sequences but not the CAGE information). However, under his experimental conditions, Basenji reaches a Pearson correlation coefficient ranging from 0.138 to 0.777, depending on the genes considered. These values would translate into a coefficient of determination r^2 between 0.019 and 0.604. In this context, the DNAPerceiver model gets consistent results.

Results on mRNA and protein levels

In this configuration, DNAPerceiver is trained on inputDNA (Xpresso sequences) and predicts the labels from the Lung and Glioblastoma datasets for mRNA and

Model	mRNA r^2
Xpresso [1] human data	0.59
Xpresso [1] mouse data	0.71
DNAPerceiver human data	0.62
DNAPerceiver mouse data	0.72

Table 4.2: Results on the test set of Xpresso dataset in predicting mRNA levels of cell-line data. The input is the InputDNA sequence, and the output is the mRNA level, expressed with the coefficient of determination r^2.

protein levels (labelGeneLung, labelGeneGlio, labelProtLung, and labelProtGlio). Table4.3 reports the results. High-throughput sequencing data from human tissues is much more complex than data obtained from cell lines. Indeed, the cell lines are systematically obtained in the laboratory to have a controlled context and genetic variability as small as possible between the cells. By contrast, the sequencing data from tissues (tumor tissues, too) has a high genetic variability as a multiplicity of regulatory factors between cells and tissues are present. Given the noisy nature of high throughput sequencing data, its mRNA level prediction is not comparable to that of a cell line culture, but it reaches 0.181 of r^2. Furthermore, our focus is to predict the protein level using only the InputDNA sequence. As a result, our model can predict the protein levels achieving 0.161 of r^2, demonstrating its capability to perceive the direct connection between the InputDNA sequence and its corresponding protein level.

Model	mRNA r^2	proteomics r^2
DNAPerceiver Lung	0.181	0.161
DNAPerceiver Glioblastoma	0.150	0.026

Table 4.3: Results on Lung and Glioblastoma data in predicting mRNA and protein level. The input is inputDNA, taken from Xpresso[1] publication, while predicted labels for mRNA and protein levels are labelGeneLung, labelGeneGlio, labelProtLung, and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

Results on protein level using protein sequence as input

Table4.4 reports the result of our model applied to protein sequence as input. In this configuration of our model, named ProteinPerceiver, the input consists of the protein sequences, as explained in the Datasets paragraph in the material and methods paragraph. The aim is to predict the protein level given the protein sequence, a peculiarly complex task, as discussed later in the next paragraph. The obtained outcome varies depending on the data: for Lung data, we found that predicting protein levels from the protein sequence is more complex, achieving a r^2 of 0.085, comparing the 0.161 obtained from the InputDNA. Nonetheless, for Glioblastoma data, our ProteinPerceiver can score a r^2 of 0.028 for protein levels, which is slightly better compared to 0.026 obtained by the DNAPerceiver.

Despite the impact of data quality and prediction task complexity on the results, our model can still capture a part of the relationship between the protein sequence and its corresponding protein level.

Model	proteomics r^2
ProteinPerceiver Lung	0.085
ProteinPerceiver Glioblastoma	0.028

Table 4.4: Results in predicting protein levels from the protein sequence. The input is InputPROT, while predicted labels for protein levels are labelProtLung and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

Results on protein levels using TSS-straddling and protein sequences as input

We wanted to investigate further the model's capabilities with a peculiar configuration, in which we give as input both the TSS-straddling (InputDNA) and the protein sequence. Our model's design facilitates this accomplishment, already treating the input in a double structure: one for the perceiver flow and one for the convolutional query. Therefore, the protein sequence was input to the perceiver and the InputDNA to the convolutional query and vice-versa. We report the results in Table4.5. In this configuration, performances also depend on the specific data: for Lung data, surprisingly, the use of both inputs does not improve the total performances of the model, reaching 0.141 of r^2 compared to the 0.161 obtained using only InputDNA sequence. On the contrary, using both inputs slightly improves the results on Glioblastoma data, achieving 0.031 of r^2.

Model	proteomics r^2
Protein&DNA Perceiver Lung	0.141
Protein&DNA Perceiver Glioblastoma	0.031

Table 4.5: Results in predicting protein levels from both the DNA sequence and the protein sequence used as inputs. The input is InputPROT and inputDNA, and the predicted labels for protein expression are labelProtLung and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

4.1.4 Discussion

In regards to predicting mRNA levels from the sequence upstream and downstream of the TSS (thus including part of the promoter and part of the gene), DNAPerceiver shows results superior to Xpresso in the case of the human cell lines and murine samples. Unlike the Xpresso model, the DNAPerceiver model exploits the self-attention mechanism to predict the mRNA levels. Having the same input sequence size and output levels as Xpresso, the DNAPerceiver model achieves superior results since long-range interactions between the most distant regions of the promoter and the gene sequence are fully exploited in the model and not limited by the size of the convolutional kernel. Moreover, as can be expected, the prediction of mRNA levels in cell lines achieves better results than mRNA level prediction in tumor samples. This aspect could be explained by the different boundary conditions of the two situations. In the first case, the mRNA expression is controlled to ensure the reproducibility and stability of the cell lines. In the second case, the intrinsic samples' variability cannot be limited and pathological conditions profoundly alter the biological context. Since no comparable studies in predicting protein expression levels are available, more distant works that predict protein expression are described. In particular, Barzine et al. [22] purpose is the imputation of unquantified proteins exploiting mRNA expression data. Indeed, it does not answer whether it is possible to predict protein expression starting from the gene sequence. In detail, mRNA expression values are known in the literature to be predictive of protein expression values, as there is often a positive correlation between the mRNA and its protein expression. Barzine et al. also consider the variability of the same protein in different samples (e.g., people) based on mRNA expression variability. As innovative as it is, Barzine et al. answer a very different question, namely quantifying the expression values of those proteins whose mRNA value is known. Fernandes et al. [92] aim to predict the expression of proteins using the encoding of their codons. However, the prediction was made for Escherichia coli from two datasets with a limited number of proteins. The first one contains the expression levels of two proteins, a DNA polymerase, and a single-chain antibody, for 55 codon encodings. The second one contains the level of a green fluorescent protein produced with 154 different codon encodings. The main limitation of Fernandes et al.'s work is the number of proteins quantified based on the specific sequence detected in the sample. Moreover, it is based on the correlation between the levels of the green fluorescent protein and the free energy of the protein itself. Although the purpose is similar to the Perceiver, there is no way to make a direct comparison with our work.

Besides the improvement in mRNA level prediction, the main novelty of this work is the first adoption of the Perceiver architecture for gene expression, and the prediction of protein levels from the TSS and protein sequences. This aspect is doubly challenging: 1. Protein extraction and quantification techniques have emerged recently, so data availability still needs to be improved compared to mRNA datasets; 2. The protein sequence has target regions for post-translation regulators; however, the promoter region is not used as input in the ProteinPerceiver model. It is noted that the prediction of protein levels is considerably lower than mRNA ones, whether the prediction exploits the TSS or the protein sequence. The complexity of the problem can explain this phenomenon. The protein level is influenced by notable post-transcriptional and post-translational regulatory phenomena (e.g., ubiquitination), which are not fed to the models. Moreover, the TSS sequences (composed of the promoter and a part of the gene) have a greater predictive power of the protein level than the protein sequences. This behavior could depend on the presence of the promoter. Indeed, the promoter is the region that favors the expression regulation (both of genes and, therefore, of proteins), and it is responsible for interacting with transcription factors. When the model is trained simultaneously with the TSS and the protein sequence, the predictive power of protein level increases; however, it remains lower than the prediction of protein levels using only the TSS sequence. In this sense, the TSS sequence seems more informative than the protein one. Although the protein expression level prediction is critical, the expression value of a protein compared to the others is crucial too. In this sense, the predictive power of the proposed model can be of interest to scientists. Indeed, 60% and 68% (globally and in medulloblastoma, respectively) of the most highly expressed proteins are predicted as expressed. In the end, the main reason for the noisy result could be attributed to post-transcriptional regulatory processes which are widely known as crucial players in protein expression.

Chapter 5

Conclusions

The goal of the research activities I carried out during my Ph.D. period was to develop new deep learning architectures based on Transformers and attention mechanism, improving artificial intelligence technologies. Specifically, I focused my efforts towards the replication of some fundamental human abilities that connects vision and language, such as describing in natural language a given visual stimuli, called image captioning, or matching images and texts, called cross-modal retrieval. Connecting vision and language motivated most of the works presented in the previous chapters, and it has been tackled by addressing the mentioned two main research tasks, along with some variants such as novel object captioning and visual question answering. Moreover, in the end, I expanded the scope of attention mechanism to the languages of life: the DNA and the protein sequences, with the goal to predict and discover knowledge from these fascinating life codes.

In the first part, we addressed the problem of image captioning by reporting a thorough quantitative and qualitative analysis of the literature on the task, and by proposing new deep neural networks based on Transformer architectures that have achieved new state-of-the-art results. We also introduce a solution for one of its variants, novel object captioning, that shows superior performances.

In the second part, we instead directed our attention to cross-modal retrieval, another important task that effectively combines vision and language by matching images and texts. In this regard, we have first tackled the problem by exploiting fully-attentive paradigm, devising an aggregation function that condenses information from a set of elements. Then, we have addressed the task by introducing an efficient Transformer architecture making use of distillation in order to fill in the gap between effectiveness and efficiency. Furthermore, we have also investigated visual-semantic models for artistic and cultural heritage domains, which represents more complex and challenging scenarios.

Finally, in the last part of the thesis we addressed the gene and protein expression, other compelling tasks associated with different languages: the DNA and the protein sequences. On this matter, we devised a method based on Perceiver architecture that achieved remarkable results.

In the following, I summarize the contributions made by this thesis and draw the final conclusions on the results achieved so far.

Image captioning and novel object captioning

Connecting vision and language by creating deep learning systems capable of automatically describing images in natural language is one of the major challenges in artificial intelligence. In this context, we addressed the image captioning task from different perspectives.

Firstly, we claimed that fully-attentive architectures represent the best performing tool available today to address vision and language tasks. To this end, we presented one of the first Transformer-based architectures for image captioning which encapsulates a multi-layer encoder for image regions and a multi-layer decoder which generates the output sentence. To exploit both low-level and high-level contributions, encoding and decoding layers are connected in a mesh-like structure, weighted through a learnable gating mechanism. Noticeably, this connectivity pattern is unprecedented for other fully-attentive architectures. Moreover, our visual encoder integrates relationships in a multi-level fashion between image regions by exploiting learned a priori knowledge, modeled via persistent memory vectors. We demonstrated that our solution surpasses all previous methods for image captioning, reaching a new state of the art on the online COCO evaluation server and ranking first among all other published proposals. As a complementary contribution, we validated the performance of our model also when describing novel objects not present in the training set.

Secondly, inspired by knowledge distillation technique, we investigated the use of the mean teacher learning paradigm applied to the image captioning task. We presented CaMEL, Captioner with Mean tEacher Learning, a novel Transformer-based network that is trained with the interaction of two different language models that learn from each other through knowledge distillation and model averaging. Experimentally, we validated our method with different knowledge distillation strategies and visual feature extractors, surpassing the current state of the art on the COCO dataset without using external data and pre-training strategies. Furthermore, CaMEL achieves similar performance to other recent proposals that make use of large-scale pre-training, while being much smaller in terms of number of parameters.

Finally, driven by the desire to apply captioning systems to more real-world scenarios, we have presented a fully-attentive approach for novel object captioning that learns to select and describe unseen visual concepts. Our method is based on a class-independent region selector and an image captioning model trained with a differentiable grid beam search algorithm that generates sentences with given constraints, in an end-to-end fashion. Experimental results have shown that our model achieves a new state of the art on the *held-out* COCO dataset, demonstrating its effectiveness in successfully describing novel objects.

Cross-modal retrieval

While image captioning methods combine vision and language in a generative manner, cross-modal retrieval architectures build common representations to connect the two domains and retrieve textual elements given visual queries, and vice versa. In this context, we addressed the problem either by using supervised solutions or semi-supervised approaches exploiting the knowledge learned on large-scale datasets to retrieve items on a different domain, such as that of art and cultural heritage.

Regarding the supervised setting and motivated by the fact that aggregating features has always played a critical role in deep learning architectures and retrieval, we proposed a novel aggregation function based on a variant of the cross-attention mechanism, that reduces sets or sequences of elements into a single compact representation in a learnable fashion. We specifically tailored our method for cross-modal retrieval and experimentally demonstrated that our approach achieves better performances when compared to other commonly used reduction functions. Furthermore, for the typical supervised setting we presented an efficient and effective architecture for cross-modal retrieval. Specifically, we proposed to learn an alignment score by independently forwarding the visual and the textual pipelines using a state-of-the-art Vision-Language Transformer as a backbone. Then, we used the scores produced by the alignment head to learn a visual-textual common space, which can produce easily indexable fixed-length features. Specifically, we approached the problem using a learn-to-rank distillation objective, which empirically demonstrated its effectiveness over the standard hinge-based triplet ranking loss to optimize the common space. The experiments conducted on COCO confirmed the validity of our approach. The results demonstrated that this method helps fill the gap between effectiveness and efficiency, enabling this system to be deployed in large-scale cross-modal retrieval scenarios.

On a different setting, we tackled the task of building visual-semantic retrieval approaches for the cultural heritage and digital humanities domain. To this aim, we proposed a semi-supervised approach which does not rely on labelled data on the artistic domain and translates the knowledge learned on ordinary visual-semantic datasets to the more challenging case of artistic data. After introducing Artpedia, a novel dataset for the task composed of illustrations and textual descriptions coming from historical manuscripts, we validated the proposed strategy through extensive experiments and analyses. Moreover, our method also discriminates between visual and contextual sentences of the same image.

Gene and protein expression

Since Transformer architectures are well suited to deal with sequential data, we also investigated their use applied to different life languages, DNA and proteins, aiming to predict the gene and protein expressions. Various papers have addressed mRNA level prediction in the literature, which mainly includes convolutional or long short-term memory networks. We instead presented three Perceiver-type architectures for mRNA levels prediction on cell lines and high-throughput human samples. Furthermore, we introduced a novel task, which consists on the prediction of protein levels from the TSS-straddling and protein sequences. The results have shown the advantages of our architecture in predicting mRNA levels compared to competitors. On the other hand, protein level prediction benefits more from the TSS-straddling sequence than the protein one. This aspect could be explained by the presence of the promoter region in the TSS-straddling sequence. Although various experimental conditions have been considered, other biological post-transcriptional and post-translation regulations can be included in the models to enhance predictions.

Future works and open problems

The Transformer revolution has undoubtedly brought astonishing advancements and breakthroughs in artificial intelligence over the last few years. For sure this technology will contribute to new improvements in the following years and in many machine intelligence applications. All disciplines employing some sort of language to express rules, relations or functions will be affected by this revolution, from NLP to genomics, finance and also vision.

Large language models (LLMs), made by Transformer building blocks, are unlocking new possibilities in areas such as search engines, natural language processing, healthcare, robotics and code generation. As an example, the recent ChatGPT released by OpenAI in December 2022 impressed the entire world due to its impressive capability in generating compelling meaningful texts given natural prompts and commands. Another worth mentioning breakthrough is AlphaFold, made by DeepMind in 2020, which predicts with extremely high accuracy the 3D protein structure of hundreds of millions of proteins, given only its amino acid sequence, a century-long problem in biology. We will surely witness other incredible uses of this technology in the future.

Regarding Image captioning and cross-modal retrieval, they are intrinsically complex challenges for machine intelligence as they integrate difficulties from both Computer Vision and NLP. However, many open challenges remain since accuracy, robustness, and generalization results are far from satisfactory. Similarly, requirements of fidelity, naturalness, and diversity are not yet met. Specifically, we can trace three main directions for the image captioning field. Firstly, procedural and architectural challenges: since image captioning models are data greedy, pre-training on large-scale datasets, even if not well-curated, is becoming a solid strategy. In this regard, promoting the public release of such datasets will be fundamental to fostering reproducibility and allowing fair comparisons. The growing size of pre-training models is also a concern, and the community will need to investigate less computationally-intensive alternatives to promote equality in the community.

Secondly, generalization, diversity and long-tail concepts: while pre-training on web-scale datasets provides a promising direction to increase generalization and promoting long-tail concepts, specializing in particular domains and generating captions with different styles and aims is still among the main open challenges. Although we discussed some attempts to encourage naturalness and diversity, further research is needed to design models that are suitable for real-world applications. In this sense, models which can deal with long-tail concepts or image captioning variants such as novel object captioning offers a valuable promise of modeling real-life scenarios and generalizing to different contexts.

Thirdly, design of trustworthy AI solutions: due to its potential in human-machine interaction, image captioning needs solutions that are transparent and acceptable for end-users, framed as overcome bias, and interpretable. Since most vision-and-language datasets share common patterns and regularities, datasets bias and overrepresented visual concepts are major issues for any vision-and-language task. In this sense, some effort should be devoted to the study of fairness and bias: two possible directions entail designing specific evaluation metrics and focusing on the robustness to unwanted correlations. Further, despite the promising performance on the benchmark datasets, state-of-the-art methods are not yet satisfactory when applied in the wild. A possible reason for this is the evaluation procedures used and their impact on the training approaches currently adopted. In this sense, the design of appropriate and reproducible evaluation protocols and insightful metrics remains an open challenge. Finally, since existing image captioning algorithms lack reliable and interpretable means for determining the cause of a particular output, further research is needed to shed more light on model explainability, focusing on how these deal with different modalities or novel concepts.

Publications and achievements

The efforts presented in this thesis have resulted in publications in international conferences and journals. Among all the others, the work on image captioning adopting fully-attentive paradigm, called Meshed-Memory Transformer reported in Sec. 2.2, has been accepted at the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020, and is among the most cited paper of the task in recent years. Also, the comprehensive survey on image captioning presented in Sec. 2.1, has resulted in a journal paper on IEEE Transactions on Pattern Analysis and Machine Intelligence. The works on cross-modal retrieval and cultural heritage, as well, have resulted in several publications which have been widely appreciated by the community. Some of the other presented results, instead, are currently under revision in major conferences or journals.

As a complementary result of this thesis, I have contributed together with other colleagues, to the development of a PyTorch library called Speaksee, specifically designed for visual-semantic tasks. It contains utility functions and re-implementations of state-of-the-art models for different tasks that combine vision and language, such as image captioning, cross-modal retrieval, and visual-question answering. It is worth noting that most of our first works that integrate visual-semantic techniques were based on this library.

As a final note, I would like to thank all my colleagues from the AImagelab and the NVIDIA AI Technology Center, together with my tutor and supervisors for the advice and opportunities, without which this work would not have been possible.

Appendix A

List of publications

The following list of publications includes all conference papers, journal articles, and book chapters published during my Ph.D. period, as well as recent pre-prints. Content and experimental results published in some of these papers have been included in the previous chapters, with explicit permission given by the other authors.

[1] Matteo Stefanini, Riccardo Lancellotti, Lorenzo Baraldi, and Simone Calderara. A Deep Learning based approach to VM behavior identification in cloud systems. In *Proceedings of the International Conference on Cloud Computing and Services Science*, 2019.

[2] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Artpedia: A New Visual-Semantic Dataset with Visual and Contextual Sentences in the Artistic Domain. In *Proceedings of the International Conference on Image Analysis and Processing*, 2019.

[3] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Explaining digital humanities by aligning images and textual descriptions. *Pattern Recognition Letters*, 129:166–172, 2020.

[4] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-Memory Transformer for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[5] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. A novel attention-based aggregation function to combine vision and language. In *Proceedings of the International Conference on Pattern Recognition*, 2021.

[6] Marco Cagrandi, Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Learning to Select: A Fully Attentive Approach for Novel Object Captioning. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, 2021.

[7] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From Show to Tell: A Survey on Deep Learning-based Image Captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[8] Manuele Barraco, Matteo Stefanini, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. CaMEL: Mean Teacher Learning for Image Captioning. In *Proceedings of the International Conference on Pattern Recognition*, 2022.

[9] Nicola Messina, Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, Giuseppe Amato, and Rita Cucchiara. ALADIN: Distilling Fine-grained Alignment Scores for Efficient Image-Text Matching and Retrieval. In *Proceedings of the 19th International Conference on Content-based Multimedia Indexing*, pages 64–70, 2022.

[10] Matteo Stefanini, Marta Lovino, Rita Cucchiara, and Elisa Ficarra. Predicting gene and protein expression levels from DNA and protein sequences with Perceiver. *bioRxiv*, 2022.

Bibliography

[1] Vikram Agarwal and Jay Shendure. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. *Cell reports*, 31(7):107663, 2020.

[2] Harsh Agrawal, Karan Desai, Xinlei Chen, Rishabh Jain, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *ICCV*, 2019.

[3] Ahmet Aker and Robert Gaizauskas. Generating image descriptions using dependency relational patterns. In *ACL*, 2010.

[4] Stefano Allegretti, Federico Bolelli, Federico Pollastri, Sabrina Longhitano, Giovanni Pellacani, and Costantino Grana. Supporting Skin Lesion Diagnosis with Content-Based Image Retrieval. In *ICPR*, 2021.

[5] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic Propositional Image Caption Evaluation. In *ECCV*, 2016.

[6] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Guided Open Vocabulary Image Captioning with Constrained Beam Search. In *EMNLP*, 2017.

[7] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[8] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

[9] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016.

[10] Jyoti Aneja, Harsh Agrawal, Dhruv Batra, and Alexander Schwing. Sequential latent spaces for modeling the intention during diverse image captioning. In *ICCV*, 2019.

[11] Jyoti Aneja, Aditya Deshpande, and Alexander G Schwing. Convolutional image captioning. In *CVPR*, 2018.

[12] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. In *ICLR*, 2018.

[13] Alfredo Ardila, Byron Bernal, and Monica Rosselli. Language and visual perception associations: meta-analytic connectivity modeling of Brodmann Area 37. *Behavioural Neurology*, 2015.

[14] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2014.

[16] Shuang Bai and Shan An. A survey on automatic image caption generation. *Neurocomputing*, 311:291–304, 2018.

[17] Zechen Bai, Yuta Nakashima, and Noa Garcia. Explain Me the Painting: Multi-Topic Knowledgeable Art Description Generation. In *ICCV*, 2021.

[18] Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. Weakly Supervised Relative Spatial Reasoning for Visual Question Answering. In *ICCV*, 2021.

[19] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshops*, 2005.

[20] Lorenzo Baraldi, Marcella Cornia, Costantino Grana, and Rita Cucchiara. Aligning text and document illustrations: towards visually explainable digital humanities. In *ICPR*, 2018.

[21] Manuele Barraco, Matteo Stefanini, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. CaMEL: Mean Teacher Learning for Image Captioning. In *ICPR*, 2022.

[22] Mitra Parissa Barzine, Karlis Freivalds, James C Wright, Mārtiņš Opmanis, Darta Rituma, Fatemeh Zamanzad Ghavidel, Andrew F Jarnuczak, Edgars Celms, Kārlis Čerāns, Inge Jonassen, et al. Using deep learning to extrapolate protein expression measurements. *Proteomics*, 2020.

[23] Huixia Ben, Yingwei Pan, Yehao Li, Ting Yao, Richang Hong, Meng Wang, and Tao Mei. Unpaired Image Captioning with Semantic-Constrained Self-Learning. *IEEE Trans. Multimedia*, 2021.

[24] Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *JAIR*, 55:409–442, 2016.

[25] Roberto Bigazzi, Federico Landi, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Explore and Explain: Self-supervised Navigation and Recounting. In *ICPR*, 2020.

[26] Adrian Bird. Perceptions of epigenetics. *Nature*, 447(7143):396, 2007.

[27] Ali Furkan Biten, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. Good news, everyone! context driven entity-aware captioning for news images. In *CVPR*, 2019.

[28] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *TACL*, pages 135–146, 2017.

[29] Federico Bolelli, Stefano Allegretti, and Costantino Grana. One DAG to Rule Them All. *IEEE Trans. PAMI*, 2021.

[30] Sebastian Bruch. An alternative cross entropy loss for learning-to-rank. In *Web Conference*, 2021.

[31] Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. MUREL: Multimodal Relational Reasoning for Visual Question Answering. In *CVPR*, 2019.

[32] Ozan Caglayan, Pranava Madhyastha, and Lucia Specia. Curious Case of Language Generation Evaluation Metrics: A Cautionary Tale. In *COLING*, 2020.

[33] Michele Cancilla, Laura Canalini, Federico Bolelli, Stefano Allegretti, Salvador Carrión, Roberto Paredes, Jon Ander Gómez, Simone Leo, Marco Enrico Piras, Luca Pireddu, Asaf Badouh, Santiago Marco-Sola, Lluc Alvarez, Miquel Moreto, and Costantino Grana. The DeepHealth Toolkit: A Unified Framework to Boost Biomedical Applications. In *ICPR*, 2021.

[34] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 2007.

[35] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021.

[36] Angelo Carraggi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Visual-Semantic Alignment Across Domains Using a Semi-Supervised Approach. In *ECCV Workshops*, 2018.

[37] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *CVPR*, 2021.

[38] Moitreya Chatterjee and Alexander G Schwing. Diverse and coherent paragraph generation from images. In *ECCV*, 2018.

[39] Fuhai Chen, Rongrong Ji, Jiayi Ji, Xiaoshuai Sun, Baochang Zhang, Xuri Ge, Yongjian Wu, Feiyue Huang, and Yan Wang. Variational structured semantic inference for diverse image captioning. In *NeurIPS*, 2019.

[40] Fuhai Chen, Rongrong Ji, Jinsong Su, Yongjian Wu, and Yunsheng Wu. StructCap: Structured Semantic Embedding for Image Captioning. In *ACM Multimedia*, 2017.

[41] Fuhai Chen, Rongrong Ji, Xiaoshuai Sun, Yongjian Wu, and Jinsong Su. GroupCap: Group-based Image Captioning with Structured Relevance and Diversity Constraints. In *CVPR*, 2018.

[42] Long Chen, Zhihong Jiang, Jun Xiao, and Wei Liu. Human-like Controllable Image Captioning with Verb-specific Semantic Roles. In *CVPR*, 2021.

[43] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning. In *CVPR*, 2017.

[44] Shi Chen and Qi Zhao. Boosted attention: Leveraging human attention for image captioning. In *ECCV*, 2018.

[45] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *CVPR*, 2020.

[46] Tseng-Hung Chen, Yuan-Hong Liao, Ching-Yao Chuang, Wan-Ting Hsu, Jianlong Fu, and Min Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *ICCV*, 2017.

[47] Xinlei Chen and C Lawrence Zitnick. Mind's Eye: A Recurrent Visual Representation for Image Caption Generation. In *CVPR*, 2015.

[48] Xinpeng Chen, Lin Ma, Wenhao Jiang, Jian Yao, and Wei Liu. Regularizing RNNs for Caption Generation by Reconstructing The Past with The Present. In *CVPR*, 2018.

[49] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, 2020.

[50] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive Faster R-CNN for object detection in the wild. In *CVPR*, 2018.

[51] Yuhua Chen, Wen Li, and Luc Van Gool. ROAD: Reality Oriented Adaptation for Semantic Segmentation of Urban Scenes. In *CVPR*, 2018.

[52] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[53] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *NeurIPS*, 2015.

[54] Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Attend to you: Personalized image captioning with context sequence memory networks. In *CVPR*, 2017.

[55] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[56] UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.

[57] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, Control and Tell: A Framework for Generating Controllable and Grounded Captions. In *CVPR*, 2019.

[58] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. SMArT: Training Shallow Memory-aware Transformers for Robotic Explainability. In *ICRA*, 2020.

[59] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Explaining transformer-based image captioning models: An empirical analysis. *AI Communications*, pages 1–19, 2021.

[60] Marcella Cornia, Lorenzo Baraldi, Giuseppe Fiameni, and Rita Cucchiara. Universal Captioner: Long-Tail Vision-and-Language Model Training through Content-Style Separation. *arXiv preprint arXiv:2111.12727*, 2021.

[61] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Visual Saliency for Image Captioning in New Multimedia Services. In *ICMEW*, 2017.

[62] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Paying More Attention to Saliency: Image Captioning with Saliency and Context Attention. *ACM TOMM*, 14(2):1–21, 2018.

[63] Marcella Cornia, Lorenzo Baraldi, Hamed R Tavakoli, and Rita Cucchiara. A unified cycle-consistent neural model for text and image retrieval. *Multimedia Tools and Applications*, 79(35):25697–25721, 2020.

[64] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Explaining digital humanities by aligning images and textual descriptions. *Pattern Recognition Letters*, 129:166–172, 2020.

[65] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-Memory Transformer for Image Captioning. In *CVPR*, 2020.

[66] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[67] Francis Crick, Leslie Barnett, Sydney Brenner, Richard J Watts-Tobin, et al. General nature of the genetic code for proteins. *Nature*, 1961.

[68] Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge Belongie. Learning to evaluate image captioning. In *CVPR*, 2018.

[69] Bo Dai, Sanja Fidler, and Dahua Lin. A neural compositional paradigm for image captioning. In *NeurIPS*, 2018.

[70] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards Diverse and Natural Image Descriptions via a Conditional GAN. In *ICCV*, 2017.

[71] Bo Dai and Dahua Lin. Contrastive learning for image captioning. In *NeurIPS*, 2017.

[72] Bo Dai, Deming Ye, and Dahua Lin. Rethinking the form of latent states in image captioning. In *ECCV*, 2018.

[73] Riccardo Del Chiaro, Andrew D Bagdanov, and Alberto Del Bimbo. Webly-supervised Zero-shot Learning for Artwork Instance Recognition. *Pattern Recognition Letters*, 2019.

[74] Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost van de Weijer. RATT: Recurrent Attention to Transient Tasks for Continual Image Captioning. In *NeurIPS*, 2020.

[75] Chaorui Deng, Ning Ding, Mingkui Tan, and Qi Wu. Length-controllable image captioning. In *ECCV*, 2020.

[76] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations From Textual Annotations. In *CVPR*, 2021.

[77] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G Schwing, and David Forsyth. Fast, diverse and accurate image captioning guided by part-of-speech. In *CVPR*, 2019.

[78] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2018.

[79] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[80] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*, 2021.

[81] Desmond Elliott, Stella Frank, and Eva Hasler. Multilingual image description with neural sequence models. *ICLR*, 2015.

[82] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30K: Multilingual English-German Image Descriptions. In *ACL Workshops*, 2016.

[83] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *CVPR*, 2018.

[84] Manel Esteller. Epigenetics in cancer. *New England Journal of Medicine*, 358(11):1148–1159, 2008.

[85] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In *BMVC*, 2017.

[86] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015.

[87] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[88] Zheng-cong Fei. Fast Image Caption Generation with Position Alignment. *AAAI Workshops*, 2019.

[89] Zhengcong Fei. Iterative Back Modification for Faster Image Captioning. In *ACM Multimedia*, 2020.

[90] Yang Feng, Lin Ma, Wei Liu, and Jiebo Luo. Unsupervised image captioning. In *CVPR*, 2019.

[91] Yansong Feng and Mirella Lapata. Automatic Caption Generation for News Images. *IEEE Trans. PAMI*, 35(4):797–812, 2012.

[92] Armando Fernandes and Susana Vinga. Improving protein expression prediction using extra features and ensemble averaging. *PloS One*, 2016.

[93] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NeurIPS*, 2018.

[94] Andrea Frome, Greg S Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. DeViSE: a deep visual-semantic embedding model. In *NeurIPS*, 2013.

[95] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. StyleNet: Generating Attractive Visual Captions with Styles. In *CVPR*, 2017.

[96] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic Compositional Networks for Visual Captioning. In *CVPR*, 2017.

[97] Junlong Gao, Shiqi Wang, Shanshe Wang, Siwei Ma, and Wen Gao. Self-critical n-step Training for Image Captioning. In *CVPR*, 2019.

[98] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *CVPR*, 2019.

[99] Peng Gao, Haoxuan You, Zhanpeng Zhang, Xiaogang Wang, and Hongsheng Li. Multi-modality latent interaction network for visual question answering. In *ICCV*, 2019.

[100] Noa Garcia and George Vogiatzis. How to Read Paintings: Semantic Art Understanding with Multi-Modal Retrieval. In *ECCV Workshops*, 2018.

[101] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[102] Hongwei Ge, Zehang Yan, Kai Zhang, Mingde Zhao, and Liang Sun. Exploring Overall Contextual Information for Image Captioning in Human-Like Cognitive Style. In *ICCV*, 2019.

[103] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[104] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *CVPR*, 2017.

[105] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*, 2018.

[106] Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-Captioning: Coarse-to-Fine Learning for Image Captioning. In *AAAI*, 2018.

[107] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, and Gang Wang. Unpaired image captioning by language pivoting. In *ECCV*, 2018.

[108] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *ICCV*, 2019.

[109] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. An Empirical Study of Language CNN for Image Captioning. In *ICCV*, 2017.

[110] Dan Guo, Yang Wang, Peipei Song, and Meng Wang. Recurrent relational memory network for unsupervised image captioning. *IJCAI*, 2020.

[111] Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Hanqing Lu. Aligning linguistic words and visual semantic units for image captioning. In *ACM Multimedia*, 2019.

[112] Longteng Guo, Jing Liu, Peng Yao, Jiangwei Li, and Hanqing Lu. Mscap: Multi-style image captioning with unpaired stylized text. In *CVPR*, 2019.

[113] Longteng Guo, Jing Liu, Xinxin Zhu, Xingjian He, Jie Jiang, and Hanqing Lu. Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. *IJCAI*, 2020.

[114] Longteng Guo, Jing Liu, Xinxin Zhu, and Hanqing Lu. Fast Sequence Generation with Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2101.09698*, 2021.

[115] Longteng Guo, Jing Liu, Xinxin Zhu, Peng Yao, Shichen Lu, and Hanqing Lu. Normalized and Geometry-Aware Self-Attention Network for Image Captioning. In *CVPR*, 2020.

[116] Ankush Gupta, Yashaswi Verma, and C Jawahar. Choosing linguistics over vision to describe images. In *AAAI*, 2012.

[117] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning Images Taken by People Who Are Blind. In *ECCV*, 2020.

[118] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *CVPR*, 2015.

[119] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[120] Sen He, Wentong Liao, Hamed R Tavakoli, Michael Yang, Bodo Rosenhahn, and Nicolas Pugeault. Image captioning through image transformer. In *ACCV*, 2020.

[121] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *ECCV*, 2016.

[122] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *ECCV*, 2018.

[123] Lisa Anne Hendricks, Subhashini Venugopalan, Marcus Rohrbach, Raymond Mooney, Kate Saenko, and Trevor Darrell. Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data. In *CVPR*, 2016.

[124] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image Captioning: Transforming Objects into Words. In *NeurIPS*, 2019.

[125] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[126] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Workshops*, 2015.

[127] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[128] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 47:853–899, 2013.

[129] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *ICML*, 2018.

[130] Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In *ACL*, 2017.

[131] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A Comprehensive Survey of Deep Learning for Image Captioning. *ACM Computing Surveys*, 51(6):1–36, 2019.

[132] Mehrdad Hosseinzadeh and Yang Wang. Image Change Captioning by Learning from an Auxiliary Task. In *CVPR*, 2021.

[133] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017.

[134] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling Up Vision-Language Pre-training for Image Captioning. *CVPR*, 2021.

[135] Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. VIVO: Visual Vocabulary Pre-Training for Novel Object Captioning. *AAAI*, 2020.

[136] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[137] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on Attention for Image Captioning. In *ICCV*, 2019.

[138] Lun Huang, Wenmin Wang, Yaxian Xia, and Jie Chen. Adaptively Aligned Image Captioning via Adaptive Attention Time. In *NeurIPS*, 2019.

[139] Qingbao Huang, Yu Liang, Jielong Wei, Cai Yi, Hanyu Liang, Ho-fung Leung, and Qing Li. Image Difference Captioning with Instance-Level Fine-Grained Feature Representation. *IEEE Trans. Multimedia*, 2021.

[140] Yiqing Huang and Jiansheng Chen. Teacher-Critical Training Strategies for Image Captioning. *arXiv preprint arXiv:2009.14405*, 2020.

[141] Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *CVPR*, 2016.

[142] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *ICLR*, 2020.

[143] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-Domain Weakly-Supervised Object Detection through Progressive Domain Adaptation. In *CVPR*, 2018.

[144] Eva Jablonka and Marion J Lamb. The changing concept of epigenetics. *Annals of the New York Academy of Sciences*, 981(1):82–96, 2002.

[145] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[146] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.

[147] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. In *EMNLP*, 2018.

[148] Jiayi Ji, Yunpeng Luo, Xiaoshuai Sun, F. Chen, Gen Luo, Yongjian Wu, Yue Gao, and Rongrong Ji. Improving Image Captioning by Leveraging Intra- and Inter-layer Global Representation in Transformer Network. In *AAAI*, 2021.

[149] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[150] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding the Long-Short Term Memory model for Image Caption Generation. In *ICCV*, 2015.

[151] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *CVPR*, 2020.

[152] Ming Jiang, Junjie Hu, Qiuyuan Huang, Lei Zhang, Jana Diesner, and Jianfeng Gao. REO-Relevance, Extraness, Omission: A Fine-grained Evaluation for Image Captioning. In *EMNLP-IJCNLP*, 2019.

[153] Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao. TIGEr: text-to-image grounding for image caption evaluation. In *ACL*, 2019.

[154] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent Fusion Network for Image Captioning. In *ECCV*, 2018.

[155] Baoyu Jing, Pengtao Xie, and Eric Xing. On the Automatic Generation of Medical Imaging Reports. In *ACL*, 2018.

[156] Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. Stroke controllable fast style transfer with adaptive receptive fields. In *ECCV*, 2018.

[157] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully convolutional Localization Networks for Dense Captioning. In *CVPR*, 2016.

[158] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. In *BMVC*, 2014.

[159] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[160] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *NeurIPS*, 2014.

[161] Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A Smith. Transparent Human Evaluation for Image Captioning. *arXiv preprint arXiv:2111.08940*, 2021.

[162] Lei Ke, Wenjie Pei, Ruiyu Li, Xiaoyong Shen, and Yu-Wing Tai. Reflective Decoding Network for Image Captioning. In *ICCV*, 2019.

[163] David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5):739–750, 2018.

[164] Abbas Khan, Zainab Rehman, Huma Farooque Hashmi, Abdul Aziz Khan, Muhammad Junaid, Abrar Mohammad Sayaf, Syed Shujait Ali, Fakhr Ul Hassan, Wang Heng, and Dong-Qing Wei. An integrated systems biology and network-based approaches to identify novel biomarkers in breast cancer cell lines using gene expression data. *Interdisciplinary Sciences: Computational Life Sciences*, 12(2):155–168, 2020.

[165] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. In *ACL*, 2017.

[166] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Dense Relational Captioning: Triple-Stream Networks for Relationship-Based Captioning. In *CVPR*, 2019.

[167] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, and In So Kweon. Image Captioning with Very Scarce Supervised Data: Adversarial Semi-Supervised Learning Approach. In *EMNLP*, 2019.

[168] Hoeseong Kim, Jongseok Kim, Hyungseok Lee, Hyunsung Park, and Gunhee Kim. Viewpoint-Agnostic Change Captioning With Cycle Consistency. In *ICCV*, 2021.

[169] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear Attention Networks. In *NeurIPS*, 2018.

[170] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[171] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[172] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *NeurIPS Workshops*, 2014.

[173] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

[174] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*, 2017.

[175] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV*, 123(1):32–73, 2017.

[176] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[177] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[178] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. BabyTalk: Understanding and generating simple image descriptions. *IEEE Trans. PAMI*, 35(12):2891–2903, 2013.

[179] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[180] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2018.

[181] Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. *TACL*, 2:351–362, 2014.

[182] Iro Laina, Christian Rupprecht, and Nassir Navab. Towards unsupervised image captioning with shared multimodal embeddings. In *ICCV*, 2019.

[183] Weiyu Lan, Xirong Li, and Jianfeng Dong. Fluency-guided cross-lingual image captioning. In *ACM Multimedia*, 2017.

[184] Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Trung Bui, and Kyomin Jung. UMIC: An Unreferenced Metric for Image Captioning via Contrastive Learning. In *ACL*, 2021.

[185] Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. ViLBERTScore: Evaluating Image Caption Using Vision-and-Language BERT. In *EMNLP Workshops*, 2020.

[186] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked Cross Attention for Image-Text Matching. In *ECCV*, 2018.

[187] Kuang-Huei Lee, Hamid Palangi, Xi Chen, Houdong Hu, and Jianfeng Gao. Learning visual relation priors for image-text matching and image captioning with neural scene graph generators. *arXiv preprint arXiv:1909.09953*, 2019.

[188] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*, 2020.

[189] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020.

[190] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled Transformer for Image Captioning. In *ICCV*, 2019.

[191] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual Semantic Reasoning for Image-Text Matching. In *ICCV*, 2019.

[192] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *ICCV*, 2019.

[193] Siming Li, Girish Kulkarni, Tamara Berg, Alexander Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *CoNLL*, 2011.

[194] Xiangyang Li, Shuqiang Jiang, and Jungong Han. Learning object context for dense captioning. In *AAAI*, 2019.

[195] Xirong Li, Chaoxi Xu, Xiaoxu Wang, Weiyu Lan, Zhengxiong Jia, Gang Yang, and Jieping Xu. COCO-CN for cross-lingual image tagging, captioning, and retrieval. *IEEE Trans. Multimedia*, 21(9):2347–2360, 2019.

[196] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[197] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Pointing novel objects in image captioning. In *CVPR*, 2019.

[198] Xiaodan Liang, Zhiting Hu, Hao Zhang, Chuang Gan, and Eric P Xing. Recurrent topic-transition GAN for visual paragraph generation. In *ICCV*, 2017.

[199] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshops*, 2004.

[200] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[201] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[202] Fenglin Liu, Yuanxin Liu, Xuancheng Ren, Xiaodong He, and Xu Sun. Aligning visual regions and textual concepts for semantic-grounded image representations. In *NeurIPS*, 2019.

[203] Fenglin Liu, Xuancheng Ren, Xian Wu, Shen Ge, Wei Fan, Yuexian Zou, and Xu Sun. Prophet Attention: Predicting Attention with Future Attention. In *NeurIPS*, 2020.

[204] Fenglin Liu, Xian Wu, Shen Ge, Wei Fan, and Yuexian Zou. Exploring and Distilling Posterior and Prior Knowledge for Radiology Report Generation. In *CVPR*, 2021.

[205] Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. Visual News: Benchmark and Challenges in News Image Captioning. In *EMNLP*, 2021.

[206] Lixin Liu, Jiajun Tang, Xiaojun Wan, and Zongming Guo. Generating diverse and descriptive image captions using visual paraphrases. In *ICCV*, 2019.

[207] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved Image Captioning via Policy Gradient Optimization of SPIDEr. In *ICCV*, 2017.

[208] Wei Liu, Sihan Chen, Longteng Guo, Xinxin Zhu, and Jing Liu. CPTR: Full Transformer Network for Image Captioning. *arXiv preprint arXiv:2101.10804*, 2021.

[209] Xiaoxiao Liu, Qingyang Xu, and Ning Wang. A survey on deep neural network-based image captioning. *The Visual Computer*, 35(3):445–470, 2019.

[210] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[211] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

[212] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[213] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020.

[214] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

[215] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural Baby Talk. In *CVPR*, 2018.

[216] Yadan Luo, Zi Huang, Zheng Zhang, Ziwei Wang, Jingjing Li, and Yang Yang. Curiosity-driven reinforcement learning for diverse visual paragraph generation. In *ACM Multimedia*, 2019.

[217] Yunpeng Luo, Jiayi Ji, Xiaoshuai Sun, Liujuan Cao, Yongjian Wu, Feiyue Huang, Chia-Wen Lin, and Rongrong Ji. Dual-Level Collaborative Transformer for Image Captioning. In *AAAI*, 2021.

[218] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 9:2579–2605, 2008.

[219] Shweta Mahajan and Stefan Roth. Diverse Image Captioning with Context-Object Split Latent Spaces. In *NeurIPS*, 2020.

[220] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NeurIPS*, 2014.

[221] Hui Mao, Ming Cheung, and James She. Deepart: Learning joint representations of visual arts. In *ACM Multimedia*, 2017.

[222] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). In *ICLR*, 2015.

[223] Yuzhao Mao, Chang Zhou, Xiaojie Wang, and Ruifan Li. Show and Tell More: Topic-Oriented Multi-Sentence Image Captioning. In *IJCAI*, 2018.

[224] Alessio Mascolini, S Puzzo, G Incatasciato, Francesco Ponzio, Elisa Ficarra, and Santa Di Cataldo. A Novel Proof-of-concept Framework for the Exploitation of ConvNets on Whole Slide Images. In *Progresses in Artificial Intelligence and Neural Systems*, pages 125–136. Springer, 2021.

[225] Alexander Mathews, Lexing Xie, and Xuming He. Semstyle: Learning to generate stylised image captions using unaligned text. In *CVPR*, 2018.

[226] Zihang Meng, Licheng Yu, Ning Zhang, Tamara Berg, Babak Damavandi, Vikas Singh, and Amy Bearman. Connecting What to Say With Where to Look by Modeling Human Attention Traces. In *CVPR*, 2021.

[227] Nicola Messina, Giuseppe Amato, Andrea Esuli, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet. Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders. *ACM TOMM*, 17(4), 2021.

[228] Nicola Messina, Giuseppe Amato, Fabrizio Falchi, Claudio Gennaro, and Stéphane Marchand-Maillet. Towards efficient cross-modal visual textual retrieval using transformer-encoder deep features. In *CBMI*, 2021.

[229] Nicola Messina, Fabrizio Falchi, Andrea Esuli, and Giuseppe Amato. Transformer reasoning network for image-text matching and retrieval. In *ICPR*, 2021.

[230] Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alexander Berg, Tamara Berg, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *ACL*, 2012.

[231] Takashi Miyazaki and Nobuyuki Shimizu. Cross-lingual image caption generation. In *ACM Multimedia*, 2016.

[232] Ron Mokady, Amir Hertz, and Amit H Bermano. ClipCap: CLIP Prefix for Image Captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[233] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[234] Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004.

[235] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011.

[236] Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu, and Christos Faloutsos. Automatic image captioning. In *ICME*, 2004.

[237] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. X-Linear Attention Networks for Image Captioning. In *CVPR*, 2020.

[238] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[239] Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Towards personalized image captioning via multimodal memory networks. *IEEE Trans. PAMI*, 2018.

[240] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust Change Captioning. In *CVPR*, 2019.

[241] Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. Areas of Attention for Image Captioning. In *ICCV*, 2017.

[242] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *EMNLP*, 2014.

[243] Eric Phizicky, Philippe IH Bastiaens, Heng Zhu, Michael Snyder, and Stanley Fields. Protein analysis on a proteomic scale. *Nature*, 422(6928):208–215, 2003.

[244] Vittorio Pipoli, Mattia Cappelli, Alessandro Palladini, Carlo Peluso, Marta Lovino, and Elisa Ficarra. Predicting gene expression levels from DNA sequences and post-transcriptional information with transformers. *Computer Methods and Programs in Biomedicine*, page 107035, 2022.

[245] Przemysław Pobrotyn, Tomasz Bartczak, Mikołaj Synowiec, Radosław Białobrzeski, and Jarosław Bojar. Context-aware learning to rank with self-attention. *arXiv preprint arXiv:2005.10084*, 2020.

[246] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *ECCV*, 2020.

[247] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. ImageBERT: Cross-modal Pre-training with Large-scale Weak-supervised Image-Text Data. *arXiv preprint arXiv:2001.07966*, 2020.

[248] Yu Qin, Jiajun Du, Yonghua Zhang, and Hongtao Lu. Look Back and Predict Forward in Image Captioning. In *CVPR*, 2019.

[249] Leigang Qu, Meng Liu, Da Cao, Liqiang Nie, and Qi Tian. Context-Aware Multi-View Summarization Network for Image-Text Matching. In *ACM Multimedia*, 2020.

[250] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[251] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[252] Vasili Ramanishka, Abir Das, Jianming Zhang, and Kate Saenko. Top-down visual saliency guided by captions. In *CVPR*, 2017.

[253] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv preprint arXiv:2102.12092*, 2021.

[254] Arnau Ramisa, Fei Yan, Francesc Moreno-Noguer, and Krystian Mikolajczyk. BreakingNews: article Annotation by Image and Text Processing. *IEEE Trans. PAMI*, 40(5):1072–1085, 2017.

[255] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016.

[256] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[257] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[258] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[259] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[260] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. PAMI*, 39(6):1137–1149, 2017.

[261] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*, 2017.

[262] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[263] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object Hallucination in Image Captioning. In *EMNLP*, 2018.

[264] Fawaz Sammani and Luke Melas-Kyriazi. Show, edit and tell: A framework for editing image captions. In *CVPR*, 2020.

[265] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. A Style-Aware Content Loss for Real-time HD Style Transfer. In *ECCV*, 2018.

[266] Shankha Satpathy, Karsten Krug, Pierre M Jean Beltran, Sara R Savage, Francesca Petralia, Chandan Kumar-Sinha, Yongchao Dou, Boris Reva, M Harry Kane, Shayan C Avanessian, et al. A proteogenomic portrait of lung squamous cell carcinoma. *Cell*, 2021.

[267] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *ACL*, 2016.

[268] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-Consistency for Robust Visual Question Answering. In *CVPR*, 2019.

[269] Naeha Sharif, Uzair Nadeem, Syed Afaq Ali Shah, Mohammed Bennamoun, and Wei Liu. Vision to Language: Methods, Metrics and Datasets. In *Machine Learning Paradigms*, pages 9–62. 2020.

[270] Naeha Sharif, Lyndon White, Mohammed Bennamoun, and Syed Afaq Ali Shah. NNEval: Neural network based evaluation metric for image captioning. In *ECCV*, 2018.

[271] Himanshu Sharma, Manmohan Agrahari, Sujeet Kumar Singh, Mohd Firoj, and Ravi Kumar Mishra. Image captioning: a comprehensive survey. In *PARC*, 2020.

[272] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[273] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How Much Can CLIP Benefit Vision-and-Language Tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[274] Xi Shen, Alexei A Efros, and Aubry Mathieu. Discovering Visual Patterns in Art Collections with Spatially-consistent Feature Learning. In *CVPR*, 2019.

[275] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *ICCV*, 2017.

[276] Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. Finding It at Another Side: A Viewpoint-Adapted Matching Encoder for Change Captioning. In *ECCV*, 2020.

[277] Zhan Shi, Xu Zhou, Xipeng Qiu, and Xiaodan Zhu. Improving Image Captioning with Better Use of Captions. In *ACL*, 2020.

[278] Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, and Jason Weston. Engaging image captioning via personality. In *CVPR*, 2019.

[279] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. TextCaps: a Dataset for Image Captioning with Reading Comprehension. In *ECCV*, 2020.

[280] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[281] Yale Song and Mohammad Soleymani. Polysemous Visual-Semantic Embedding for Cross-Modal Retrieval. In *CVPR*, 2019.

[282] Yuqing Song, Shizhe Chen, Yida Zhao, and Qin Jin. Unpaired cross-lingual image caption generation with self-supervised rewards. In *ACM Multimedia*, 2019.

[283] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. WIT: Wikipedia-based Image Text Dataset for Multimodal Multilingual Machine Learning. *arXiv preprint arXiv:2103.01913*, 2021.

[284] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Silvia Cascianelli, Giuseppe Fiameni, and Rita Cucchiara. From Show to Tell: A Survey on Deep Learning-based Image Captioning. *IEEE Trans. PAMI*, 2022.

[285] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. Artpedia: A New Visual-Semantic Dataset with Visual and Contextual Sentences in the Artistic Domain. In *ICIAP*, 2019.

[286] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. A novel attention-based aggregation function to combine vision and language. In *ICPR*, 2021.

[287] Gjorgji Strezoski and Marcel Worring. Omniart: A large-scale artistic benchmark. *ACM TOMM*, 2018.

[288] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *ICLR*, 2020.

[289] Yusuke Sugano and Andreas Bulling. Seeing with Humans: Gaze-Assisted Neural Image Captioning. *arXiv preprint arXiv:1608.05203*, 2016.

[290] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.

[291] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[292] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[293] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019.

[294] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[295] Hamed R Tavakoli, Rakshith Shetty, Ali Borji, and Jorma Laaksonen. Paying attention to descriptions generated by image captioning models. In *ICCV*, 2017.

[296] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[297] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*, 2016.

[298] Matteo Tomei, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. What was Monet seeing while painting? Translating artworks to photo-realistic images. In *ECCV Workshops*, 2018.

[299] Matteo Tomei, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Art2Real: Unfolding the Reality of Artworks via Semantically-Aware Image-to-Image Translation. In *CVPR*, 2019.

[300] Matteo Tomei, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Image-to-Image Translation to Unfold the Reality of Artworks: an Empirical Analysis. In *ICIAP*, 2019.

[301] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[302] Alasdair Tran, Alexander Mathews, and Lexing Xie. Transform and Tell: Entity-Aware News Image Captioning. In *CVPR*, 2020.

[303] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning Robust Visual-Semantic Embeddings. In *ICCV*, 2017.

[304] Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. BERTTune: Fine-Tuning Neural Machine Translation with BERTScore. *ACL*, 2021.

[305] Emiel Van Miltenburg, Desmond Elliott, and Piek Vossen. Measuring the diversity of automatic image descriptions. In *COLING*, 2018.

[306] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[307] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based Image Description Evaluation. In *CVPR*, 2015.

[308] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

[309] Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. Captioning images with diverse objects. In *CVPR*, 2017.

[310] Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David J Crandall, and Dhruv Batra. Diverse Beam Search for Improved Description of Complex Scenes. In *AAAI*, 2018.

[311] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[312] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Trans. PAMI*, 39(4):652–663, 2016.

[313] Akiyoshi Wada and Haruki Nakamura. Nature of the charge distribution in proteins. *Nature*, 293(5835):757–758, 1981.

[314] Jing Wang, Jianlong Fu, Jinhui Tang, Zechao Li, and Tao Mei. Show, Reward and Tell: Automatic Generation of Narrative Paragraph From Photo Stream by Adversarial Training. In *AAAI*, 2018.

[315] Jing Wang, Jinhui Tang, and Jiebo Luo. Multimodal Attention with Image Text Spatial Relationship for OCR-Based Image Captioning. In *ACM Multimedia*, 2020.

[316] Jing Wang, Jinhui Tang, Mingkun Yang, Xiang Bai, and Jiebo Luo. Improving OCR-based Image Captioning by Incorporating Geometrical Relationship. In *CVPR*, 2021.

[317] Josiah Wang, Pranava Madhyastha, and Lucia Specia. Object Counts! Bringing Explicit Detections Back into Image Captioning. In *NAACL*, 2018.

[318] Li Wang, Zechen Bai, Yonghua Zhang, and Hongtao Lu. Show, Recall, and Tell: Image Captioning with Recall Mechanism. In *AAAI*, 2020.

[319] Liang-Bo Wang, Alla Karpova, Marina A Gritsenko, Jennifer E Kyle, Song Cao, Yize Li, Dmitry Rykunov, Antonio Colaprico, Joseph H Rothstein, Runyu Hong, et al. Proteogenomic and metabolomic characterization of human glioblastoma. *Cancer Cell*, 39(4):509–528, 2021.

[320] Liwei Wang, Alexander G Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *NeurIPS*, 2017.

[321] Qingzhong Wang and Antoni B Chan. Describing like humans: on diversity in image captioning. In *CVPR*, 2019.

[322] Qingzhong Wang, Jia Wan, and Antoni B Chan. On Diversity in Image Captioning: Metrics and Methods. *IEEE Trans. PAMI*, 2020.

[323] Sijin Wang, Ziwei Yao, Ruiping Wang, Zhongqin Wu, and Xilin Chen. FAIEr: Fidelity and Adequacy Ensured Image Caption Evaluation. In *CVPR*, 2021.

[324] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In *ECCV*, 2018.

[325] Yaxiong Wang, Hao Yang, Xueming Qian, Lin Ma, Jing Lu, Biao Li, and Xin Fan. Position Focused Attention Network for Image-Text Matching. In *IJCAI*, 2019.

[326] Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W Cottrell. Skeleton Key: Image Captioning by Skeleton-Attribute Decomposition. In *CVPR*, 2017.

[327] Zeyu Wang, Berthy Feng, Karthik Narasimhan, and Olga Russakovsky. Towards unique and informative captioning of images. In *ECCV*, 2020.

[328] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

[329] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

[330] Keyu Wen, Xiaodong Gu, and Qingrong Cheng. Learning Dual Semantic Relations with Graph Attention for Image-Text Matching. *IEEE TCSVT*, 2020.

[331] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.

[332] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton Van Den Hengel. What Value Do Explicit High Level Concepts Have in Vision to Language Problems? In *CVPR*, 2016.

[333] Qi Wu, Peng Wang, Chunhua Shen, Anton van den Hengel, and Anthony Dick. Ask Me Anything: Free-form Visual Question Answering Based on Knowledge from External Sources. In *CVPR*, 2016.

[334] Shaomei Wu, Jeffrey Wieland, Omid Farivar, and Julie Schiller. Automatic alt-text: Computer-generated image descriptions for blind users on a social network service. In *CSCW*, 2017.

[335] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. Learning fragment self-attention embeddings for image-text matching. In *ACM Multimedia*, 2019.

[336] Yu Wu, Linchao Zhu, Lu Jiang, and Yi Yang. Decoupled Novel Object Captioner. In *ACM Multimedia*, 2018.

[337] Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou. XGPT: Cross-modal Generative Pre-Training for Image Captioning. *arXiv preprint arXiv:2003.01473*, 2020.

[338] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-Training With Noisy Student Improves ImageNet Classification. In *CVPR*, 2020.

[339] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[340] Guanghui Xu, Shuaicheng Niu, Mingkui Tan, Yucheng Luo, Qing Du, and Qi Wu. Towards Accurate Text-based Image Captioning with Content Diversity Exploration. In *CVPR*, 2021.

[341] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[342] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[343] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation. In *CVPR*, 2017.

[344] Linjie Yang, Kevin Tang, Jianchao Yang, and Li-Jia Li. Dense captioning with joint inference and visual context. In *CVPR*, 2017.

[345] Min Yang, Wei Zhao, Wei Xu, Yabing Feng, Zhou Zhao, Xiaojun Chen, and Kai Lei. Multitask learning for cross-domain image captioning. *IEEE Trans. Multimedia*, 21(4):1047–1061, 2018.

[346] Xingyi Yang, Muchao Ye, Quanzeng You, and Fenglong Ma. Writing by Memorizing: Hierarchical Retrieval-based Medical Report Generation. In *ACL-IJCNLP*, 2021.

[347] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-Encoding Scene Graphs for Image Captioning. In *CVPR*, 2019.

[348] Xu Yang, Hanwang Zhang, and Jianfei Cai. Learning to Collocate Neural Modules for Image Captioning. In *ICCV*, 2019.

[349] Xuewen Yang, Svebor Karaman, Joel Tetreault, and Alex Jaimes. Journalistic Guidelines Aware News Image Captioning. *arXiv preprint arXiv:2109.02865*, 2021.

[350] Xuewen Yang, Heming Zhang, Di Jin, Yingru Liu, Chi-Hao Wu, Jianchao Tan, Dongliang Xie, Jue Wang, and Xin Wang. Fashion Captioning: Towards Generating Accurate Descriptions with Semantic Rewards. In *ECCV*, 2020.

[351] Yezhou Yang, Ching Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[352] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: Text-Aware Pre-training for Text-VQA and Text-Caption. In *CVPR*, 2021.

[353] Zhilin Yang, Ye Yuan, Yuexin Wu, William W Cohen, and Russ R Salakhutdinov. Review Networks for Caption Generation. In *NeurIPS*, 2016.

[354] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.

[355] Benjamin Z Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2T: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010.

[356] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Incorporating Copying Mechanism in Image Captioning for Learning Novel Objects. In *CVPR*, 2017.

[357] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring Visual Relationship for Image Captioning. In *ECCV*, 2018.

[358] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Hierarchy Parsing for Image Captioning. In *ICCV*, 2019.

[359] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

[360] Yanzhi Yi, Hangyu Deng, and Jinglu Hu. Improving image captioning evaluation by considering inter references variance. In *ACL*, 2020.

[361] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, and Jing Shao. Context and attribute grounded dense captioning. In *CVPR*, 2019.

[362] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[363] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[364] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014.

[365] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019.

[366] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for fine-grained image captioning. *IEEE Trans. PAMI*, 2019.

[367] Fangfei Zhang, Weigang Ge, Guan Ruan, Xue Cai, and Tiannan Guo. Data-independent acquisition mass spectrometry-based proteomics and software tools: a glimpse in 2020. *Proteomics*, 20(17-18):1900276, 2020.

[368] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why ADAM Beats SGD for Attention Models, 2019.

[369] Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. Actor-Critic Sequence Training for Image Captioning. In *NeurIPS*, 2017.

[370] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *ICCV*, 2019.

[371] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *CVPR*, 2021.

[372] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *ICLR*, 2020.

[373] Wei Zhang, Xiaoli Xue, Chengwang Xie, Yuanyuan Li, Junhong Liu, Hailin Chen, and Guanghui Li. CEGSO: boosting essential proteins prediction by integrating protein complex, gene expression, gene ontology, subcellular localization and orthology information. *Interdisciplinary Sciences: Computational Life Sciences*, 13(3):349–361, 2021.

[374] Wei Zhang, Yue Ying, Pan Lu, and Hongyuan Zha. Learning Long-and Short-Term User Literal-Preference with Multimodal Hierarchical Transformer Network for Personalized Image Caption. In *AAAI*, 2020.

[375] Xuying Zhang, Xiaoshuai Sun, Yunpeng Luo, Jiayi Ji, Yiyi Zhou, Yongjian Wu, Feiyue Huang, and Rongrong Ji. RSTNet: Captioning with Adaptive Attention on Visual and Non-Visual Words. In *CVPR*, 2021.

[376] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable Visual Question Answering by Visual Grounding from Attention Supervision Mining. In *WACV*, 2018.

[377] Wentian Zhao, Xinxiao Wu, and Xiaoxun Zhang. MemCap: Memorizing style knowledge for image captioning. In *AAAI*, 2020.

[378] Yue Zheng, Yali Li, and Shengjin Wang. Intention oriented image captions with guiding objects. In *CVPR*, 2019.

[379] Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. Comprehensive image captioning via scene graph decomposition. In *ECCV*, 2020.

[380] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8):1171–1179, 2018.

[381] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified Vision-Language Pre-Training for Image Captioning and VQA. In *AAAI*, 2020.

[382] Yuanen Zhou, Meng Wang, Daqing Liu, Zhenzhen Hu, and Hanwang Zhang. More grounded image captioning by distilling image-text matching model. In *CVPR*, 2020.

[383] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[384] Qi Zhu, Chenyu Gao, Peng Wang, and Qi Wu. Simple is not Easy: A Simple Strong Baseline for TextVQA and TextCaps. In *AAAI*, 2021.

[385] Xinxin Zhu, Weining Wang, Longteng Guo, and Jing Liu. AutoCaption: Image Captioning with Neural Architecture Search. *arXiv preprint arXiv:2012.09742*, 2020.