Chapter 9
Cross-Modal Representation

Abstract  Cross-modal representation learning is an essential part of representation learning, which aims to learn latent semantic representations for modalities including texts, audio, images, videos, etc. In this chapter, we first introduce typical cross-modal representation models. After that, we review several real-world applications related to cross-modal representation learning including image captioning, visual relation detection, and visual question answering.

9.1 Introduction

As introduced in Wikipedia, a modality is the classification of a single independent channel of sensory input/output between a computer and a human. To be more general, modalities are different means of information exchange between human beings and the real world. The classification is usually based on the form in which information is presented to a human. Typical modalities in the real world include texts, audio, images, videos,

Cross-modal representation learning is an important part of representation learning. In fact, artificial intelligence is inherently a multi-modal task [30]. Human beings are exposed to multi-modal information every day, and it is normal for us to integrate information from different modalities and make comprehensive judgments. Furthermore, different modalities are not independent, but they have correlations more or less. For example, the judgment of a syllable is made by not only the sound we hear but also the movement of the lips and tongue of the speaker we see. An experiment in [48] shows that a voiced /ba/ with a visual /ga/ is perceived by most people as a /da/. Another example is human beings’ ability to consider the 2D image and 3D scan of the same object together and reconstruct its structure: correlations between image and scan can be found based on the fact that a discontinuity of depth in the scan usually indicates a sharp line in the image [52]. Inspired by this, it is natural for us to consider the possibility of combining inputs from multi-modalities in our artificial intelligence systems and generate cross-modal representation.

Ngiam et al. [52] explore the probability of merging multi-modalities into one learning task. The authors divide a typical machine learning task into three stages:
feature learning, supervised learning, and prediction. And they further propose four kinds of learning settings for multi-modalities:

1. **Single-modal learning**: all stages are all done on just one modality.
2. **Multi-modal fusion**: all stages are all done with all modalities available.
3. **Cross-modal learning**: in the feature learning stage, all modalities are available, but in supervised learning and prediction, only one modality is used.
4. **Shared representation learning**: in the feature learning stage, all modalities are available. In supervised learning, only one modality is used, and in prediction, a different modality is used.

Experiments show promising results for these multi-modal tasks. When more modalities are provided (such as multi-modal fusion, cross-modal learning, and shared representation learning), the performance of the system is generally better.

In the following part of this chapter, we will first introduce cross-modal representation models, which are fundamental parts of cross-modal representation learning in NLP. And then, we will introduce several critical applications, such as image captioning, visual relationship detection, and visual question answering.

### 9.2 Cross-Modal Representation

Cross-modal representation learning aims to build embeddings using information from multiple modalities. Existing cross-modal representation models involving text modality can be generally divided into two categories: (1) [30, 77] try to fuse information from different modalities into unified embeddings (e.g., visually grounded word representations). (2) Researchers also try to build embeddings for different modalities in a common semantic space, which allows the model to compute cross-modal similarity. Such cross-modal similarity can be further utilized for downstream tasks, such as zero-shot recognition [5, 14, 18, 53, 65] and cross-media retrieval [23, 55]. In this section, we will introduce these two kinds of cross-modal representation models, respectively.

#### 9.2.1 Visual Word2vec

Computing word embeddings is a fundamental task in representation learning for natural language processing. Typical word embedding models (like Word2vec [49]) are trained on a text corpus. These models, while being extremely successful, cannot discover implicit semantic relatedness between words that could be expressed in other modalities. Kottur et al. [30] provide an example: even though *eat* and *stare at* seem are unrelated from text, images might show that when people are eating something they would also tend to stare at it. This implies that considering other modalities when constructing word embeddings may help capture more implicit semantic relatedness.
9.2 Cross-Modal Representation

Two ducks in water swim walk sit

CNN

Fig. 9.1 The architecture for word embedding with global visual context

Vision, being one of the most critical modalities, has attracted attention from researchers seeking to improve word representation. Several models that incorporate visual information and improve word embeddings with vision have been proposed. We introduce two typical word representation models incorporating visual information in the following.

9.2.1.1 Word Embedding with Global Visual Context

Xu et al. [77] propose a model that makes a natural attempt to incorporate visual features. It claims that in most word representation models, only local context information (e.g., trying to predict a word using neighboring words and phrases) is considered. Global text information (e.g., the topic of the passage), on the other hand, is often neglected. This model extends a simple local context model by using visual information as global features (see Fig. 9.1).

The input of the model is an image $I$ and a sequence describing it. It is based on a simple local context language model: when we consider a certain word $w_t$ in a sequence, its local feature is the average of embeddings of words in a window, i.e., $\{w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k}\}$. The visual feature is computed directly from the image $I$ using a CNN and then used as the global feature. The local feature and the global feature are then concatenated into a vector $f$. The predicted probability of a word $w_t$ (in this blank) is the softmax normalized product of $f$ and the word embedding $w_t$:

$$o_{w_t} = w_t^T f,$$  \hspace{1cm} (9.1)

$$P(w_t|w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k}; I) = \frac{\exp(o_{w_t})}{\sum_i \exp(o_{w_i})}.$$  \hspace{1cm} (9.2)

The model is optimized by maximizing the average of log probability:
\[ \mathcal{L} = \frac{1}{T} \sum_{t=k}^{T-k} \log P(w_t | w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k}; I). \] (9.3)

The classification error will be back-propagated to local text vector (i.e., word embeddings), visual vector, and all model parameters. This accomplishes jointly learning for a set of word embeddings, a language model, and the model used for visual encoding.

### 9.2.1.2 Word Embedding with Abstract Visual Scene

Kottur et al. [30] also propose a neural model to capture fine-grained semantics from visual information. Instead of focusing on literal pixels, the abstract scene behind the vision is considered. The model takes a pair of the visual scene and a related word sequence \((I, w)\) as input. At each training step, a window is used upon the word sequence \(w\), forming a subsequence \(S_w\). All the words in \(S_w\) will be fed into the input layer using one-hot encoding, and therefore the dimension of the input layer is \(|V|\), which is also the size of the vocabulary. The words are then transformed into their embeddings, and the hidden layer is the average of all these embeddings. The size of the hidden layer is \(N_H\), which is also the dimension of the word embeddings. The hidden layer and the output layer are connected by a full connection matrix of dimension \(N_H \times N_K\) and a softmax function. The output layer can be regarded as a probability distribution over a discrete-valued function \(g(\cdot)\) of the visual scene \(I\) (details will be given in the following paragraph). The entire model is optimized by minimizing the objective function:

\[ \mathcal{L} = -\log P(g(w)|S_w). \] (9.4)

The most important part of the model is the function \(g(\cdot)\). It maps the visual scene \(I\) into the set \(\{1, 2, \ldots, N_K\}\), which indicates what kind of abstract scene it is. In practice, it is learned offline using K-means clustering, and each cluster represents the semantics of one kind of visual scenes, consequently, the word sequence \(w\), which is designed to be related to the scene.

### 9.2.2 Cross-Modal Representation for Zero-Shot Recognition

Large-scale datasets partially support the success of deep learning methods. Even though the scales of datasets continue to grow larger, and more categories are involved, the annotation of datasets is expensive and time-consuming. For many categories, there are very limited or even no instances, which restricts the scalability of recognition systems.
Zero-shot recognition is proposed to solve the problem as mentioned above, which aims to classify instances of categories that have not been seen during training. Many works propose to utilize cross-modal representation for zero-shot image classification [5, 14, 18, 53, 65]. Specifically, image representation and category representation are embedded into a common semantic space, where similarities between image and category representations can serve for further classification. For example, in such a common semantic space, the embedding of an image of cat is expected to be closer to the embedding of category cat than the embedding of category truck.

9.2.2.1 Deep Visual-Semantic Embedding

The challenge of zero-shot learning lies in the absence of instances of unseen categories, which makes it challenging to obtain well-performed classifiers of unseen categories. Frome et al. [18] present a model that utilizes both labeled images and information from the large-scale plain text for zero-shot image classification. They try to leverage semantic information from word embeddings and transfer it to image classification systems.

Their model is motivated by the fact that word embeddings incorporate semantic information of concepts or categories, which can be potentially utilized as classifiers of corresponding categories. Similar categories cluster well in semantic space. For example, in word embedding space, the nearest neighbors of the term tiger shark are similar kinds of sharks, such as bull shark, blacktip shark, sandbar shark, and oceanic whitetip shark. In addition, boundaries between different clusters are clear. The aforementioned properties indicate that word embeddings can be further utilized as classifiers for recognition systems.

Specifically, the model first pretrains word embeddings using the Skip-gram text model on large-scale Wikipedia articles. For visual feature extraction, the model pretrains a deep convolutional neural network for 1,000 object categories on ImageNet. The pretrained word embeddings and the convolutional neural network are used to initialize the proposed Deep Visual-Semantic Embedding model (DeViSE).

To train the proposed model, they replace the softmax layer of the pretrained convolutional neural network with a linear projection layer. The model is trained to predict the word embeddings of categories for images using a hinge ranking loss:

$$\mathcal{L}(I, y) = \sum_{j \neq y} \max[0, \gamma - w_y^\top M I + w_j^\top M I],$$

(9.5)

where $w_y$ and $w_j$ are the learned word embeddings of the positive label and sampled negative label, respectively. $I$ denotes the feature of the image obtained from the convolutional neural network, $M$ is the trainable parameters in linear projection layer, and $\gamma$ is a hyperparameter in hinge ranking loss. Given an image, the objective requires the model to produce a higher score for the correct label than randomly chosen labels, where the score is defined as the dot product of the projected image feature and word embedding of terms.
At test time, given a test image, the score of each possible category is obtained using the same approach during training. Note that a crucial difference at test time is that the classifiers (word embeddings) are expanded to all possible categories, including unseen categories. Thus the model is capable of predicting unseen categories.

Experiment results show that DeViSE can make zero-shot predictions with more semantically reasonable errors, which means that even if the prediction is not exactly correct, it is semantically related to the ground truth class. However, a drawback is that although the model can utilize semantic information in word embeddings to make zero-shot image classification, using word embeddings as classifiers restricts the flexibility of the model, which results in inferior performance in the original 1,000 categories compared to the original softmax classifier.

9.2.2.2 Convex Combination of Semantic Embeddings

Inspired by DeViSE, [53] proposes a model ConSE that tries to utilize semantic information from word embeddings for zero-shot classification. A vital difference to DeViSE is that they obtain the semantic embedding of test image using a convex combination of word embeddings of seen categories. The score of the corresponding category determines the weights of the composing word embeddings.

Specifically, they train a deep convolutional neural network on seen categories. At test time, given a test image $I$ (possibly from unseen categories), they obtain the top $T$ confident predictions of seen categories, where $T$ is a hyperparameter. Then the semantic embedding $f(I)$ of $I$ is determined by the convex combination of semantic embeddings of the top $T$ confident categories, which can be formally defined as follows:

$$f(I) = \frac{1}{Z} \sum_{t=1}^{T} P(\hat{y}_0(I, t) \mid I) \cdot w(\hat{y}_0(I, t)),$$  \hspace{1cm} (9.6)

where $\hat{y}_0(I, t)$ is the $t$th most confident training label for $I$, $w(\hat{y}_0(I, t))$ is the semantic embedding (word embedding) of $\hat{y}_0(I, t)$, and $Z$ is a normalization factor given by

$$Z = \sum_{t=1}^{T} P(\hat{y}_0(I, t) \mid I).$$  \hspace{1cm} (9.7)

After obtaining the semantic embedding $f(I)$, the score of the category $m$ is given by the cosine similarity of $f(I)$ and $w(m)$.

The motivation of ConSE is that they assume novel categories can be modeled as the convex combination of seen categories. If the model is highly confident about a prediction, (i.e., $P(\hat{y}_0(I, 1) \mid I) \approx 1$), the semantic embedding $f(I)$ will be close to $w(\hat{y}_0(I, 1))$. If the predictions are ambiguous, (e.g., $P(\text{tiger} \mid I) = 0.5$, $P(\text{lion} \mid I) = 0.5$), the semantic embedding $f(I)$ will be between $w(\text{lion})$ and $w(\text{tiger})$. And they expect the semantic embedding $f(I) = 0.5w(\text{lion}) +
0.5\(w(\text{tiger})\) to be close to the semantic embedding \(w(\text{liger})\) (a hybrid cross between lions and tigers).

Although ConSE and DeViSE share many similarities, there are also some crucial differences. DeViSE replaces the softmax layer of the pretrained visual model with a projection layer, while ConSE preserves the softmax layer. ConSE does not need to be further trained and uses a convex combination of semantic embeddings to perform zero-shot classification at test time. Experiment results show that ConSE outperforms DeViSE on unseen categories, indicating better generalization capability. However, the performance of ConSE on seen categories is not as competitive as DeViSE and the original softmax classifier.

### 9.2.2.3 Cross-Modal Transfer

Socher et al. \cite{65} present a cross-modal representation model for zero-shot recognition. In their model, all word vectors are initialized with pretrained 50-dimensional word vectors and are kept fixed during training. Each image is represented by a vector \(I\) constructed by a deep convolutional neural network. They first project an image into semantic word spaces by minimizing

\[
\mathcal{L}(\Theta) = \sum_{y \in Y_s} \sum_{I(i) \in X_y} \|w_y - \theta^{(2)} f(\theta^{(1)} I(i))\|^2, \tag{9.8}
\]

where \(Y_s\) denotes the set of images’ classes which can be seen in training data, \(X_y\) denotes the set of images’ vectors of class \(y\), \(w_y\) denotes the word vector of class \(y\), and \(\Theta = (\theta^{(1)}, \theta^{(2)})\) denotes parameters of the 2-layer neural network with \(f(\cdot) = \tanh(\cdot)\) as activation function.

They observe that instances from unseen categories are usually outliers of the complete data manifold. Following this observation, they first classify an instance into seen and unseen categories via outlier detection methods. Then the instance is classified using corresponding classifiers.

Formally, they marginalize a binary random variable \(V \in \{s, u\}\) which denotes whether an instance belongs to seen categories or unseen categories separately, which means probability is given as

\[
P(y|I) = \sum_{V \in \{s, u\}} P(V|I) P(y|V, I) P(V|I). \tag{9.9}
\]

For seen image classes, they simply use softmax classifier to determine \(P(y|s, I)\), while for unseen classes, they assume an isometric Gaussian distribution around each of the novel class word vectors and assign classes based on their likelihood. To detect novelty, they calculate a Local Outlier Probability by Gaussian error function.
9.2.3 Cross-Modal Representation for Cross-Media Retrieval

Learning cross-modal representation from different modalities in a common semantic space allows one to easily compute cross-modal similarities, which can facilitate many important cross-modal tasks, such as cross-media retrieval. With the rapid growth of multimedia data such as text, image, video, and audio on the Internet, the need to retrieve information across different modalities has become stronger. Cross-media retrieval is an important task in the multimedia area, which aims to perform retrieval across different modalities such as text and image. For example, a user may submit an image of a white horse, and retrieve relevant information from different modalities, such as textual descriptions of horses, and vice versa.

A significant challenge of cross-modal retrieval is the domain discrepancies between different modalities. Besides, for a specific area of interest, cross-modal data can be insufficient, which limits the performance of existing cross-modal retrieval methods. Many works have focused on the challenges as mentioned above in cross-modal retrieval [23, 24].

9.2.3.1 Cross-Modal Hybrid Transfer Network

Huang et al. [24] present a framework that tries to relieve the cross-modal data sparsity problem by transfer learning. They propose to leverage knowledge from a large-scale single-modal dataset to boost the model training on the small-scale dataset. The massive auxiliary dataset is denoted as the source domain, and the small-scale dataset of interest is denoted as the target domain. In their work, they adopt ImageNet [12], a large-scale image database as the source domain.

Formally, a training set consists of data from source domain $\text{Src} = \{(I_p^s, y_p^s)\}_{p=1}^P$ and target domain $\text{Tar}_{tr} = \{(I_j^s, t_j^s), y_j^s\}_{j=1}^J$, where $(I, t)$ is the image/text pair with label $y$. Similarly, a test set can be denoted as $\text{Tar}_{te} = \{(I_m^m, t_m^m), y_m^m\}_{m=1}^M$. The goal of their model is to transfer knowledge from $\text{Src}$ to boost the model performance on $\text{Tar}_{te}$ for cross-media retrieval.

Their model consists of a modal-sharing transfer subnetwork and a layer-sharing correlation subnetwork. In modal-sharing transfer subnetwork, they adopt the convolutional layers of AlexNet [32] to extract image features for source and target domains, and use word vectors to obtain text features. The image and text features pass through two fully connected layers, where single-modal and cross-modal knowledge transfer are performed.

Single-modal knowledge transfer aims to transfer knowledge from images in the source domain to images in the target domain. The main challenge is the domain discrepancy between the two image datasets. They propose to solve the domain discrepancy problem by minimizing the Maximum Mean Discrepancy (MMD) of image modality between the source and target domains. MMD is calculated in a layer-wise style in the fully connected layers. By minimizing MMD in reproduced kernel Hilbert space, the image representations from source and target domains are
encouraged to have the same distribution, so knowledge from images in the source domain is expected to transfer to images in the target domain. Besides, the image encoder in the source domain is also fine-tuned by optimizing softmax loss on labeled image instances.

Cross-modal knowledge transfer aims to transfer knowledge between image and text in the target domain. Text and image representations from an annotated pair in the target domain are encouraged to be close to each other by minimizing their Euclidean distance. The cross-modal transfer loss of image and text representations is also computed in a layer-wise style in the fully connected layers. The domain discrepancy between image and text modalities is expected to be reduced in high-level layers.

In layer-sharing correlation subnetwork, representations from modal-sharing transfer subnetwork in the target domain are fed into shared fully connected layers to obtain the final common representation for both image and text. As the parameters are shared between two modalities, the last two fully connected layers are expected to capture the cross-modal correlation. Their model also utilizes label information in the target domain by minimizing softmax loss on labeled image/text pairs. After obtaining the final common representations, cross-media retrieval can be achieved by simply computing the nearest neighbors in semantic space.

### 9.2.3.2 Deep Cross-Media Knowledge Transfer

As an extension of [23, 24] also focuses on dealing with domain discrepancy and insufficient cross-modal data for cross-media retrieval in specific areas, Huang and Peng [23] present a framework that transfers knowledge from a large-scale cross-media dataset (source domain) to boost the model performance on another small-scale cross-media dataset (target domain).

A crucial difference from [24] is that the dataset in the source domain also consists of image/text pairs with label annotations instead of a single-modal setting in [24]. Since both domains contain image and text media types, domain discrepancy comes from the media-level discrepancy in the same media type, and correlation-level discrepancy in image/text correlation patterns between different domains. They propose to transfer intra-media semantic and inter-media correlation knowledge by jointly reducing domain discrepancies on media-level and correlation-level.

To extract the distributed features for different media types, they adopt VGG19 [63] for image encoder and Word CNN [29] for text encoder. The two domains have the same architecture but do not share parameters. The extracted image/text features pass through two fully connected layers, respectively, where the media-level transfer is performed. Similar to [24], they reduce domain discrepancies within the same modalities by minimizing Maximum Mean Discrepancy (MMD) between the source and target domains. The MMD is computed in a layer-wise style to transfer knowledge within the same modalities. They also minimize Euclidean distance between image/text representations pairs in both source and target domains to preserve the semantic information across modalities.
Correlation-level transfer aims to reduce domain discrepancy in image/text correlation patterns in different domains. In two domains, both image and text representations share the last two fully connected layers to obtain the common representation for each domain. They optimize layer-wise MMD loss between the shared fully connected layers in different domains for correlation-level knowledge transfer, which encourages source and target domains to have the same image/text correlation patterns. Finally, both domains are trained with label information of image/text pairs. Note that the source domain and target domain do not necessarily share the same label set.

In addition, they propose a progressive transfer mechanism, which is a curriculum learning method aiming to promote the robustness of the model training. This is achieved by selecting easy samples for model training in the early period, and gradually increases the difficulty during the training. The difficulty of training samples is measured according to the bidirectional cross-media retrieval consistency.

9.3 Image Captioning

Image captioning is the task of automatically generating natural language descriptions for images. It is a fundamental task in artificial intelligence, which connects natural language processing and computer vision. Compared with other computer vision tasks, such as image classification and object detection, image captioning is significantly harder for two reasons: first, not only objects but also relationships between them have to be detected; second, besides basic judgments and classification, natural language sentences have to be generated.

Traditional methods for image captioning are usually using retrieval models or generation models, of which the ability to generalize is comparatively weaker compared with that of novel deep neural network models. In this section, we will introduce several typical models of both genres in the following.

9.3.1 Retrieval Models for Image Captioning

The primary pipeline of retrieval models is (1) represent images and/or sentences using special features; (2) for new images and/or sentences, search for probable candidates according to the similarity of features.

Linking words to images has a rich history, and [50] (a retrieval model) is the first image annotation system. This paper tries to build a keyword assigning system for images from labeled data. The pipeline is as follows:

(1) Image segmentation. Every image is divided into several parts, using the simplest rectangular division. The reason for doing so is that an image is typically annotated with multiple labels, each of which often corresponds to only a part of it. Segmentation would help reduce noises in labeling.
Feature extraction. Features of every part of the image are extracted.

Clustering. Feature vectors of image segments are divided into several clusters. Each cluster accumulates word frequencies and thereby calculates word likelihood. Concretely,

\[ P(w_i | c_j) = \frac{P(c_j | w_i) P(w_i)}{\sum_k P(c_j | w_k) P(w_k)} = \frac{n_{ji}}{N_j}, \quad (9.10) \]

where \( n_{ji} \) is the number of times word \( w_i \) appears in cluster \( j \), and \( N_j \) is the number of times that all words appear in cluster \( j \). The calculation is based on using frequencies as probabilities.

Inference. For a new image, the model divides it into segments, extracts features for every part, and finally, aggregates keywords assigned to every part to obtain the final prediction.

The key idea of this model is image segmentation. Take a landscape picture, for instance, there are two parts: mountain and sky, and both parts will be annotated with both labels. However, if another picture has two parts mountain and river, the two mountain parts would hopefully be in the same cluster and discover that they share the same label mountain. In this way, labels can be assigned to the correct part of the image, and noises could be alleviated.

Another typical retrieval model is proposed by [17], which can assign a linking score between an image and a sentence. An intermediate space of meaning calculates this score of linking. The representation of the meaning space is a triple in the form of \( \langle object, action, scene \rangle \). Each slot of the triple has a finite discrete candidate set. The problem of mapping images and sentences into the meaning space involves solving a Markov random field.

Different from the previous model, this system can do not only image caption, but also do the inverse, that is, given a sentence, the model provides certain probable associated images. At the inference stage, the image (sentence) is first mapped to the intermediate meaning space, then we search in the pool for the sentence (image) that has the best matching score.

After that, researchers also proposed a lot of retrieval models which consider different kinds of characteristics of the images, such as [21, 28, 34].

### 9.3.2 Generation Models for Image Captioning

Different from the retrieval-based model, the basic pipeline of generation models is (1) use computer vision techniques to extract image features, (2) generate sentences from these features using methods such as language models or sentence templates.

Kulkarni et al. [33] propose a system that makes a tight connection between the particular image and the sentence generating process. The model uses visual detectors to detect specific objects, as well as attributes of a single object and relationships between multiple objects. Then it constructs a conditional random field to incorporate unary image potentials and higher order text potentials and thereby predicts labels.
for the image. Labels predicted by conditional random fields (CRF) is arranged as a triple, e.g., ⟨⟨white, cloud⟩, in, ⟨blue, sky⟩⟩.

Then sentences are generated according to the labels. There are two ways to build a sentence based on the triple skeleton. (1) The first is to use an n-gram language model. For example, when trying to decide whether or not to put a glue word \( x \) between a pair of meaningful words (which means they are inside the triple) \( a \) and \( b \), the probabilities \( \hat{p}(axb) \) and \( \hat{p}(ab) \) are compared for the decision. \( \hat{p} \) is the standard length-normalized probability of the n-gram language model. (2) The second is to use a set of descriptive language templates, which alleviates the problem of grammar mistakes in the language model.

Further, [16] proposes a novel framework to explicitly represent the relationship between image structure and its caption sentence’s structure. The method, Visual Dependency Representation, detects objects in the image, and detects the relationship between these objects based on the proposed Visual Dependency Grammar, which includes eight typical relations like beside or above. Then the image can be arranged as a dependency graph, where nodes are objects and edges are relations. This image dependency graph can be aligned with the syntactic dependency representation of the caption sentence. The paper further provides four templates to generating descriptive sentences from the extracted dependency representation.

Besides these two typical works, there are massive generation models for image captioning, such as [15, 35, 78].

### 9.3.3 Neural Models for Image Captioning

In [33], it was claimed in 2011 that in image captioning tasks: *Natural language generation still remains an open research problem. Most previous work is based on retrieval and summarization.* From 2015, inspired by advances in neural language model and neural machine translation, a number of end-to-end neural image captioning models based on the encoder-decoder system have been proposed. These new models significantly improve the ability to generate natural language descriptions.

#### 9.3.3.1 The Basic Model

Traditional machine translation models typically stitch many subtasks together, such as individual word translation and reordering, to perform sentence and paragraph translation. Recent neural machine translation models, such as [8], use a single encoder-decoder model, which can be optimized by stochastic gradient descent conveniently. The task of image captioning is inherently analogous to machine translation because it can also be regarded as a translation task, where the source “language” is an image. The encoders and decoders used for machine translations are typically RNNs, which is a natural selection for sequences of words. For image captioning, CNN is chosen to be the encoder, and RNN is still used as the decoder.
Vinyals et al. [70] is the most typical model which uses encoder-decoder for image captioning (see Fig. 9.2). Concretely, a CNN model is used to encode the image into a fixed length vector, which is believed to contain the necessary information for captioning. With this vector, an RNN language model is used to generate natural language descriptions, and this is the decoder. Here, the decoder is similar to the LSTM used for machine translation. The first unit takes the image vector as the input vector, and the rest units take the previous word embedding as input. Each unit outputs a vector $o$ and passes a vector to the next unit. $o$ is further fed into a softmax layer, whose output $p$ is the probability of each word within the vocabulary. The ways to deal with these calculated probabilities are different in training and testing:

**Training.** These probabilities $p$ are used to calculate the likelihood of the provided description sentences. Considering the nature of RNNs, it is easy to model the joint probability into conditional probabilities.

$$\log P(s|I) = \sum_{t=0}^{N} \log P(w_t|I, w_0, \ldots, w_{t-1}),$$  \hspace{0.5cm} (9.11)

where $s = \{w_1, w_2, \ldots, w_N\}$ is the sentence and its words, $w_0$ is a special START token, and $I$ is the image. Stochastic gradient descent can thereby be performed to optimize the model.

**Testing.** There are multiple approaches to generate sentences given an image. The first one is called Sampling. For each step, the single word with the highest probability in $p$ is chosen, and used as the input of the next unit until the END token is generated or a maximal length is reached. The second one is called Beam Search. For each step (now the length of sentences is $t$), $k$ best sentences are kept. Each of them generates several new sentences of length $t + 1$, and again, only $k$ new
sentences are kept. Beam Search provides a better approximation for

\[ s^* = \arg \max_s \log P(s|I). \] (9.12)

9.3.3.2 Variants of the Basic Model

The research on image captioning tightly follows that on machine translation. Inspired by [6], which uses attention mechanism in machine translation, [76] introduces visual attention into the encoder-decoder image captioning model.

The major bottleneck of [70] is the fact that information from the image is shown to the LSTM decoder only at the first decoding unit, which actually requires the encoder to squeeze all useful information into one fixed-length vector. In contrast, [76] does not require such compression. The CNN encoder does not produce one vector for the entire image; instead, it produces \( L \) region vectors \( I_i \), each of which is the representation of a part of the image. At every step of decoding, the inputs include standard LSTM inputs (i.e., output and hidden state of last step \( o_{t-1} \) and \( h_{t-1} \)), and an input vector \( z \) from the encoder. Here, \( z \) is the weighted sum of image vectors \( I_i \): \( z = \sum_i \alpha_i I_i \), where \( \alpha_i \) is the weight computed from \( I_i \) and \( h_{t-1} \). Throughout the training process, the model learns to focus on parts of the image for generating the next word by producing larger weights \( \alpha \) on more relevant parts, as shown in Fig. 9.3.1

While the above paper uses soft attention for the image, [27] makes explicit alignment between image fragments and sentence fragments before generating a description for the image. In the first stage, the alignment stage, sentence and image fragments are aligned by being mapped to a shared space. Concretely, sentence fragments (i.e., \( n \) consecutive words) are encoded using a bidirectional LSTM into the embeddings \( s \), and image fragments (i.e., part of the image, and also the entire image) are encoded using a CNN into the embeddings \( I \). The similarity score between image \( I \) and sentence \( s \) is computed as

\[ s^* = \arg \max_s \log P(s|I). \] (9.12)

---

1The example is obtained from the implementation of Yunjey Choi (https://github.com/yunjey/show-attend-and-tell).
\[ \text{sim}(I, s) = \sum_{t \in g_I} \max_{i \in g_s}(0, I_t^\top s_i), \quad (9.13) \]

where \(g_s\) is the sentence fragment set of sentence \(s\), and \(g_I\) is the image fragment set of image \(I\). The alignment is then optimized by minimizing the ranking loss \(\mathcal{L}\) for both sentences and images:

\[ \mathcal{L} = \sum_I \left[ \sum_s \max(0, \text{sim}(I, s) - \text{sim}(I, I) + 1) + \sum_s \max(0, \text{sim}(s, I) - \text{sim}(I, I) + 1) \right]. \quad (9.14) \]

The assumption for this alignment procedure is similar to [50] (see Sect. 9.3.1): all description sentences are regarded as (possibly noisy) labels for every image section and are based on the massive training data, the model would hopefully be trained to align caption sentences to their corresponding image fragments. The second stage is similar to the basic model in [70], but the alignment results are used to provide more precise training data.

As mentioned above, [76] makes the decoder have the ability to focus attention on the different parts of the image for different words. However, there are some nonvisual words in the decoding process. For example, words such as the and of are more dependent on semantic information than visual information. Furthermore, words such as phone followed by cell or meter before near the parking are usually generated by the language model. To avoid the gradient of a nonvisual word decreasing the effectiveness of visual attention, in the process of generating captions, [43] adopts an adaptive attention model with a visual sentinel. At each time step, the model needs to determine that it depends on an image region or a visual sentinel.

Adaptive attention model [43] uses attention in the process of generating a word rather than updating the LSTM state; it utilizes “visual sentinel” vector \(x_t\) and image region vectors \(I_i\). Here, \(x_t\) is produced by the inputs and states of LSTM at time step \(t\), while \(I_i\) is provided from CNN encoder. Then the adaptive context vector \(\hat{c}_t\) is the weighted sum of \(L\) image region vectors \(I_i\) and visual sentinel \(x_t\):

\[ \hat{c}_t = \sum_{i=1}^{L} \alpha_i I_i + \alpha_{L+1} x_t, \quad (9.15) \]

where \(\alpha_i\) are the weights computed by \(I_i, x_t\), and the LSTM hidden state \(h_t\). We have \(\sum_{i=1}^{L+1} \alpha_i = 1\). Finally the probability of a word in vocabulary at time \(t\) can be calculated as a residual form:

\[ p_t = \text{Softmax}(W_p(\hat{c}_t + h_t)), \quad (9.16) \]

where \(W_p\) is a learned weight parameter.

Many existing image captioning models with attention allocate attention over image’s regions, whose size is often \(7 \times 7\) or \(14 \times 14\) decided by the last pooling
layer in CNN encoder. Anderson et al. [2] first calculate attention at the level of objects. It first employs Faster R-CNN [58] which is trained on ImageNet [60] and Genome [31] to predict attribute class, such as an open oven, green bottle, floral dress, and so on. After that, it applies attention over valid bounding boxes to get fine-grained attention for helping the caption generation.

Besides, [11] rethinks the form of latent states in image captioning, which usually compresses two-dimensional visual feature maps encoded by CNN to a one-dimensional vector as the input of the language model. They find that the language model with 2D states can preserve the spatial locality, which can link the input visual domain and output linguistic domain observed by visualizing the transformation of hidden states.

Word embeddings and hidden states in [11] are 3D tensors of size $C \times H \times W$, which means $C$ channels, each of size $H \times W$. The encoded features maps will be directly inputted to the 2D language model instead of going through an average pooling layer. In the 2D language model, the convolution operator takes the place of matrix multiplication in the 1D model, and mean pooling will be used to generate the output word probability distribution from 2D hidden states. Figure 9.4 shows activated region of a latent channel at the $t$th step. When we set a threshold for the activated regions, it is revealed that the special channels are associated with specific nouns in the decoding process, which help get a better understanding of the process of generating captions.

Traditional methods train the caption model by maximizing the likelihood of training examples, which forms a gap between the optimization objective and evaluating metrics. To alleviate the problem, [59] uses reinforcement learning to directly maximize the CIDEr metric [69]. CIDEr reflects the diversity of generated captions by giving high weights to the low-frequency $n$-grams in the training set, which demonstrates that people prefer detailed captions rather than universal ones, like a boy is playing a game. To encourage the distinctiveness of captions, [10] adopts contrastive learning. Their model learns to discriminate the caption of a given image and the caption of an alike image by maximizing the difference between ground truth positive pair and mismatch negative pair. The experiment shows that contrastive learning increases the diversity of captions significantly.

Furthermore, automatic evaluation metrics, such as BLEU [54], METEOR [13], ROUGE [38], CIDEr [69], SPICE [1], and so on, may neglect some novel expressions restrained by the ground truth captions. To better evaluate the naturalness and diversity of captions, [9] proposes a framework based on Conditional Generative Adversarial Networks, whose generator tries to achieve a higher score in the evalua-
9.3 Image Captioning

While the evaluator tries to distinguish between the generated caption and human descriptions for a given image, as well as between the given image and the mismatch description. The user study shows that the trained generator can generate natural and diverse captions than the model trained by maximum likelihood estimate, while the trained evaluator is more consistent with human’s evaluation.

Besides the works we introduced above, there are also a mass of variants of the basic encoder-decoder model such as [20, 26, 40, 45, 51, 71, 73].

9.4 Visual Relationship Detection

Visual relationship detection is the task of detecting objects in an image and understanding the relationship between them. While detecting the objects is always based on semantic segmentation or object detection methods, such as R-CNN, understanding the relationship is the key challenge of this task. While detecting visual relation with image information is intuitive and effective [25, 62, 84], leveraging information from language can further boost the model performance [37, 41, 82].

9.4.1 Visual Relationship Detection with Language Priors

Lu et al. [41] propose a model that uses language priors to enhance the performance on infrequent relationships for which sufficient training instances are hard to obtain solely from images. The overall architecture is shown in Fig. 9.5.

They first train a CNN to calculate the unnormalized relations’ probability obtained from visual inputs by

![Diagram](image)

**Fig. 9.5** The architecture of visual relationship detection with language prior
\[
P_V(R_{i,j,k}, \Theta|\langle O_1, O_2 \rangle) = P_i(O_1)(z_k^\top \text{CNN}(O_1, O_2) + s_k)P_j(O_2),
\]

where \(P_i(O_j)\) denotes the probability that bounding box \(O_j\) is entity \(i\), and \(\text{CNN}(O_1, O_2)\) is the joint feature of box \(O_1\) with box \(O_2\). \(\Theta = \{z_k, s_k\}\) is the set of parameters.

Besides, language prior is considered in this model by calculating the unnormalized probability that the entity pair \(\langle i, j \rangle\) has the relation \(k\):

\[
P_f(R, W) = r_k^\top [w_i; w_j] + b_k,
\]

where \(w_i\) and \(w_j\) are the word embeddings of the text of subject and object, respectively, \(r_k\) is the learned relational embedding of the relation \(k\).

Given the probabilities of a relation from visual and textual inputs, respectively, the authors combine them into the integrated probability of a relation. The final prediction is the one with maximal integrated probability:

\[
R^* = \max_R P_V(R_{i,j,k}|\langle O_1, O_2 \rangle)P_f(R, W).
\]

The rank of the ground truth relationship \(R\) with bounding boxes \(O_1\) and \(O_2\) is maximized using the following rank loss function:

\[
C(\Theta, W) = \sum_{\langle O_1, O_2 \rangle, R} \max\{1 - P_V(R, \Theta|\langle O_1, O_2 \rangle)P_f(R, W) + \max_{\langle O'_1, O'_2 \rangle \neq \langle O_1, O_2 \rangle, R' \neq R} P_V(R', \Theta|\langle O'_1, O'_2 \rangle)P_f(R', W), 0\}. 
\]

In addition to the loss that optimizes the rank of the ground truth relationships, the authors also propose two regularization functions for language priors. The final loss function of this model is defined as

\[
\mathcal{L} = C(\Theta, W) + \lambda_1 L(W) + \lambda_2 K(W).
\]

\(K(W)\) is a variance function to make the similar relationships’ corresponding \(f(\cdot)\) function closer:

\[
K(W) = \text{Var}\left[\frac{P_f(R, W) - P_f(R', W)}{d(R, R')}\right], \forall R, R',
\]

where \(d(R, R')\) is the sum of the cosine distances (in Word2vec space) between the two objects and the predicates of the two relationships \(R\) and \(R'\).

\(L(W)\) is a function to encourage less-frequent relation to have a lower \(f(\cdot)\) score. When \(R\) occurs more frequently than \(R'\), we have

\[
L(W) = \sum_{R, R'} \max\{P_f(R', W) - P_f(R, W) + 1, 0\}.
\]
9.4 Visual Relationship Detection

![Diagram of VTransE model](image)

**Fig. 9.6** The architecture of VTransE model

- **Object Detection Module**
- **Relation Prediction Module**
- Yannan Extraction Layer
- **CNN**

(a) A scene.  
(b) The corresponding scene graph.

**Fig. 9.7** An illustration for scene graph generation

### 9.4.2 Visual Translation Embedding Network

Inspired by recent progress in knowledge representation learning, [82] proposes VTransE, a visual translation embedding network. Objects and the relationship between objects are modeled as TransE [7] like vector translation. VTransE first projects subject and object into the same space as relation translation vector \( r \in \mathbb{R}^r \). Subject and object could be denoted as \( x_s, x_o \in \mathbb{R}^M \) in the feature space, where \( M \gg r \). Similar to TransE relationship, VTransE establishes a relationship as

\[
W_s x_s + r \sim W_o x_o, \tag{9.24}
\]

where \( W_s \) and \( W_o \) are projection matrices. The overall architecture is shown in Fig. 9.6.

### 9.4.3 Scene Graph Generation

Li et al. [37] further formulate visual relation detection as a scene graph generation task, where nodes correspond to objects and directed edges correspond to visual relations between objects, as shown in Fig. 9.7.

This formulation allows [37] to leverage different levels of context information, such as information from objects, phrases (i.e., \( \langle \text{subject}, \text{predicate}, \text{object} \rangle \) triples),
and region captions, to boost the performance of visual relation detection. Specifically, [37] proposes to construct a graph that aligns these three levels of information and perform feature refinement via message passing, as shown in Fig. 9.8. By leveraging complementary information from different levels, the performances of different tasks are expected to be mutually improved.

**Dynamic Graph Construction.** Given an image, they first generate three kinds of proposals that correspond to three kinds of nodes in the proposed graph structure. The proposals include object proposals, phrase proposals, and region proposals. The object and region proposals are generated using Region Proposal Network (RPN) [57] trained with ground truth bounding boxes. Given \( N \) object proposals, phrase proposals are constructed based on \( N(N - 1) \) object pairs that fully connect the object proposals with direct edges, where each direct edge represents a potential phrase between an object pair.

Each phrase proposal is connected to the corresponding subject and object with two directed edges. A phrase proposal and a region proposal are connected if their overlap exceeds a certain fraction (e.g., 0.7) of the phrase proposal. There are no direct connections between objects and regions since they can be indirectly connected via phrases.

**Feature Refinement.** After obtaining the graph structure of different levels of nodes, they perform feature refinement by iterative message passing. The message passing procedure is divided into three parallel stages, including object refinement, phrase refinement, and region refinement.

In object feature refinement, the object proposal feature is updated with gated features from adjacent phrases. Given an object \( i \), the aggregated feature from regions that are linked to object \( i \) via subject-predicate edges \( \tilde{x}_{i}^{p \rightarrow s} \) can be defined as follows:

\[
\tilde{x}_{i}^{p \rightarrow s} = \frac{1}{\|E_{i,p}\|} \sum_{(i,j) \in E_{i,p}} f_{(o,p)}(\tilde{x}_{i}^{(o)}, \tilde{x}_{j}^{(p)}) \tilde{x}_{j}^{(p)}, \quad (9.25)
\]
where $E_{s,p}$ is the set of subject predicate connections, and $E_{i,p}$ denotes the number of phrases connected with the object $i$ as the subject predicate pairs. $f_{(o,p)}$ is a learnable gate function that controls the weights of information from different sources:

$$f_{(o,p)}(x^{(o)}_i, x^{(p)}_j) = \sum_{k=1}^K \text{Sigmoid}(\omega^{(k)}_{(o,p)} \cdot [x^{(o)}_i; x^{(p)}_j]),$$

(9.26)

where $\omega^{(k)}_{(o,p)}$ is a gate template used to calculate the importance of the information from a subject-predicate edge and $K$ is the number of templates. The aggregated feature from object-predicate edges $\hat{x}^{p\rightarrow o}_i$ can be similarly computed.

After obtaining information $\hat{x}^{p\rightarrow s}_i$ and $\hat{x}^{p\rightarrow o}_i$ from adjacent phrases, the object refinement at time step $t$ can be defined as follows:

$$x^{(o)}_{i,t+1} = x^{(o)}_{i,t} + f^{(p\rightarrow s)}(\hat{x}^{p\rightarrow s}_i) + f^{(p\rightarrow o)}(\hat{x}^{p\rightarrow o}_i),$$

(9.27)

where $f(\cdot) = \text{WReLU}(\cdot)$, $\text{W}$ is a learnable parameter and not shared between $f^{(p\rightarrow s)}(\cdot)$ and $f^{(p\rightarrow o)}(\cdot)$.

The refinement scheme of phrases and regions is similar to objects. The only difference is the information sources: Phrase proposals receive information from adjacent objects and regions, and region proposals receive information from phrases.

After feature refinement via iterative message passing, the feature of different levels of nodes can be used for corresponding tasks. Region features can be used as the initial state of a language model to generate region captions. Phrase features can be used to predict visual relation between objects, which composes of the scene graph of the image.

In comparison with scene graph generation methods that model the dependencies between relation instances by attention mechanism or message passing, [47] decomposes the scene graph task into a mixture of two phases: extracting primary relations from input, and completing the scene graph with reasoning. The authors propose a Hybrid Scene Graph generator (HRE) that combines these two phases in a unified framework and generates scene graphs from scratch.

Specifically, HRE first encodes the object pair into representations and then employs a neural relation extractor resolving primary relations from inputs and a differentiable inductive logic programming model that iteratively completes the scene graph. As shown in Fig. 9.9, HRE contains two units, a pair selector and a relation predictor, and runs in an iterative way.

At each time step, the pair selector takes a look at all object pairs $P^-$ that have not been associated with a relation and chooses the next pair of entities whose relation is to be determined. The relation predictor utilizes the information contained in all pairs $P^+$ whose relations have been determined, and the contextual information of the pair to make the prediction on the relation. The prediction result is then added to $P^+$ and benefits future predictions.

To encode object pair into representations, HRE extends the union box encoder proposed by [41] by adding the object features (what are the objects) and their
locations (where are the objects) into the object pair representation, as shown in Fig. 9.10.

**Relation Predictor.** The relation predictor is composed of two modules: a neural module predicting the relations between entities based on the given context (i.e., a visual image) and a differentiable inductive logic module performing reasoning on $P^+$. Both modules predict the relation score between a pair of objects individually. The relation scores from the two modules are finally integrated by multiplication.

**Pair Selector.** The selector works as the predictor’s collaborator with the goal to figure out the next relation which should be determined. Ideally, the choice $p^*$ made by the selector should satisfy the condition that all relations that will affect the predictor’s prediction on $p^*$ should be sent to the predictor ahead of $p^*$. HRE implements the pair selector as a greedy selector which always chooses the entity pair from $P^-$ to be added to $P^+$ as the entity pair of which the relation predictor is most confident in its prediction.
It is worth noting that the task of scene graph generation resembles document-level relation extraction in many aspects. Both tasks seek to extract structured graphs consisting of entities and relations. Also, they need to model the complex dependencies between entities and relations in a rich context. We believe both tasks are worthy to explore for future research.

9.5 Visual Question Answering

Visual Question Answering (VQA) aims to answer natural language questions about an image, and can be seen as a single turn of dialogue about a picture. In this section, we will introduce widely used VQA datasets and several typical VQA models.

9.5.1 VQA and VQA Datasets

VQA was first proposed in [46]. They first propose a single-world approach by modeling the probability of an answer $a$ given question $q$ and a world $w$ by

$$P(a|q, w) = \sum_{z} P(a|z, w)P(z|q),$$

(9.28)

where $z$ is a latent variable associated with the question and the world $w$ is a representation of the image. They further extend the single-world approach to a multi-world approach by marginalizing over different segments $s$ of the given image. The probability of an answer $a$ given question $q$ and a world $w$ is given by

$$P(a|q, s) = \sum_{w} \sum_{z} P(a|w, z)P(w|s)P(z|q).$$

(9.29)

They also release the first dataset of VQA named as DAQUAR in their paper. Besides DAQUAR, researchers also release a lot of VQA datasets with various characteristics. The most widely used dataset was released in [4], where the authors provided cases and experimental evidence to demonstrate that to answer these questions, a human or an algorithm should use features of the image and external knowledge. Figure 9.11 shows examples of VQA dataset released in [4]. It is also demonstrated that this problem cannot be solved by converting images to captions and answering questions according to captions. Experiment results show that the performance of vanilla methods is still far from human.

In fact, there are also other existing datasets for Visual QA such as Visual7W [85], Visual Madlibs [80], COCO-QA [56], and FM-IQA [19].
308 9 Cross-Modal Representation

Question: Why are the men jumping?
Answer: to catch frisbee

Question: Is the water still?
Answer: no

Question: What is the kid doing?
Answer: skateboarding

Question: What is hanging on the wall above the headboard?
Answer: pictures

Fig. 9.11 Examples of VQA dataset

9.5.2 **VQA Models**

Besides, [4, 46] further investigate approaches to solve specific types of questions in VQA. Moreover, [83] proposes an approach to solve “YES/NO” questions. Note that the model is an ensemble model of two similar models: Q-model and Tuple-model, the difference between which will be described later. The overall approach can be divided into two steps: (1) Language Parsing and (2) Visual Verification. In the former step, they extract \( \langle P, R, S \rangle \) tuples from questions first by parsing it and assigning an entity to each word. Then they summarize the parsed sentences through removing “stop words”, auxiliary verbs, and all words before a nominal subject or passive nominal subject, and further split the summary into PRS arguments according to the part of speech of phrases. The difference between Q-Model and Tuple-model is that the Q-model is the one used in their previous work [4], embedding the question into a dense 256-dim vector by LSTM, while Tuple-model is to convert \( \langle P, R, S \rangle \) tuples into 256-dim embeddings by MLP. As for the Visual Verification step, they use the same feature of images as in [39] which was encoded into the dense 256-dim vector.
by an inner-product layer followed by a $tanh$ layer. These two vectors are passed through an MLP to produce the final output (“Yes” or “No”).

Moreover, [61] proposes a method to calculate attention $\alpha_j$ by the set of image features $I = (I_1, I_2, \ldots, I_K)$ and the question embedding $q$ by

$$\alpha_j = (W_1 I_j + b_1)\top (W_2 q + b_2),$$

(9.30)

where $W_1, W_2, b_1, b_2$ are trainable parameters.

Attention based techniques are quite efficient for filtering noises that are irrelevant to the question. However, some questions are only related to some small regions, which encourages researchers to use stacked attention to further filtering noises. We refer readers to Fig. 1b in [79] for an example of stacked attention.

Yang et al. [79] further extend the attention-based model used in [61], which employs LSTMs to predict the answer. They take the question as input and attend to different regions in the image to obtain additional input. The key idea is to gradually filter out noises and pinpoint the regions that are highly relevant to the answer by reasoning through multiple stacked attention layers progressively. The stacked attention could be calculated by stacking:

$$h^k_A = \tanh(W^k_1 I \oplus (W^k_2 u^{k-1} + b^k_A)).$$

(9.31)

Note that we denote the addition of a matrix and a vector by $\oplus$. The addition between a matrix and a vector is performed by adding each column of the matrix by the vector. $u$ is a refined query vector that combines information from the question and image regions. $u^0$ (i.e., $u$ from the first attention layer with $k = 0$) could be initialized as the feature vector of the question. $h^k_A$ is then used to compute $p^k_I$, which corresponds to the attention probability of each image region,

$$p^k_I = \text{Softmax}(W^k_3 h^k_A + b^k_P).$$

(9.32)

$u^k$ could be iterated by

$$\tilde{I}^k = \sum_i p^k_I I_i,$$

(9.33)

$$u^k = u^{k-1} + \tilde{I}^k.$$  

(9.34)

That is, in every layer, the model progressively uses the combined question and image vector $u^{k-1}$ as the query vector for attending the image region to obtain the new query.

The above models attend only on images, but questions should also be attended. [44] calculates co-attention by

$$Z = \tanh(Q\top W I),$$

(9.35)
What color on the stop light is lit up?

Answer: green

Fig. 9.12 The architecture of hierarchical co-attention model

Another intuitive approach is to use external knowledge from knowledge bases, which will help us better explain the implicit information hiding behind the image. Such an approach was proposed in [75], which first encodes the image into captions and vectors representing different attributes of the image to retrieve documents about a different part of the images from knowledge bases. Documents are encoded through doc2vec [36]. The representation of captions, attributes, and documents are transformed and concatenated to form the initial vector of an LSTM, which is trained in Seq2seq fashion. Details of the model are shown in Fig. 9.13.

Neural Module Network is a framework for constructing deep networks with a dynamic computational structure, which was first proposed in [3]. In such a framework, every input is associated with a layout that provides a template for assembling an instance-specific network from a collection of shallow network fragments called...
modules. The proposed method processes the input question through two separate ways: (1) parsing and laying out several modules, and (2) encoding by an LSTM. The corresponding picture is processed by the modules laid out according to the question, the types of which are predefined, find, transform, combine, describe, and measure. The authors defined find to be a transformation from Image to Attention map, transform to be a mapping from one Attention to another, combine to be a combination of two Attention, describe to be a description relying on Image and Attention, and Measure to be a measure only relying on Attention. The model is shown in Fig. 9.14.

A key drawback of [3] is that it relies on the parser to generate modules. [22] proposes an end-to-end model to generate a sequence of Reverse-Polish expression to describe the module network, as shown in Fig. 9.15. And the overall architecture is shown in Fig. 9.16.

Graph Neural Networks (GNNs) have also been applied to VQA tasks. [68] tries to build graphs about both the scene and the question. The authors described a deep neural network to take advantage of such a structured representation. As shown in Fig. 9.17, the GNN-based VQA model could capture the relationships between words and objects.

9.6 Summary

In this chapter, we first introduce the concept of cross-modal representation learning. Cross-modal learning is essential since many real-world tasks require the ability to understand the information from different modalities, such as text and image. Next, we introduce the concept of cross-modal representation learning, which aims to exploit the links and enable better utilization of information from different modalities.
Fig. 9.15  The architecture of Reverse-Polish expression and corresponding module network model

Fig. 9.16  The architecture of end-to-end module network model

Fig. 9.17  The architecture of GNN-based VQA models
ties. And we overview existing cross-modal representation learning methods for several representative cross-modal tasks, including zero-shot recognition, cross-media retrieval, image captioning, and visual question answering. These cross-modal learning methods either try to fuse information from different modalities into unified embeddings, or try to build embeddings for different modalities in a common semantic space, allowing the model to compute cross-modal similarity. Cross-modal representation learning is drawing more and more attention and can serve as a promising connection between different research areas.

For further understanding of cross-modal representation learning, there are also some recommended surveys and books including:

- Skocaj et al., Cross-modal learning [64].
- Spence, Crossmodal correspondences: A tutorial review [66].
- Wang et al., A comprehensive survey on cross-modal retrieval [72].

In the future, for better cross-modal representation learning, some directions are requiring further efforts:

1. **Fine-grained Cross-modal Grounding.** Cross-modal grounding is a fundamental ability in solving cross-modal tasks, which aims to align semantic units in different modalities. For example, visual grounding aims to ground textual symbols (e.g., words or phrases) into visual objects or regions. Many existing works [27, 74, 76] have been devoted to cross-modal grounding, which mainly focuses on coarse-grained semantic unit grounding (e.g., grounding of sentences and images). Better fine-grained cross-modal grounding (e.g., grounding of words and objects) could promote the development of a broad variety of cross-modal tasks.

2. **Cross-modal Reasoning.** In addition to recognizing and grounding semantic units in different modalities, understanding and inferring the relationship between semantic units are also crucial to cross-modal tasks. Many existing works [37, 41, 82] have investigated detecting visual relation between objects. However, most visual relations in existing visual relation detection datasets do not require complex reasoning. Some works [81] have made preliminary attempts on cross-modal commonsense reasoning. Inferring the latent semantic relationships in cross-modal context is critical for cross-modal understanding and modeling.

3. **Utilizing Unsupervised Cross-modal Data.** Most current cross-modal learning approaches rely on human-annotated datasets. The scale of such supervised datasets is usually limited, which also limits the capability of data-hungry neural models. With the rapid development of the World Wide Web, cross-modal data on the Web have become larger and larger. Some existing works [42, 67] have leveraged unsupervised cross-modal data for representation learning. They first pretrained cross-modal models on large-scale image-caption pairs, and then fine-tuned the models on those downstream tasks, which shows significant improvement in a broad variety of cross-modal tasks. It is thus promising to better leverage the vast amount of unsupervised cross-modal data for representation learning.
References

1. Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. In Proceedings of ECCV, 2016.
2. 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 Proceedings of CVPR, 2018.
3. Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In Proceedings of CVPR, pages 39–48, 2016.
4. Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In Proceedings of ICCV, 2015.
5. Lei Jimmy Ba, Kevin Swersky, Sanja Fidler, and Ruslan Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In Proceedings of ICCV, 2015.
6. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Proceedings of ICLR, 2015.
7. Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Proceedings of NeurIPS, 2013.
8. Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnntokenizer--decoding for statistical machine translation. In Proceedings of EMNLP, 2014.
9. Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional GAN. In Proceedings of ICCV, 2017.
10. Bo Dai and Dahua Lin. Contrastive learning for image captioning. In Proceedings of NeurIPS, 2017.
11. Bo Dai, Deming Ye, and Dahua Lin. Rethinking the form of latent states in image captioning. In Proceedings of CVPR, 2018.
12. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of CVPR, 2009.
13. Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In Proceedings of ACL, 2014.
14. Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In Proceedings of ICCV, 2013.
15. Desmond Elliot and Arjen de Vries. Describing images using inferred visual dependency representations. In Proceedings of ACL, 2015.
16. Desmond Elliot and Frank Keller. Image description using visual dependency representations. In Proceedings of EMNLP, 2013.
17. 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 Proceedings of ECCV, 2010.
18. Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In Proceedings of NeurIPS, 2013.
19. Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question. In Proceedings of NeurIPS, 2015.
20. Jixiang Gu, Jianfei Cai, Gang Wang, and Tszu Han Chen. Stack-captioning: Coarse-to-fine learning for image captioning. In Proceedings of AAAI, 2018.
21. Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Machine Learning Research, 47:853–899, 2013.
22. Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In Proceedings of ICCV, 2017.
23. Xin Huang and Yuxin Peng. Deep cross-media knowledge transfer. In *Proceedings of CVPR*, 2018.
24. Xin Huang, Yuxin Peng, and Mingkuan Yuan. Cross-modal common representation learning by hybrid transfer network. In *Proceedings of IJCAI*, 2017.
25. Zhaoxin Jia, Andrew Gallagher, Ashutosh Saxena, and Tsuhan Chen. 3d-based reasoning with blocks, support, and stability. In *Proceedings of ICCV*, 2013.
26. Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of CVPR*, 2016.
27. Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of CVPR*, 2015.
28. Andrej Karpathy, Armand Joulin, and Fei Fei F Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of NeurIPS*, 2014.
29. Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, 2014.
30. Satwik Kottur, Ramakrishna Vedantam, José MF Moura, and Devi Parikh. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *Proceedings of CVPR*, 2016.
31. Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
32. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of NeurIPS*, 2012.
33. Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating image descriptions. In *Proceedings of CVPR*, 2011.
34. Polina Kuznetsova, Vicente Ordonez, Alexander C Berg, Tamara L Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of ACL*, 2012.
35. Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, 2(10):351–362, 2014.
36. Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of ICML*, 2014.
37. Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of ICCV*, 2017.
38. Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.
39. Xiao Lin and Devi Parikh. Don’t just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *Proceedings of CVPR*, 2015.
40. Chenxi Liu, Junhua Mao, Fei Sha, and Alan L. Yuille. Attention correctness in neural image captioning. In *Proceedings of AAAI*, 2017.
41. Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of ECCV*, 2016.
42. Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of NeurIPS*, 2019.
43. Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of CVPR*, 2017.
44. Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Proceedings of NeurIPS*, 2016.
45. Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of CVPR*, 2018.
46. Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Proceedings of NeurIPS*, 2014.
316 9 Cross-Modal Representation

47. Jiayuan Mao, Yuan Yao, Stefan Heinrich, Tobias Hinz, Cornelius Weber, Stefan Wermter, Zhiyuan Liu, and Maosong Sun. Bootstrapping knowledge graphs from images and text. *Frontiers in Neurorobotics*, 13:93, 2019.

48. Harry McCarley and John MacDonald. Hearing lips and seeing voices. *Nature*, 1976.

49. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*, 2013.

50. Yasuhide Mori, Hironobu Takahashi, and Ryuichi Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of WMISR*, 1999.

51. Jonghwan Mun, Minsu Cho, and Bohyung Han. Text-guided attention model for image captioning. In *Proceedings of AAAI*, 2017.

52. Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of ICML*, 2011.

53. Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. In *Proceedings of ICLR*, 2014.

54. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, 2002.

55. Yuxin Peng, Xin Huang, and Yunzhen Zhao. An overview of cross-media retrieval: Concepts, methodologies, benchmarks and challenges. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

56. Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *Proceedings of NeurIPS*, 2015.

57. Shaqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of NeurIPS*, 2015.

58. Shaqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Proceedings of NeurIPS*, 2015.

59. Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of CVPR*, 2017.

60. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

61. Kevin J Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In *Proceedings of CVPR*, 2016.

62. Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of ECCV*, 2012.

63. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

64. Danijel Skocaj, Ales Leonardis, and Geert-Jan M. Kruijff. *Cross-Modal Learning*, pp. 861–864. Boston, MA, 2012.

65. Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Proceedings of NeurIPS*, 2013.

66. Charles Spence. Crossmodal correspondences: A tutorial review. *Attention, Perception, & Psychophysics*, 73(4):971–995, 2011.

67. Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of ICCV*, 2019.

68. Damien Teney, Lingqiao Liu, and Anton Van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of CVPR*, 2017.

69. Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of CVPR*, 2015.

70. Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of CVPR*, 2015.

71. Cheng Wang, Haojin Yang, Christian Bartz, and Christoph Meinel. Image captioning with deep bidirectional lstms. In *Proceedings of ACM MM*, 2016.
72. Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. arXiv:1607.06215, 2016.
73. Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W. Cottrell. Skeleton key: Image captioning by skeleton-attribute decomposition. In Proceedings of CVPR, 2017.
74. Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In Proceedings of CVPR, 2019.
75. Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In Proceedings of CVPR, 2016.
76. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of ICML, 2015.
77. Ran Xu, Jiasen Lu, Caiming Xiong, Zhi Yang, and Jason J Corso. Improving word representations via global visual context. In Proceedings of NeurIPS Workshop, 2014.
78. Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In Proceedings of EMNLP, 2011.
79. Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In Proceedings of CVPR, 2016.
80. Licheng Yu, Eunhyung Park, Alexander C Berg, and Tamara L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In Proceedings of ICCV, 2015.
81. Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In Proceedings of CVPR, 2019.
82. Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In Proceedings of CVPR, 2017.
83. Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In Proceedings of CVPR, 2016.
84. Bo Zheng, Yibiao Zhao, Joey Yu, Katsushi Ikeuchi, and Song-Chun Zhu. Scene understanding by reasoning stability and safety. International Journal of Computer Vision, 112(2):221–238, 2015.
85. Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In Proceedings of CVPR, 2016.

Open Access  This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.